Luís Seabra Lopes
Nuno Lau
Pedro Mariano
Luís M. Rocha (Eds.)

New Trends in Artificial Intelligence

14th Portuguese Conference on Artificial Intelligence, EPIA 2009
Aveiro, October 12-15, 2009
Proceedings

Universidade de Aveiro

Luís Seabra Lopes
Nuno Lau
Pedro Mariano
Luís M.Rocha (Eds.)

New Trends in Artificial Intelligence

EPIA 2009

EPIA 2009

ISBN 978-972-96895-4-3

Luís Seabra Lopes
Nuno Lau
Pedro Mariano
Luís M. Rocha (Eds.)

New Trends in Artificial Intelligence

14^{th} Portuguese Conference on Artificial Intelligence, EPIA 2009
Aveiro, October 12-15, 2009
Proceedings

Universidade de Aveiro

Title: *New Trends in Artificial Intelligence. 14th Portuguese Conference on Artificial Intelligence. EPIA 2009. Aveiro, October 12-15, 2009. Proceedings*

Volume editors: Luís Seabra Lopes, Nuno Lau, Pedro Mariano, Luís M. Rocha

Printing and binding: DESIGNEED - DESIGN E PUBLICIDADE, UNIP. LDA.

ISBN: 978-972-96895-4-3

Depósito legal: 298966/09

Preface

The *14th Portuguese Conference on Artificial Intelligence*, EPIA'2009, took place at Hotel Meliá Ria, in Aveiro, Portugal, in October 12-15, 2009. This international conference was organized by the University of Aveiro and, as in previous years, was held under the auspices of the Portuguese Association for Artificial Intelligence (APPIA). The purpose of EPIA'2009 was to promote research in all areas of Artificial Intelligence (AI), covering both theoretical/foundational issues and applications, and the scientific exchange among researchers, engineers and practitioners in related disciplines.

To promote discussions among participants, EPIA'2009 was structured as a set of thematic tracks proposed by the international AI research community. Thematic tracks are intended to provide an informal environment that fosters the active cross-fertilization of ideas between researchers within specific subareas of AI, including theoretical/foundational, integrative, application-oriented and newly emerging areas. The proposals received in response to a public Call for Thematic Tracks were evaluated by the Program Chair and the General Chairs in consultation with the Advisory Committee. The following tracks were selected and included in the conference program:

AITUM – Artificial Intelligence in Transportation and Urban Mobility
ALEA – Artificial Life and Evolutionary Algorithms
CMBSB – Computational Methods in Bioinformatics and Systems Biology
COLA – Computational Logic with Applications
EAC – Emotional and Affective Computing
GAI – General Artificial Intelligence
IROBOT – Intelligent Robotics
KDBI – Knowledge Discovery and Business Intelligence
MASTA – Multi-Agent Systems: Theory and Applications
SSM – Social Simulation and Modelling
TEMA – Text Mining and Applications
WNI – Web and Network Intelligence

Each track was coordinated by an Organizing Committee composed of, at least, two researchers in the field, from different institutions. An international Program Committee, with recognized researchers within each track's scientific areas, was created.

In response to the Call for Papers, a total of 163 paper submissions were received from 21 countries, namely Australia, Austria, Belgium, Brazil, Bulgaria, Cuba, France, Germany, Hungary, India, Iran, Italy, Kuwait, Netherlands, Portugal, Singapore, Spain, Taiwan, Tunisia, United Kingdom and USA. All submissions were evaluated by at least three members of the Program Committees of the respective tracks and were selected for presentation at the conference on the basis of quality and relevance to the issues each track is addressing. A selection of higher quality papers presented in the different tracks is published in a book edited by Springer. The remaining papers are published in this proceedings volume. The numbers of submitted papers in each track and the respective numbers of accepted papers are given in Table 1. The overall acceptance rate was 65%.

Table 1. Numbers of papers submitted, accepted and selected for inclusion in the Springer book.

	Submitted	Accepted	Springer LNAI 5816
AITUM	11	7	3
ALEA	12	7	3
CMBSB	5	3	2
COLA	10	8	5
EAC	7	5	3
GAI	11	7	3
IROBOT	27	17	10
KDBI	17	12	7
MASTA	22	13	7
SSM	10	8	4
TEMA	25	15	7
WNI	6	4	1
Total	163	106	55

This edition of EPIA featured, for the first time, a *Nectar Track*, that is, a set of plenary sessions with a selection of top-quality papers accepted in the different tracks. The Nectar Track aims to promote the dissemination of information between research groups with different interests within the AI field, as well as the cooperation between different research groups and the development of integrative research projects. The Nectar Track also aims to provide increased visibility to some of the best papers in the conference program and, in so doing, provide a general view of the AI field and its currently hot topics. After consultation with the EPIA'2009 Advisory Committee, the following papers were included in the Nectar Track:

- "A Data-Based Approach to Integrating Representations of Personality Traits, Values, Beliefs and Behavior Descriptions", by Boon-Kiat Quek, Kayo Sakamoto, and Andrew Ortony
- "Comparing Different Properties Involved in Word Similarity Extraction", by Pablo Gamallo Otero
- "Constraint-based strategy for pairwise RNA secondary structure prediction", by Olivier Perriquet and Pedro Barahona

- "Cost-Sensitive Learning Vector Quantization for Credit Scoring", by Ning Chen, Armando S. Vieira, João Duarte, Bernardete Ribeiro, and João C. Neves
- "Efficient Coverage of Case Space with Active Learning", by Nuno Filipe Escudeiro and Alípio Mário Jorge
- "How much should agents remember? The role of memory size on convention emergence efficiency", by Paulo Urbano, João Balsa, Paulo Ferreira Jr., and Luis Antunes
- "Roles, Positionings and Set Plays to Coordinate a RoboCup MSL Team", by Nuno Lau, Luís Seabra Lopes, Nelson Filipe, and Gustavo Corrente
- "Semantic Image Search and Subset Selection for Classifier Training in Object Recognition", by Rui Pereira, Luís Seabra Lopes, and Augusto Silva
- "Sensitivity Analysis of a Tax Evasion Model Applying Automated Design of Experiments", by Attila Szabó, László Gulyás, and István János Tóth
- "Type Parametric Compilation of Algebraic Constraints", by Marco Correia and Pedro Barahona
- "Using Operator Equalisation for Prediction of Drug Toxicity with Genetic Programming", by Leonardo Vanneschi and Sara Silva

In addition to the parallel sessions for the different tracks and the plenary nectar sessions, the program of EPIA'2009 included plenary talks by distinguished researchers in the AI field, namely:

- Hod Lipson, from Cornell University, with a talk on "The Robotic Scientist: Mining experimental data for dynamical invariants, from cognitive robotics to computational biology".
- Marie-Francine Moens, from Katholieke Universiteit Leuven, with a talk on "More than Just Words: Discovering the Semantics of Text with a Minimum of Supervision".
- Demetri Terzopoulos, from University of California, Los Angeles, with a talk on "Artificial Life Simulation of Humans and Lower Animals: From Biomechanics to Intelligence".

Finally, the program of EPIA'2009 also included the *2nd Doctoral Symposium on Artificial Intelligence* (SDIA).

EPIA'2009 was organized in cooperation with the Special Interest Group on Artificial Intelligence of the Association for Computing Machinery (ACM-SIGART) and the Portuguese Chapter of the IEEE Computational Intelligence Society. The conference was co-sponsored by the Portuguese Research Foundation (FCT) and IEEE Portugal Section. The participation of Demetri Terzopoulos as invited speaker was funded by the Organizing Committee of the ALEA Thematic Track. We highly appreciate and thank the collaboration of the members of all committees, namely the Advisory Committee, the Organizing Committees and Program Committees of the different tracks and the Organizing Committee of SDIA. We also thank invited speakers, authors, referees and session chairs for their contributions to the conference program. Thanks are also due to Microsoft Research, for their Conference Management Service (CMT), which was freely used for managing the paper submission and evaluation processes in EPIA'2009.

Aveiro, October 2009

Luís Seabra Lopes
Nuno Lau
Pedro Mariano
Luís M. Rocha

EPIA 2009 Conference Organization

Conference Chairs

Luís Seabra Lopes	Universidade de Aveiro, Portugal
Nuno Lau	Universidade de Aveiro, Portugal
Pedro Mariano	Universidade de Aveiro, Portugal

Program Chair

Luís M. Rocha	Indiana University, USA

Advisory Committee

Salvador Abreu	Universidade de Évora, Portugal
Pedro Barahona	Universidade Nova de Lisboa, Portugal
Pavel Brazdil	Universidade do Porto, Portugal
Luís Camarinha-Matos	Universidade Nova de Lisboa, Portugal
Amílcar Cardoso	Universidade de Coimbra, Portugal
Helder Coelho	Universidade de Lisboa, Portugal
Luís Correia	Universidade de Lisboa, Portugal
Ernesto Costa	Universidade de Coimbra, Portugal
Gaël Dias	Universidade da Beira Interior, Portugal
Pedro Rangel Henriques	Universidade do Minho, Portugal
Gabriel Pereira Lopes	Universidade Nova de Lisboa, Portugal
José Manuel Machado	Universidade do Minho, Portugal
José Maia Neves	Universidade do Minho, Portugal
Arlindo Oliveira	Universidade Técnica de Lisboa, Portugal
Eugénio Oliveira	Universidade do Porto, Portugal
Luís Moniz Pereira	Universidade Nova de Lisboa, Portugal
José Carlos Príncipe	University of South Florida, USA
Carlos Ramos	Instituto Politécnico do Porto, Portugal
Manuel Filipe Santos	Universidade do Minho, Portugal
Manuela Veloso	Carnegie Mellon University, USA

Thematic Track Chairs

AITUM

Rosaldo Rossetti	Universidade do Porto, Portugal
Ronghui Liu	University of Leeds, UK
Elisabete Arsénio	Laboratório Nacional de Engenharia Civil, Portugal
Jorge Lopes	BRISA S.A., Portugal

ALEA

Agostinho Rosa	Universidade Técnica de Lisboa, Portugal
Luís Correia	Universidade de Lisboa, Portugal

CMBSB

Miguel Rocha	Universidade do Minho, Portugal
Rui Camacho	Universidade do Porto, Portugal
Florentino Fdez-Riverola	Universidad de Vigo, Spain

COLA

Carlos Viegas Damásio	Universidade Nova de Lisboa, Portugal
Fernando Silva	Universidade do Porto, Portugal

EAC

Goreti Marreiros	Instituto Politécnico do Porto, Portugal
Andrew Ortony	Northwestern University, Chicago, USA
Ana Paiva	Universidade Técnica de Lisboa, Portugal
Luis Antunes	Universidade de Lisboa, Portugal

GAI

Luís Seabra Lopes	Universidade de Aveiro, Portugal
Nuno Lau	Universidade de Aveiro, Portugal
Pedro Mariano	Universidade de Aveiro, Portugal
Luís M. Rocha	Indiana University, USA

IROBOT

Luís Paulo Reis	Universidade do Porto, Portugal
Luís Correia	Universidade de Lisboa, Portugal
António José Neves	Universidade de Aveiro, Portugal

KDBI

Nuno Marques	Universidade Nova de Lisboa, Portugal
Paulo Cortez	Universidade do Minho, Portugal
João Moura Pires	Universidade Nova de Lisboa, Portugal
Luís Cavique	Universidade Aberta, Portugal
Manuel Filipe Santos	Universidade do Minho, Portugal
Margarida Cardoso	ISCTE, Portugal
Robert Stahlbock	University of Hamburg, Germany
Zbigniew Michalewicz	University of Adelaide, Australia

MASTA

Ana Paula Rocha	Universidade do Porto, Portugal
Paulo Urbano	Universidade de Lisboa, Portugal
César Analide	Universidade do Minho, Portugal

SSM

Luis Antunes	Universidade de Lisboa, Portugal
Marco Janssen	Arizona State University, USA
Antônio Rocha Costa	Universidade Católica de Pelotas, Brazil
Laszlo Gulyas	Lorand Eotvos University, Hungary

TEMA

Joaquim Ferreira da Silva	Universidade Nova de Lisboa, Portugal
Gabriel Pereira Lopes	Universidade Nova de Lisboa, Portugal
Gaël Dias	Universidade da Beira Interior, Portugal
Vitor Jorge Rocio	Universidade Aberta, Portugal

WNI

Alípio M. Jorge	Universidade do Porto, Portugal
Alneu de Andrade Lopes	Universidade de São Paulo, Brazil
Solange Rezende	Universidade de São Paulo, Brazil

Program Committee

AITUM

Ana Almeida, Portugal
Constantinos Antoniou, USA
Ana Bazzan, Brazil
Moshe Ben-Akiva, USA
Carlos Bento, Portugal
João Bento, Portugal
Eduardo Camponogara, Brazil
Hussein Dia, Australia
Alberto Fernandez, Spain
Franziska Klügl, Sweden
Henry Liu, USA
Kai Nagel, Germany
Luís Nunes, Portugal
Eugénio Oliveira, Portugal
Luís Osório, Portugal
Sascha Ossowski, Spain
Francisco C. Pereira, Portugal
Luís Paulo Reis, Portugal
Giovanna Di Marzo Serugendo, UK
Agachai Sumalee, Hong Kong
Shuming Tang, China
José Telhada, Portugal
Harry Timmermans, The Netherlands
Giuseppe Vizzari, Italy
Fei-Yue Wang, USA
Geert Wets, Belgium
Danny Weyns, Belgium
Chris Zegras, USA

ALEA

Helio Barbosa, Brazil
Ernesto Costa, Portugal
Peter Eggenberger, Denmark

Carlos M. Fernandes, Portugal
Carlos M. Fonseca, Portugal
Inman Harvey, UK
Francisco Herrera, Spain
Colin G Johnson, UK
Fernando Lobo, Portugal
Pedro Mariano, Portugal
Fernando M. Melicio, Portugal
José Ferreira Mendes, Portugal
Juan Julián Merelo, Spain
Francisco B. Pereira, Portugal
José A. Pereira, Portugal
Vitorino Ramos, Portugal
Marc Schonenauer, France
Sara Silva, Portugal
Miroslav Snorek, Czech Republic
Enrique C. Pelaez, Spain
Yonghong Peng, UK
Arlo Zan Randall, USA
Isabel Rocha, Portugal
Luís M. Rocha, USA
Marie-France Sagot, France
Santiago Schnell, USA
Mário Silva, Portugal
Jorge Vieira, Portugal
Mohammed Zaki, USA

CMBSB

Alexessander Alves, UK
Paulo Azevedo, Portugal
Pierre Baldi, USA
Lourdes Borrajo, Spain
Philip Bourne, USA
Pavel Brazdil, Portugal
Chris Bystroff, USA
André Carvalho, Brazil
Vítor Santos Costa, Portugal
Francisco Couto, Portugal
Luc Dehaspe, Belgium
Fernando Diaz-Gomez, Spain
Inês Dutra, Portugal
Timothy Ebbels, UK
Fazel Famili, Canada
Eugénio C. Ferreira, Portugal
Nuno Fonseca, Portugal
Elaine Holmes, UK
Eva Iglesias, Spain
Hasan Jamil, USA
Alípio M. Jorge, Portugal
Rosalia Laza, Spain
Rui Mendes, Portugal
José Mendez-Reboredo, Spain
Kiran Patil, Denmark
Reyes Pavon, Spain

COLA

Salvador Abreu, Portugal
Manuel Ojeda Aciego, Spain
José Júlio Alferes, Portugal
Grigoris Antoniou, Greece
Vítor Santos Costa, Portugal
Bart Demoen, Belgium
Inês Dutra, Portugal
Paulo Gomes, Portugal
Gopal Gupta, USA
Pascal Hitzler, Germany
Priscila Lima, Brazil
Frank Pfenning, USA
Enrico Pontelli, USA
Nuno Silva, Portugal
Umberto Straccia, Italy
Terrance Swift, Portugal
David Warren, USA
Gregory Wheeler, Portugal

EAC

César Analide, Portugal
Luis Antunes, Portugal
Robert Axtell, USA
Win Burleson, USA
Lola Canamero, UK
Amílcar Cardoso, Portugal
Cristiano Castelfranchi, Italy
Helder Coelho, Portugal
Nigel Gilbert, UK
Samer Hassan, Spain
Jennifer Healey, USA

Ian Horswill, USA
Eva Hudlicka, USA
José Machado, Portugal
Goreti Marreiros, Portugal
Stacy Marsella, USA
José Maia Neves, Portugal
Paulo Novais, Portugal
Andrew Ortony, USA
Ana Paiva, Portugal
Juan Pavón, Spain
Paolo Petta, Austria
Boon Kiat Quek, Singapore
Carlos Ramos, Portugal
Ricardo Santos, Portugal
Michelle See, Singapore
Juan Velásquez, USA
Yinping Yang, Singapore

GAI

Salvador Abreu, Portugal
Michael Anderson, USA
Francisco Azevedo, Portugal
Pedro Barahona, Portugal
Carlos Bento, Portugal
Luís Camarinha-Matos, Portugal
Amílcar Cardoso, Portugal
Helder Coelho, Portugal
Jonathan Connell, USA
Juan Corchado, Spain
Luís Correia, Portugal
Anna Really Costa, Brazil
Ernesto Costa, Portugal
Manuel Delgado, Spain
Gaël Dias, Portugal
Inês Dutra, Portugal
Florentino Fdez-Riverola, Spain
Cipriano Galindo, Spain
Pedro Rangel Henriques, Portugal
João Leite, Portugal
Gabriel Pereira Lopes, Portugal
Inês Lynce, Portugal
José Machado, Portugal
Rui Mendes, Portugal
Bernd Neumann, Germany
Arlindo Oliveira, Portugal
Eugénio Oliveira, Portugal
Luis Moniz Pereira, Portugal
José Carlos Príncipe, USA
Ricardo Rabelo, Brazil
Carlos Ramos, Portugal
Fariba Sadri, UK
Manuel Filipe Santos, Portugal
Paulo Urbano, Portugal
Zita Vale, Portugal
Rosa Vicari, Brazil
Michael Witbrock, USA

IROBOT

César Analide, Portugal
Gabriella Sanniti Di Baja, Italy
Stephen Balakirsky, USA
Jacky Baltes, Canada
Reinaldo Bianchi, Brazil
Hans-Dieter Burkhard, Germany
Carlos Carreto, Portugal
Xiaoping Chen, China
Luís Correia, Portugal
Augusto Loureiro da Costa, Brazil
Jorge Dias, Portugal
Marco Dorigo, Belgium
Paulo Gonçalves, Portugal
Huosheng Hu, UK
Nuno Lau, Portugal
Luís Seabra Lopes, Portugal
António Paulo Moreira, Portugal
António José Neves, Portugal
Urbano Nunes, Portugal
Paulo Oliveira, Portugal
Fernando Osório, Brazil
Enrico Pagello, Italy
Fernando Lobo Pereira, Portugal
Armando J. Pinho, Portugal
Mikhail Prokopenko, Australia
Luís Paulo Reis, Portugal
Fernando Ribeiro, Portugal
Maria Isabel Ribeiro, Portugal
Martin Riedmiller, Germany
Jose Miguel Sanchiz, Spain

Eduardo Silva, Portugal
Armando Sousa, Portugal
Guy Theraulaz, France
Flavio Tonidandel, Brazil
Paulo Urbano, Portugal

KDBI

António Abelha, Portugal
Vasilis Aggelis, Greece
Victor Alves, Portugal
Carlos Alzate, Belgium
Orlando Belo, Portugal
Agnès Braud, France
Rui Camacho, Portugal
Logbing Cao, Australia
Emilio Carrizosa, Spain
André Carvalho, Brazil
Ning Chen, Portugal
José Costa, Brazil
Mário Figueiredo, Portugal
Cristian Figueroa-Sepulveda, Chile
João Gama, Portugal
Peter Geczy, Japan
Paulo Gomes, Portugal
Pascal Hitzler, Germany
Beatriz de la Iglesia, UK
Wolfgang Jank, USA
Stéphane Lallich, France
Philippe Lenca, France
Stefan Lessmann, Germany
Wolfram-M. Lippe, Germany
Victor Lobo, Portugal
José Machado, Portugal
Ernestina Menasalvas, Spain
Armando Mendes, Portugal
Patrick Meyer, France
João Pedro Neto, Portugal
José Maia Neves, Portugal
Fátima Rodrigues, Portugal
Joaquim Ferreira da Silva, Portugal
Carlos Soares, Portugal
Murat Caner Testik, Turkey
Theodore Trafalis, USA
Armando Vieira, Portugal
Gregory Wheeler, Portugal

MASTA

César Analide, Portugal
Luis Antunes, Portugal
João Balsa, Portugal
Ana Bazzan, Brazil
Reinaldo Bianchi, Brazil
Luis Botelho, Portugal
Amílcar Cardoso, Portugal
Helder Coelho, Portugal
Antônio Rocha Costa, Brazil
Yves Demazeau, France
Virginia Dignum, The Netherlands
Edmund Durfee, USA
Michael Fisher, UK
Graça Gaspar, Portugal
Marie-Pierre Gleizes, France
Wojtek Jamroga, Germany
Nuno Lau, Portugal
Paulo Leitão, Portugal
Jorge Louçã, Portugal
Luis Macedo, Portugal
José Machado, Portugal
Andreia Malucelli, Brazil
Carlos Martinho, Portugal
John-Jules Meyer, The Netherlands
Jörg Müller, Germany
Paulo Novais, Portugal
Luís Nunes, Portugal
Oliver Obst, Australia
Eugénio Oliveira, Portugal
Andrea Omicini, Italy
Julian Padget, UK
Paolo Petta, Austria
Isabel Praca, Portugal
Mikhail Prokopenko, Australia
Luís Paulo Reis, Portugal
Ana Paula Rocha, Portugal
Rosaldo Rossetti, Portugal
Jordi Sabater, Spain
Jaime Sichman, Brazil
Paulo Urbano, Portugal

SSM

Frederic Amblard, France
Pedro Andrade, Brazil
Robert Axtell, USA
João Balsa, Portugal
Ana Bazzan, Brazil
Pedro Campos, Portugal
Amílcar Cardoso, Portugal
Claudio Cioffi-Revilla, USA
Helder Coelho, Portugal
Nuno David, Portugal
Paul Davidsson, Sweden
Guillaume Deffuant, France
Alexis Drogoul, France
Julie Dugdale, France
Bruce Edmonds, UK
Nigel Gilbert, UK
Nick Gotts, UK
Samer Hassan, Spain
Rainer Hegselmann, Germany
Wander Jager, The Netherlands
Michel Klein, The Netherlands
Jorge Louçã, Portugal
Fernando Neto, Brazil
Mario Paolucci, Italy
Juan Pavón, Spain
Carlos Ramos, Portugal
Juliette Rouchier, France
Keith Sawyer, USA
Jaime Sichman, Brazil
Carles Sierra, Spain
Elizabeth Sklar, USA
Ron Sun, USA
Keiki Takadama, Japan
Jan Treur, The Netherlands
Klaus Troitzsch, Germany
Harko Verhagen, Sweden

TEMA

António Branco, Portugal
Pavel Brazdil, Portugal
Eric de la Clergerie, France
Guillaume Cleuziou, France
Luisa Coheur, Portugal
Walter Daelemans, Belgium
Gaël Dias, Portugal
Antoine Doucet, France
Tomaz Erjavec, Slovenia
Pablo Gamallo, Spain
Pablo Gervás, Spain
Adam Kilgarriff, UK
Mark Lee, UK
Gabriel Pereira Lopes, Portugal
Belinda Maia, Portugal
Nuno Mamede, Portugal
Nuno Marques, Portugal
Veska Noncheva, Bulgaria
Maria das Graças Volpe Nunes, Brazil
Manuel Palomar, Spain
Paulo Quaresma, Portugal
Vitor Jorge Rocio, Portugal
Irene Rodrigues, Portugal
Antonio Sanfilippo, USA
Frédérique Segond, France
Joaquim Ferreira da Silva, Portugal
Aline Villavicencio, Brazil
Spela Vintar, Slovenia
Christel Vrain, France
Katerzyna Wegrzyn-Wolska, France
Pierre Zweigenbaum, France

WNI

Sarabjot Anand, UK
Paulo Azevedo, Portugal
José Luís Borges, Portugal
Pavel Brazdil, Portugal
Pedro Campos, Portugal
Gaël Dias, Portugal
Alípio M. Jorge, Portugal
José Paulo Leal, Portugal
Alneu de Andrade Lopes, Brazil
Nuno Marques, Portugal
Maria Carolina Monard, Brazil
Maria Cristina F. de Oliveira, Brazil
Thiago Pardo, Brazil
Lubos Popelinsky, Czech Republic
Solange Rezende, Brazil

Eduarda Mendes Rodrigues, UK
Giovanni Semeraro, Italy
Mário Silva, Portugal
Carlos Soares, Portugal
Maarten van Someren, Netherlands

External Reviewers

Susana Aguiar, UK
Bettina Berendt, Belgium
Anders Christensen, Portugal
Ruth Cobos, Spain
Debora Donato, Spain
Grzegorz Dziczkowski, France
Luiz Faria, Portugal
Sylvie Guillaume, France
Aleem Hossain, UK
Ido Iurgel, Portugal
Jonathan Klein, USA
Yue-Shi Lee, Taiwan
Zhao Liang, Brazil
Honghai Liu, UK
A. Gil Lopes, Portugal
Henrique Lopes Cardoso, Portugal
Analia Lourenco, Portugal
Ana Madureira, Portugal
P. Mahanti, Canada
Florent Masseglia, France
Jackson Matsuura, Portugal
Dunja Mladenic, Slovenia
Cristi Munteanu, Spain
Kvsvn Raju, India
Arnau Ramisa, Spain
Joaquim Reis, Portugal
Ricardo Rocha, Portugal
Jose Sanchiz, Spain
Maguelonne Teisseire, France
Joana Urbano, Portugal
Antonio Varlaro, Italy
Gary Weiss, USA
Nicolau Werneck, Brazil
Geert Wets, Belgium
Denis Wolf, Brazil
Weizhong Yan, USA
Show-Jane Yen, Taiwan

Table of Contents

Chapter 1 - AITUM - Artificial Intelligence in Transportation and Urban Mobility

Minimizing Airport Peaks Problem by Improving Airline Operations Performance through an Agent Based System . . . 3
António J. M. Castro, António Mota, Luís Paulo Reis, and Eugénio Oliveira

Prediction of Congested Traffic on the Critical Density Point using Machine Learning and Decentralised Collaborating Sensors . . . 15
Wouter Labeeuw, Kurt Driessens, Danny Weyns, Tom Holvoet, Geert Deconinck

Intelligent Transportation Systems: a Ubiquitous Perspective . . . 27
Lúcio Sanchez Passos and Rosaldo J. F. Rossetti

Evaluating Policies for Reservation-based Intersection Control . . . 39
Matteo Vasirani and Sascha Ossowski

Chapter 2 - ALEA - Artificial Life and Evolutionary Algorithms

Genetic Algorithms using Populations based on Multisets . . . 53
António Manso and Luís Correia

How to Build the Network of Contacts . . . 65
Pedro Mariano, Luís Correia, and Carlos Grilo

A Tool for Automatic Routing of Auxiliary Circuits in Ships . . . 77
Paulo Triunfante Martins and Victor J. A. S. Lobo

Predicting the Outcome of Mutation in Genetic Algorithms . . . 87
Sandeep Rajoria, Carlos Soares, Jorge Pinho de Sousa, and Joydip Dhar

Chapter 3 - CMBSB - Computational Methods in Bioinformatics and Systems Biology

Approximate 3D Motif Search in Proteins with Domain Specific Knowledge . . . 101
Arno Formella, Thorsten Pöschel, and Cástor Sánchez Chao

Chapter 4 - COLA - Computational Logic with Applications

Tabling for P-log Probabilistic Query Evaluation 115
Han The Anh, Carroline Kencana Ramli, and Carlos Viegas Damásio

Learning and Reasoning about Uncertainty in the Semantic Web 127
Pedro Oliveira and Paulo Gomes

Using a Contextual Logic Programming Language to Acess Data in Warehousing Systems 139
Valéria Pequeno, Salvador Abreu, and João Carlos Moura Pires

Chapter 5 - EAC - Emotional and Affective Computing

Personality, Emotion and Mood Simulation in Decision Making 155
Ricardo Santos, Goreti Marreiros, Carlos Ramos, José Neves, and José Bulas-Cruz

An Emotion-Driven Interactive System 167
Francisco Ventura, António Oliveira, and Amílcar Cardoso

Chapter 6 - GAI - General Artificial Intelligence

An Intelligent Interface Agent for an Airline Company Web Portal 181
Luís Barbosa, António J. M. Castro, and Luís Paulo Reis

Restoring CSP Satisfiability with MaxSAT 193
Inês Lynce and Joao Marques-Silva

An Evaluation of Heuristic Functions for Bicriterion Shortest Path Problems 205
E. Machuca, L. Mandow and J.L. Pérez de la Cruz

Bringing Relevant Topics to Foster Learning with the Aid of Ontologies .. 217
Elaine Harada Teixeira de Oliveira, Erika Handa Nozawa, Rosa Maria Vicari, and Alberto Nogueira de Castro Júnior

Chapter 7 - IROBOT - Intelligent Robotics

Intelligent Robotic Mapping and Exploration with Converging Target Localization 231
João Certo, João Lobato Oliveira, and Luís Paulo Reis

Robustness and Precision Analysis in Map-Matching based Mobile Robot Self-Localization 243
Manuel Gouveia, António Paulo Moreira, Paulo Costa, Luís Paulo Reis, and Marcos Ferreira

MODEN: Obstacle-driven Elastic Network for Line-of-Sight Communication 254
Francisco S. Melo and Manuela Veloso

Environment Mapping using the Lego Mindstorms NXT and leJOS NXJ . 267
Gerardo Oliveira, Ricardo Silva, Tiago Lira, and Luís Paulo Reis

NeatSqueak on Wheels: Neural Networks Applied to Movement Optimization 279
Hugo Peixoto, João Portela, Rui Teixeira, Filipe Castro, and Luís Paulo Reis

Using Accelerometers To Command A Cleaning Service Robot 288
Armando Sousa and Luís Paulo Reis

Using Barcodes for Robotic Landmarks 300
Armando Sousa, Catarina Santiago, Paulo Malheiros, Paulo Costa, and António Paulo Moreira

Chapter 8 - KDBI - Knowledge Discovery and Business Intelligence

Exploiting Generalized Association Rules 315
Marcos Aurélio Domingues and Solange Oliveira Rezende

An Extension of the Core Method for Continuous Values: Learning with Probabilities 319
Nuno C. Marques

Histogram Based Payload Processing for Unsupervised Anomaly Detection Systems in Network Intrusion 329
Iñigo Perona, Iñaki Albisua, Olatz Arbelaitz, Ibai Gurrutxaga, José I. Martín, Javier Muguerza, and Jesús M. Pérez

Knowledge Discovery Methodology for Medical Reports 341
Vitor Pinheiro and Victor Alves

A Study on Change Detection Methods 353
Raquel Sebastião and João Gama

Chapter 9 - MASTA - Multi-Agent Systems: Theory and Applications

A Formal Notion of Objective Expectations in the Context of Multiagent Systems Routines 367
Antônio Carlos da Rocha Costa and Graçaliz Pereira Dimuro

Towards a Learning Agent Architecture for Cross-Map Transfer 379
Cédric Herpson and Vincent Corruble

A Flexible Agent-Based Framework to Control Virtual Characters 391
Luís Moniz, Graça Gaspar, Ricardo Abreu, Ana Paula Cláudio, and Maria Beatriz Carmo

Telecommunications Fraud: Problem Analysis - an Agent-based KDD Perspective 402
Eugénio Rosas and Cesar Analide

Knowledge Acquisition and Intelligent Agency on the Web of Data 413
Elder Rizzon Santos, Rosa Maria Vicari, and Helder Coelho

A Study of Agents with Self-awareness for Collaborative Behavior 425
Sheng-Wen Wang, Chuen-Tsai Sun, and Chung-Yuan Huang

Chapter 10 - SSM - Social Simulation and Modelling

On the Intelligence of Moral Agency 439
Helder Coelho and Antônio Carlos da Rocha Costa

Fuzzy Evaluation of Social Exchanges Between Personality-based Agents . 451
Graçaliz Pereira Dimuro, A.V. Santos, B.C. Bedregal, and Antônio Carlos da Rocha Costa

A Multi-agent Model for Panic Behavior in Crowds 463
Robson dos Santos França, Maria das Graças Bruno Marietto, and Margarethe Born Steinberger

A Critical Study of the Coherence between EBMs and ABMs in the Simulation of the Hawks-Doves-LawAbiders Society 475
Fernanda Mendez Jeannes and Antônio Carlos da Rocha Costa

Chapter 11 - TEMA - Text Mining and Applications

Combining Unigrams and Bigrams in Semi-Supervised Text Classification 489
Igor Assis Braga, Maria Carolina Monard, Edson Takashi Matsubara

Extraction of Definitions in Portuguese: An Imbalanced Data Set Problem 501
Rosa Del Gaudio and António Branco

Parallel Texts Alignment 513
Luís Gomes, José Aires, and Gabriel Pereira Lopes

Recognizing Polarity and Attitude of Words in Text 525
Laritza Hernández, A. López-Lopez, and José E. Medina

Standardisation of Hotel Descriptions 537
Nuno Miranda, Ricardo Raminhos, Pedro Seabra, José Saias, Teresa Gonçalves, and Paulo Quaresma

Relations Extracted from a Portuguese Dictionary: Results and First Evaluation 541
Hugo Gonçalo Oliveira, Diana Santos, Paulo Gomes

Multi-label Hierarchical Text Classification using the ACM Taxonomy ... 553
António Paulo Santos and Fátima Rodrigues

The Design of OPTIMISM, an Opinion Mining System for Portuguese Politics 565
Mário J. Silva, Paula Carvalho, Luís Sarmento, Pedro Magalhães, and Eugénio Oliveira

Chapter 12 - WNI - Web and Network Intelligence

An Ultra-Fast Modularity-Based Graph Clustering Algorithm 579
Leonardo Jesus Almeida and Alneu de Andrade Lopes

An Updated Portrait of the Portuguese Web 589
João Miranda and Daniel Gomes

Improving Web User Experience with Document Activity Sparklines..... 601
Sérgio Nunes, Cristina Ribeiro and Gabriel David

Author Index 605

Table of Contents - Springer LNAI 5816

Chapter 1 - AITUM - Artificial Intelligence in Transportation and Urban Mobility

Genetic Algorithm for the Calibration of Vehicle Performance Models of Microscopic Traffic Simulators S-3
André Luiz Cunha, José Elievam Bessa Jr., and José Reynaldo Setti

Simulating Communication in a Service-Oriented Architecture for V2V Networks S-15
João F. B. Gonçalves, Edgar F. Esteves, Rosaldo J. F. Rossetti, and Eugénio Oliveira

Applying Event Stream Processing on Traffic Problem Detection S-27
Oliver Pawlowski, Jürgen Dunkel, Ralf Bruns, and Sascha Ossowski

Chapter 2 - ALEA - Artificial Life and Evolutionary Algorithms

Instability in Spatial Evolutionary Games S-41
Carlos Grilo and Luís Correia

Agent-based model of *Aedes aegypti* Population Dynamics S-53
Carlos Isidoro, Nuno Fachada, Fábio Barata, and Agostinho Rosa

Using Operator Equalisation for Prediction of Drug Toxicity with Genetic Programming S-65
Leonardo Vanneschi and Sara Silva

Chapter 3 - CMBSB - Computational Methods in Bioinformatics and Systems Biology

Syntactic Parsing for Bio-Molecular Event Detection from Scientific Literature S-79
Sérgio Matos, Anabela Barreiro, and José Luis Oliveira

Constraint-based strategy for pairwise RNA secondary structure prediction S-86
Olivier Perriquet and Pedro Barahona

Chapter 4 - COLA - Computational Logic with Applications

A Logic Programming System for Evolving Programs with Temporal Operators S-101
José Júlio Alferes, Alfredo Gabaldon, and João Leite

On Improving the Efficiency of Deterministic Calls and Answers in Tabled Logic Programs S-113
Miguel Areias and Ricardo Rocha

On Just In Time Indexing of Dynamic Predicates in Prolog S-126
Vítor Santos Costa

Intention Recognition via Causal Bayes Networks plus Plan Generation S-138
Luís Moniz Pereira and Han The Anh

An ILP system for learning Head Output Connected predicates S-150
José C. A. Santos, Alireza Tamaddoni-Nezhad, and Stephen Muggleton

Chapter 5 - EAC - Emotional and Affective Computing

A Data-Fusion Approach to Representing Personality Traits, Values, Beliefs and Behavior Descriptions S-163
Boon-Kiat Quek, Kayo Sakamoto, and Andrew Ortony

A Formal Model of Emotion-based Action Tendency for Intelligent Agents S-174
Bas R. Steunebrink, Mehdi Dastani, and John-Jules Ch. Meyer

An Emotional and Context-aware Model for Adapting RSS News to Users and Groups S-187
Eugénia Vinagre, Goreti Marreiros, Carlos Ramos, and Lino Figueiredo

Chapter 6 - GAI - General Artificial Intelligence

Type Parametric Compilation of Algebraic Constraints S-201
Marco Correia and Pedro Barahona

Colored Nonograms: an Integer Linear Programming Approach S-213
Luís Mingote and Francisco Azevedo

Learning Visual Object Categories with Global Descriptors and Local Features S-225
Rui Pereira and Luís Seabra Lopes

Chapter 7 - IROBOT - Intelligent Robotics

Analysis and Forecast of Team Formation in the Simulated Robotic Soccer Domain S-239
Rui Almeida, Luís Paulo Reis, and Alípio Mário Jorge

A Cooperative CiberMouse RTSS08 Team S-251
João Azevedo, Miguel Oliveira, Pedro Pacheco, and Luís Paulo Reis

Embodied Language Acquisition: a Proof of Concept S-263
Aneesh Chauhan, Amanda Nascimento, Bruno Werneck, and Luís Seabra Lopes

Predictive Control for Behavior Generation of Omni-Directional Robots S-275
João Cunha, Nuno Lau, João Rodrigues, Bernardo Cunha, and José Luís Azevedo

Towards a Spatial Model for Humanoid Social Robots S-287
Dario Figueira, Manuel Lopes, Rodrigo Ventura, and Jonas Ruesch

Control and Monitoring of a Robotic Soccer Team: The Base Station Application S-299
Nuno M. Figueiredo, António J. R. Neves, Nuno Lau, Artur Pereira, and Gustavo Corrente

Multirobot Task Assignment in Active Surveillance S-310
Nelson Gonçalves and João Sequeira

Roles, Positionings and Set Plays to Coordinate a MSL Robot Team .. S-323
Nuno Lau, Luís Seabra Lopes, Gustavo Corrente, and Nelson Filipe

Semantic Image Search and Subset Selection for Classifier Training in Object Recognition S-338
Rui Pereira, Luís Seabra Lopes, and Augusto Silva

Obstacle Detection, Identification and Sharing on a Robotic Soccer Team S-350
João Silva, Nuno Lau, António J. R. Neves, João Rodrigues, and José Luís Azevedo

Chapter 8 - KDBI - Knowledge Discovery and Business Intelligence

An Algorithm to Discover the k-Clique Cover in Networks S-363
Luís Cavique, Armando B. Mendes, and Jorge M. A. Santos

Cost-Sensitive Learning Vector Quantization for Financial Distress Prediction S-374
Ning Chen, Armando S. Vieira, João Duarte, Bernardete Ribeiro, and João C. Neves

An Intelligent Alarm Management System for Large-Scale Telecommunication Companies S-386
Raúl Costa, Nuno Cachulo, and Paulo Cortez

Construction of a Local Domain Ontology from News Stories S-400
Brett Drury and J. J. Almeida

Efficient Coverage of Case Space with Active Learning S-411
Nuno Filipe Escudeiro and Alípio Mário Jorge

Tracking Recurring Concepts with Meta-learners S-423
João Gama and Petr Kosina

Detecting Errors in Foreign Trade Transactions: Dealing with Insufficient Data S-435
Luis Torgo, Welma Pereira, and Carlos Soares

Chapter 9 - MASTA - Multi-Agent Systems: Theory and Applications

Modeling Autonomous Adaptive Agents with Functional Language for Simulations S-449
Richárd Legéndi, László Gulyás, Rajmund Bocsi, and Tamás Máhr

Recovering from Airline Operational Problems with a Multi-Agent System: a Case Study S-461
António Mota, António J. M. Castro, and Luís Paulo Reis

EcoSimNet: a Multi-Agent System for Ecological Simulation and Optimization S-473
António Pereira, Luís Paulo Reis, and Pedro Duarte

DarkBlade: a Program that Plays Diplomacy S-485
João Ribeiro, Pedro Mariano, and Luís Seabra Lopes

Agent Inferencing Meets The Semantic Web S-497
Paulo Trigo and Helder Coelho

How Much Should Agents Remember? The Role of Memory Size on Convention Emergence Efficiency S-508
Paulo Urbano, João Balsa, Paulo Ferreira Jr., and Luis Antunes

Computing Confidence Values: Does Trust Dynamics Matter? S-520
Joana Urbano, Ana Paula Rocha, and Eugénio Oliveira

Chapter 10 - SSM - Social Simulation and Modelling

Games on Cellular Spaces: An Evolutionary Approach S-535
Pedro Ribeiro de Andrade, Antonio Miguel Vieira Monteiro, and Gilberto Câmara

Context Switching versus Context Permeability in Multiple Social Networks S-547
Luis Antunes, Davide Nunes, Helder Coelho, João Balsa, and Paulo Urbano

Risk Tolerance and Social Awareness: Adapting Deterrence Sanctions to Agent Populations S-560
Henrique Lopes Cardoso and Eugénio Oliveira

Sensitivity Analysis of a Tax Evasion Model Applying Automated Design of Experiments S-572
Attila Szabó, László Gulyás, and István János Tóth

Chapter 11 - TEMA - Text Mining and Applications

Phrase Translation Extraction from Aligned Parallel Corpora Using Suffix Arrays and Related Structures S-587
José Aires, Gabriel Pereira Lopes, and Luís Gomes

Classifying Documents According to Locational Relevance S-598
Ivo Anastácio, Bruno Martins, and Pável Calado

Relieving Polysemy Problem for Synonymy Detection S-610
Gaël Dias and Rumen Moraliyski

Sentiment Classification across Domains S-622
Dinko Lambov, Gaël Dias, and Veska Noncheva

Comparing Different Properties Involved in Word Similarity Extraction S-634
Pablo Gamallo Otero

A Document Descriptor Extractor Based on Relevant Expressions S-646
Joaquim Ferreira da Silva and Gabriel Pereira Lopes

Topic-Related Polarity Classification of Blog Sentences S-658
Michael Wiegand and Dietrich Klakow

Chapter 12 - WNI - Web and Network Intelligence

Item-based and User-based Incremental Collaborative Filtering for web Recommendations S-673
Catarina Miranda and Alípio Mário Jorge

Chapter 1

AITUM - Artificial Intelligence in Transportation and Urban Mobility

Minimizing Airport Peaks Problem by Improving Airline Operations Performance through an Agent Based System

António J.M. Castro[1], António Mota[1], Luís Paulo Reis[1], Eugénio Oliveira[1]

[1] LIACC-NIAD&R, FEUP, Department of Informatics Engineering, University of Porto, 4200-465 Porto, Portugal
{ajmc, amota, lpreis, eco}@fe.up.pt

Abstract. Airports are important infra-structures for the air transportation business. One of the major operational constraints is the peaks of passengers in specific periods of time. Airline companies take into consideration the airport capacity when building the airline schedule and, because of that, the execution of the airline operational plan can contribute to improve or avoid airport peaks problems. The Airline Operations Control Center (AOCC) tries to solve unexpected problems that might occur during the airline operation. Problems related to aircrafts, crewmembers and passengers are common and the actions towards the solution of these problems are usually known as operations recovery. In this paper we propose a way of measuring the AOCC performance that takes into consideration the relation that exists between airline schedule and airport peaks. The implementation of a Distributed Multi-Agent System (MAS) representing the existing roles in an AOCC, is presented. We show that the MAS contributes to minimize airport peaks without increasing the operational costs of the airlines.

Keywords: Disruption management, multi-agent system, airport operation.

1 Introduction

Airports are a very important infrastructure for air transportation. They provide services for airlines and, also, for the passengers that fly on those airlines. Airport operators like to speak of the business of the airport in terms of throughput of passengers and cargo as represented by the annual number of passengers processed or the annual turnover of tons of air freight. This is entirely understandable because, most likely, the annual income is determined by these two parameters. However, from an operational point of view, it is the peak flows that determine the physical and operational costs involved in running a facility. Assigning staff and physical facilities are much more dependent on hourly and daily requirements than on annual throughput.

Airline companies, during their Airline Scheduling Process (especially during the Flight Schedule Generation phase) take into consideration the agreed schedule regarding departures/arrivals of the airports, especially important on hub airports (for more information regarding this process see [8]). Given the above, we believe that it is important for the airline company to operate according to the schedule, not only due to the fact that the airline schedule is the optimal one from the airline perspective but, also, because it takes into consideration the airport capacity and, therefore, the airport peaks. Through operations control mechanisms the airline company monitors all the flights checking if they follow the schedule that was previously defined by other areas of the company. Unfortunately, some problems arise during this phase [5]. Those problems are related to crewmembers, aircrafts and passengers. The Airline Operations Control Centre (AOCC) is composed by teams of people specialized in solving the above problems under the supervision of an operation control manager. Each team has a specific goal contributing to the common and general goal of having the airline operation running with few problems as possible. The process of solving these problems is known as Disruption Management [11] or Operations Recovery.

In this paper we propose a way of measuring the AOCC performance so that, in the decision process, the AOCC takes into account the relation that exists between airport peaks and the airline schedule. We present the architecture and specification of a multi-agent system that was developed for a real airline company, that uses our proposed measure (among other criteria) to solve operational problems.

The rest of the paper is organized as follows. In section 2 we present some related work. Section 3 explains the relation between airport peaks and airline schedule and proposes the AOCC performance criteria. Section 4 shows the architecture and specification of our MAS. In section 5 we present the scenario used to evaluate the system as well as the results of the evaluation. Finally, we discuss and conclude our work in section 6.

2 Related work

We divided the bibliography we have analyzed in three main areas: aircraft recovery, crew recovery and integrated recovery.

Aircraft Recovery: Liu et al. [12] proposes a "multi-objective genetic algorithm to generate an efficient time-effective multi-fleet aircraft routing algorithm" in response to disruption of flights. It uses a combination of a traditional genetic algorithm with a multi-objective optimization method, attempting to optimize objective functions involving flight connections, flight swaps, total flight delay time and ground turn-around times. According to the authors "(…) the proposed method has demonstrated the ability to solve the dynamic and complex problem of airline disruption management". As in other approaches, the authors do use the delay time in the objective functions trying to minimize the delays for all aircrafts and flights. Although there are other differences regarding our approach, the main one is that we emphasize the role of the airport trying to minimize the difference between the real and schedule plan of the airline at each airport.

Crew Recovery: In Abdelgahny et al. [1] the flight crew recovery problem for an airline with a hub-and-spoke network structure is addressed. The paper details and sub-divides the recovery problem into four categories: misplacement problems, rest problems, duty problems, and unassigned problems. The proposed model is an assignment model with side constraints. Due to the stepwise approach, the proposed solution is sub-optimal. According to the authors the tool is able to "solve for the most efficient crew recovery plan with least deviation from originally planned schedule". The major drawback is that it only includes one resource (crew) and does not consider the passenger dimension.

Integrated Recovery: Bratu et al. [4] presents two models that considers aircraft and crew recovery and through the objective function focuses on passenger recovery. They include delay costs that capture relevant hotel costs and ticket costs if passengers are recovered by other airlines. The objective is to minimize jointly airline operating costs and estimated passenger delay and disruption costs. According to the authors, "(…) decisions from our models can potentially reduce passenger arrival delays (…) without increasing operating costs". The main difference regarding our approach is that we emphasize the role of the airport trying to minimize the difference between the real and schedule plan of the airline at each airport. Castro and Oliveira [6] present a Multi-Agent System (MAS) to solve airline operations problems, using specialized agents in each of the three usual dimensions of this problem: crew, aircraft and passengers. The authors only use operational costs on the decision process ignoring if the AOCC is near the original schedule or not.

Other Application Domains: Agents and multi-agent systems have been applied both to other problems in air transportation domain and in other application domains. A brief and incomplete list of such applications follows. Tumer and Agogino [14] developed a multi-agent algorithm for traffic flow management. Wolfe et al. [15] uses agents to compare routing selection strategies in collaborative traffic flow management. For ATC Tower operations, Jonker et al. [9] have also proposed the use of multi-agent systems. As a last example, a multi-agent system for the integrated dynamic scheduling of steel production has been proposed by Ouelhadj [13].

3 Airport Peaks, Airline Schedule and Operations Performance

According to Ashford et al. [2] there are four ways of describing variations in demand level with time:

1. Annual variation over time.
2. Monthly peaks within a particular year.
3. Daily peaks within a particular month or week.
4. Hourly peaks within a particular day.

The first one is very important from the viewpoint of planning and provision of facilities. For our work, we concentrate on monthly, daily and hourly peaks, because these are the ones that have more impact on day-to-day operations of the airports and airlines. The goal of airport operators is to spread demand more evenly over the operating day in order to decrease the costs associated with running the airport at peak times, avoiding, as much as possible, situations like the one presented in Figure 1.

On the other hand, airlines are looking to maximize fleet utilization and offer flights in more attractive slots. Additionally, airlines that operate in a hub-and-spoke, due to the characteristics of such an operation, want to minimize the total travel time and, for that, they need to rapidly connect the passengers that are arriving from long-haul flights to short-haul flights and vice-versa.

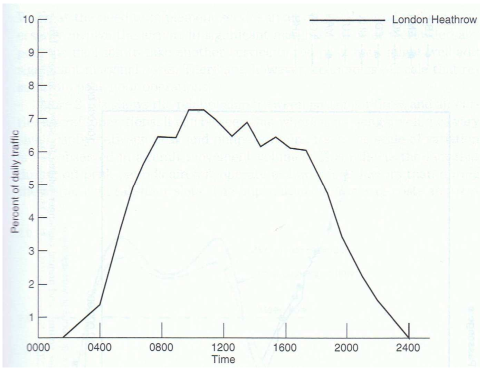


Fig. 1. Hourly variations of passenger traffic in a typical peak day (Source: BAA plc.)

As it is possible to see in Figure 2 this type of network makes airline companies to schedule *waves*, that is, a high number of aircrafts arriving or departing at the hub in a short time interval.

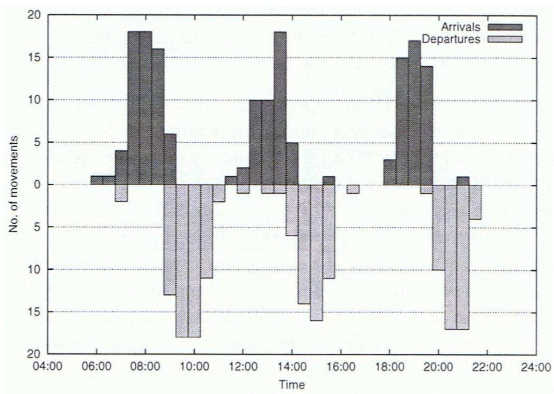


Fig. 2. Schedule structure of the Alitalia hub in Milan (MXP). (Source: [7])

Given the difference between the goal of the airport operator and the goal of the airline, there is, therefore, "a potential conflict between the airline satisfying its customer, the passenger, and the airport attempting to influence the demands of its customer, the airline" [2]. Because of that, it is important that airport operators and airlines cooperate regarding the flight schedule definition.

International Air Transport Association (IATA) has developed a general policy in scheduling so that, at some airports with official limitations, general government authorities carry out the coordination. From our own personal knowledge of the air transportation business as well as according to Ashford et al. [2], it is much more common the situation where the airlines establish themselves an agreed schedule through the mechanism of airport coordinator. The largest or national carrier of the airport, assumes the role of airport coordinator (TAP in Lisbon, Lufthansa in Frankfurt and BA in Heathrow, for example) and at semi-annual IATA scheduling conferences, they are able to set an agreed schedule for the airports they represent. As we stated before, the Airline Scheduling Process takes into consideration the agreed schedule. The goal of that process is to create an airline schedule that is optimal in regard to a given objective, usually operating profit [8], that is, minimum costs and maximum revenues. To operate according to the schedule is not an easy task. Airline companies face a lot of unexpected events during the operations of their flights [11]. However a good disruption management process should exist to minimize the impact of the unexpected events and, according to Yu [16] return to the original schedule as soon as possible[1]. For that, the AOCC should take decisions when solving disruptions that tend to the original schedule. We propose to measure the performance of AOCC's according to Equation 1.

$$\rho = \sum_{t=1}^{n} \sum_{a=1}^{|A|} \sum_{f=1}^{|F|} \left|\Delta dt_{\{f,a,t\}}\right| + \left|\Delta at_{\{f,a,t\}}\right| \tag{1}$$

$\underline{\textbf{\textit{where}}}$

$f \in F; F = \{flights\}, \quad a \in A; A = \{airports\}$

$t = time\ period\ (days), t \geq \alpha \leq \beta,$

$\alpha = start\ datetime\ of\ the\ AOCC$

$\beta = end\ datetime\ of\ the\ AOCC$

$\Delta dt_{\{f,a,t\}} : schedule/actual\ departure\ variation$

$\Delta at_{\{f,a,t\}} : schedule/actual\ arrival\ variation$

We might say that if ρ->0 (tends to) then the real operation of the airline is running near the original schedule contributing to improve the performance of the airport during peak times. At the same time, we might say that the AOCC is contributing to minimize the real operational costs. In the next section we present the multi-agent system (MAS) we have developed to help the AOCC. The MAS uses the performance criteria above (Equation 1) and, also, other criteria related with operational costs.

4 System Architecture and Specification

System overview: This section presents the architecture and specification of the multi-agent system (MAS) we have developed for the airline operations control centre (AOCC). Figure 3 shows one instance of the architecture of the system. There are seven types of agents:

---

[1] Assuming the original schedule as the optimal one.

- *Monitor*, which monitors the operation of the airline company.
- *EventType*, which defines the types of events that must be detected.
- *ResolutionManager*, which receives a problem and manages the resolution in cooperation with the specialist agents.
- *SimmAnneal* and *HillClimb*, specialist agents responsible for the resolution of a problem, using simulated annealing and hill climb algorithms, respectively.
- *Supervisor*, the agent that interacts with the human supervisor, showing the solutions proposed and requesting authorization to apply them.
- *ApplySolution*, the agent that is responsible to apply the solution in the environment.

Figure 3 shows also the existence of a data store, which has information about the airline company operations. The data store is accessed by the *Monitor*, *Specialist* and *ApplySolution* agent. The communication between the agents is done through the JADE system [3] and the data is passed between agents as serializable Java objects.

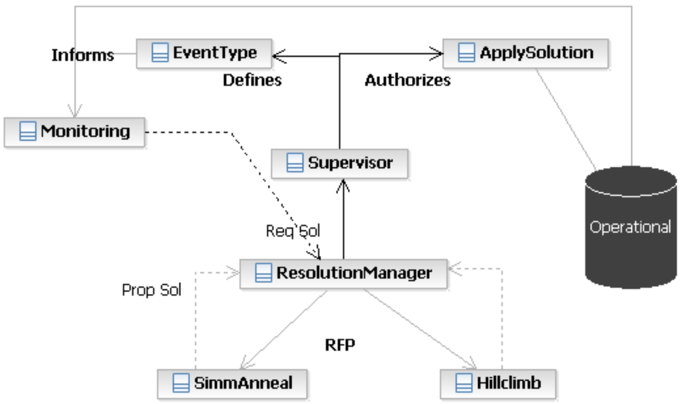


Fig. 3. Overall architecture of the Multi-Agent System

Airline Schedule and Actual Operational Plan Definition: In a simplified version of an airline schedule, we may say that it is composed of flights and the resources necessary to perform those flights (aircraft and crewmembers), in a specific period. The AOCC typically takes control of the airline operational plan some hours/days before the operation of a flight until some hours/days after. We can define the airline schedule plan S as:

$$S = \{s_{\{f,a,t\}}: s_{\{f,a,t\}} = (id, dp^s_{\{f,a,t\}}, ar^s_{\{f,a,t\}}, dt^s_{\{f,a,t\}}, at^s_{\{f,a,t\}}, ac^s_{\{f,a,t\}}, cm^s_{\{f,a,t\}}, to^s_{\{f,a,t\}})\} \quad \textbf{(2)}$$
$$t \geq \alpha \leq \beta$$

where

$id : flight\ identification$

$dp^s_{\{f,a,t\}} : sched.\ departure\ airport;\ \ dp^s_{\{f,a,t\}} \in A, dp^s_{\{f,a,t\}} = a$

$ar^s_{\{f,a,t\}} : sched.\ arrival\ airport;\ \ ar^s_{\{f,a,t\}} \in A, ar^s_{\{f,a,t\}} \neq a$

$dt^s_{\{f,a,t\}} : sched.\ departure\ date\ time$

$at^s_{\{f,a,t\}} : sched.\ arrival\ date\ time$

$ac^s_{\{f,a,t\}} : sched.\ aircraft\ assigned\ flight;\ \ ac^s_{\{f,a,t\}} \in AC: AC = \{aircrafts\}$

$cm^s_{\{f,a,t\}} : sched.\ crew\ assigned\ flight;\ \ cm^s_{\{f,a,t\}} \in CM: CM = \{crewmembers\}$

$to^s_{\{f,a,t\}} : sched.\ total\ operational\ cost$

Similarly, the actual operational plan R can be defined as:

$$R = \{r_{\{f,a,t\}}: r_{\{f,a,t\}} = (id, dp^r_{\{f,a,t\}}, ar^r_{\{f,a,t\}}, dt^r_{\{f,a,t\}}, at^r_{\{f,a,t\}}, ac^r_{\{f,a,t\}}, cm^r_{\{f,a,t\}}, to^r_{\{f,a,t\}})\} \quad \textbf{(3)}$$
$$t \geq \alpha \leq \beta$$

In the case of Equation 3 the components reflect the actual data as opposed to the schedule data in S (the airline schedule plan).

Problem Specification: When agent *Monitoring* detects a problem that needs to be solved, a problem is raised and a solution is requested (through a *Fipa-Request*[2] protocol) from the *ResolutionManager* agent. Equation 4 represents the problem.

$$P = \{p_{\{f,a,t\}}: p_{\{f,a,t\}} = (id, dt^p_{\{f,a,t\}}, et^p_{\{f,a,t\}}, cm^p_{\{f,a,t\}}, dly^p_{\{f,a,t\}}, svr^p_{\{f,a,t\}}, tw^p_{\{f,a,t\}}, bd^p_{\{f,a,t\}}, csd^p_{\{f,a,t\}}, to^p_{\{f,a,t\}}, s_{\{f,a,t\}})\} \quad \textbf{(4)}$$
$$t \geq \alpha \leq \beta$$

where

id : problem id.

$dt^p_{\{f,a,t\}} = dt^s_{\{f,a,t\}}$

$et^p_{\{f,a,t\}} \in E: E = \{flight\ delay, crew\ delay, pax\ delay\}$

$dly^p_{\{f,a,t\}}$: minutes of delay

$to^p_{\{f,a,t\}} = to^s_{\{f,a,t\}}$

$svr^p_{\{f,a,t\}} \in SV: SV = \{warning, problem\}$

$tw^p_{\{f,a,t\}}$: time window for change, $\alpha < (tw^p_{\{f,a,t\}} - dt^p_{\{f,a,t\}});\ \beta > (tw^p_{\{f,a,t\}} + dt^p_{\{f,a,t\}})$

$$cm^p_{\{f,a,t\}} = \begin{cases} cm^s_{\{f,a,t\}}, & if\ et^p_{\{f,a,t\}} = \{crew\ delay\} \\ \emptyset, & otherwise \end{cases}$$

$$bd^p_{\{f,a,t\}} = \begin{cases} 0, & if\ p_{\{f,a,t\}}\ raised\ by\ Monitor\ agent \\ > 0, & after\ CFP\ by\ ResolutionManager\ agent \end{cases} : bid\ dealine\ in\ minutes.$$

$$csd^p_{\{f,a,t\}} = \begin{cases} 0, & if\ p_{\{f,a,t\}}\ raised\ by\ Monitor\ agent \\ > 0, & after\ accept\_proposal\ by\ ResolutionManager\ agent \end{cases} : candidate\ solution\ deadline\ in\ minutes.$$

For example: crewmember 231, delayed 10 minutes for flight TP438, departing from Lisbon would be represented as problem:

$p_{\{tp438,lis,0715\}}$=(001,09/07/01-0715,crew delay,231,10,problem,3,0,0,$s_{\{tp438,lis,0715\}}$).

At this stage, the *Monitoring* agent only adds to the *Problem*, information related with the operational plan time window that should be involved in the resolution process. In the example above, each specialist agent only considers flights between 04:15 and 10:15 of day 09/07/01 (09/07/01 07:15 ± 3 hours).

Resolution Manager: Agent *ResolutionManager* responds to the request from *Monitoring* agent, issuing a RFP to the specialist agents *SimmAnneal* and *HillClimb* and adding to the *Problem* the bid deadline and the deadline to receive candidate solutions (for example, $bd^p_{\{tp438,lis,0715\}}=1$ and $csd^p_{\{tp438,lis,0715\}}=5$). At this level a *Fipa-Contract.net* protocol is used to negotiate with the specialist agents. The specialist agents interested to respond to the RFP, should manifest that intention before the bid deadline. The specialist agents have a limited amount of time to find a candidate solution that is represented by $csd^p_{\{tp438,lis,0715\}}$ in $p_{\{tp438,lis,0715\}}$ (example above). The first step of the specialist agents is to obtain the flights that are in the time window of the problem, represented by $tw^p_{\{tp438,lis,0715\}}$. The set $FT_{\{p\}}$ represents these initial

[2] http://www.fipa.org

flights and will be the initial solution of the problem. The crewmembers and aircraft exchanges are made between flights of $FT_{\{p\}}$. Finally, when a candidate solution is found, the specialist agents send it to the *ResolutionManager* agent. Equation 5 defines a candidate solution.

$$PS = \{ps_{\{p\}}: ps_{\{p\}} = \left(id, fc^{ps}_{\{p\}}, ic^{ps}_{\{p\}}, \{ft^{ps}_{\{p\}}\}, \{cm^{ps}_{\{p\}}\}, r_{\{f,a,t\}}\right)\} \quad (5)$$

where
$p \in PR$
id : *problem solution identification*
$ic^{ps}_{\{p\}}$: *initial solution cost*
$fc^{ps}_{\{p\}}$: *final solution cost*
$\{ft^{ps}_{\{p\}}\} \subset FT_{\{p\}}$
$\{cm^{ps}_{\{p\}}\} \subset CM$

A representation of a candidate solution for the example problem above could be:

$ps_{\{001\}}$=(1200,959,{flights},{crewmembers},$r_{\{tp438,lis,0715\}}$).

For each and all candidate solutions agent *ResolutionManager* calculates the AOCC Performance ρ, using Equation 1. The candidate solution with the minimum value of ρ will be the one that it is sent to the *Supervisor* for approval.

Solution generation and evaluation: The generation of a new solution, by the specialist agents *HillClimb* and *SimmAnneal*, is made by finding a successor that distances itself to the current solution by one unit, that is, the successor is obtained by one, and only one, of the following operations:
- Swap two aircrafts between flights that belong to the flights that are in the time window of the problem.
- Swap two crewmembers between flights that belong to the flights that are in the time window of the problem.
- Swap an aircraft that belongs to the flights that are in the time windows of the problem with an aircraft that that is not being used.
- Swap a crewmember of a flight that belongs to the flights that are in the time window of the problem with a crewmember that isn't on duty, but is on standby.

When choosing the first element (crewmember or aircraft) to swap, there are two possibilities:
- Choose randomly
- Choose an element that is delayed.

This choice is made based on the probability of choosing an element that is late, which was given a value of 0.9, so that the algorithms can proceed faster to good solutions (exchanges are highly penalized, so choosing an element that is not late probably won't reduce the cost, as a possible saving by choosing a less costly element probably won't compensate the penalization associated with the exchange). If the decision is to exchange an element that is delayed, the list of flights will be examined and the first delayed element is chosen. If the decision is to choose randomly, then a random flight is picked, and a crewmember or the aircraft is chosen, depending on the probability of choosing a crewmember, which was given a value of 0.85. When choosing the second element that is going to swap with the first, there are two possibilities:

– Swap between elements of flights.
– Swap between an element of a flight and an element that isn't on duty.

This choice is made based on the probability of choosing a swap between elements of flights, which was given a value of 0.5. The evaluation of the solution is done by an objective function that measures four types of costs:

– The costs with crewmembers. Those costs take into consideration the amount that has to be paid to the crewmember (depends on the duration of the flight), and the base of the crewmember (for instance, assign a crewmember from Oporto to a flight departing from Lisbon has an associated cost that would not be present if the crewmember's base was Lisbon).
– The costs with aircrafts. Those costs take into consideration the amount that has to be spent on the aircraft (depends on the duration of the flight), and the base where the flight actually is.
– The penalization for exchanging elements.
– The penalization for delayed elements. The cost associated with this aspect is the highest, because the goal is to have no delayed elements.

These types of costs are taken into account in Equation 6:

$$tc = cmc + amc + exW * numE + dlW + numD \qquad \textbf{(6)}$$

Where

$$cmc = \sum_{i=1}^{|CM|} (c_i * bcf)/numCm \qquad \textbf{(7)}$$

$$\begin{aligned}&where\\&i \in CM; CM = \{all\ crewmembers\ in\ flight\}\\&1 < bcf \leq 2 : base\ crew\ factor\\&numCm : number\ of\ crew\ members\ in\ CM\end{aligned}$$

$$amc = \sum_{j=1}^{|AC|} (ac_j * baf)/numAc \qquad \textbf{(8)}$$

$$\begin{aligned}&where\\&j \in AC; AC = \{aircraft\ same\ fleet\}\\&1 < baf \leq 2 : base\ aircraft\ factor\\&numAc : number\ of\ aircrafts\ in\ AC\end{aligned}$$

exW was given a value of 1000, and *dlW* a value of 20000.

Regarding the agent that implements a *Simmulated Annealing* algorithm [10], there is a probability that a new solution is selected even if the cost is not smaller than the previous one. Our agent has used the following values for calculating this probability: α=0.8, T=10 and T updated every N iterations (N=2).

5 Scenario and Experiments

Scenario: To evaluate our approach we have setup the same scenario used by the authors in [6] that include 3 operational bases (A, B and C). Each base, corresponding to a different airport, includes their crewmembers each one with a specific roster. Airport B is the Hub of the airline. In this small experiment it is included 15 flights, 36 crewmembers and 4 aircrafts. After setting-up the scenario we found the solutions for each crew event using our system (running only once). After that, the AOCC performance for each method was calculated according to Equation 1 and considering a one month period. As a final step, the solutions found by our system were presented to AOCC users to be validated regarding feasibility and correctness.

Results: Table 1 presents the results that compare our method (method B) with the one used by Castro and Oliveira [6]. From the results obtained we can see that method B increased 1.38 times the performance of the AOCC. As we stated in section 3, if the AOCC performance tends to zero it means that the airline is operating (in terms of flight departure and arrivals times) more according to the airline schedule and, because of the relation that exist between airline schedule and airport peaks (as we explained in section 3), it means that the airline contributes also to a better passenger flow at the airports. From Table 1 we see that performance of AOCC in our method (B) is closer to zero than previous method.

Regarding the performance in each airport, our approach improved the performance of the AOCC in airport A and B by 2 and 1.75 times, respectively. For airport C the performance is the same of previous method. Another important result is the total costs. Our method is 23% less expensive than the previous one.

Table 1. Comparison of the results

		Method A		**Method B**		**A/B**
	Flights	**ρ**		**ρ**		
Global		180		130		1.38
- Airport A	3	40		20		2.00
(0-13h)	1	30		20		
(13-20h)	2	10		0		
(20-24h)	0			0		
- Airport B	7	70		40		1.75
(0-13h)	2	10		10		
(13-20h)	4	50		30		
(20-24h)	1	10		0		
- Airport C	5	70		70		0.00
(0-13h)	3	20		15		
(13-20h)	2	50		55		
(20-24h)	0	0		0		
Total costs		11628		8912		-23%

6 Discussion and Conclusions

In this paper we proposed a way of evaluating the performance of the AOCC that takes into consideration the relation that exists between the airport peaks (airport capacity) and the airline schedule.

We have implemented a MAS that represents the roles in the AOCC and that solves the unexpected problems that usually happens on airline operations. Our MAS is able to take decisions taking into consideration the AOCC performance as well as the airline operational costs. Preliminary results show that it is possible to contribute to minimize the airports peaks without increasing the airline operational costs. However, due to the probabilistic nature of the simulated annealing algorithm and due to the fact that we have run our system only once to get the results, we cannot generalize the results presented here. Another conclusion that we are able to take from the results in Table 1, is that it would be important to collect the reason that caused the flight/crew delay, i.e., due to weather conditions, ATC and/or airport restrictions, aircraft malfunction, etc. This information would help to understand some of the results. For example, it could help to understand why the performance on airport C is the same as the one obtained by the previous method.

We also point out that these results, *per se*, do not mean that we are able to solve all the airport peak problems in a specific airport. The airport peak problems are the result of the passenger flow that is generated by several airline schedules that operate at the airport. It would be necessary that all airlines implemented a similar system to reach to such a conclusion. However, in the airports where an airline has a hub-and-spoke network, the majority of the passenger flow is generated by a single airline company. In those cases, our approach could contribute significantly to minimize the airport peak problems.

Finally, our MAS is an integrated system that automates much of the disruption management process, from the monitoring of the operation of the company in its several bases, to the detection of events and the resolution of the problems encountered. Additionally, our MAS is oriented to the future: its distributed nature and the fact that it is based on agents that are specialists in solving problems easily allows the insertion into the system of new agents that solve new kinds of problems that were identified in the meantime, or that resolve the current types of problems using different methods. It is thus a truly scalable solution, prepared to sustain the growth of the airline company. Although the goals have been achieved, it is important to consider a number of improvements that could be made on future developments, and that could enrich it. In terms of the algorithms used to solve the problems, other meta-heuristics can be implemented, as well as methods based in the area of operational research. The fact that this is a distributed system means that there is no theoretical limit to the number of agents that try to solve, at the same time, the same problem. It is also important to collect more data and run the system several times to get more conclusive and generic results.

Acknowledgments. The first author is supported by FCT (Fundação para a Ciência e Tecnologia) under research grant SFRH/BD/44109/2008. The authors are grateful to TAP Portugal for allowing the use of real data from the company.

References

1. Abdelgahny, A., Ekollu, G., Narisimhan, R., and Abdelgahny, K.: A Proactive Crew Recovery Decision Support Tool for Commercial Airlines during Irregular Operations. Annals of Operations Research, 127:309-331 (2004).
2. Ashford, N., Stanton, H., and Moore, C.: Airport Operations. Second Edition, McGraw-Hill, ISBN: 978-0-07-003077-0 (1997).
3. Bellifemine, F., Caire, G., Trucco, T., and Rimassa, G.: JADE Programmer's Guide. JADE 3.3 TILab S.p.A. (2004).
4. Bratu, S., and Barhhart, C.: Flight Operations Recovery: New Approaches Considering Passenger Recovery. Journal of Scheduling 9(3):279-298 (2006).
5. Clausen, J., Larsen, A., and Larsen, J.: Disruption Management in the Airline Industry – Concepts, Models and Methods. Technical Report, 2005-01, Informatics and Mathematical Modeling, Technical University of Denmark, DTU (2005).
6. Castro, A.J.M., and Oliveira, E.: Using Specialized Agents in a Distributed MAS to Solve Airline Operations Problems: a Case Study. Proceedings of IAT 2007 (Intelligent Agent Technology Conference), pp. 473-476, Silicon Valley, California, USA 2-5 November 2007, IEEE Computer Society, ISBN: 0-7695-3027-3.
7. Danesi, A.: Measuring airline hub timetable coordination and connectivity: definition of a new index and application to a sample of European hubs. European Transport 34, pp. 54-74 (2006).
8. Grosche, T.: Computational Intelligence in Integrated Airline Scheduling. Studies in Computational Intelligence 173, Springer-Verlag Berlin Heidelberg, ISBN: 978-3-540-89886-3 (2009).
9. Jonker, G., Meyer, J.J., and Dignum, F.: Towards a Market Mechanism for Airport Traffic Control, 12^{th} Portuguese Conference on Artificial Intelligence (EPIA 2005), Covilhã, Portugal (2005).
10. Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P.: Optimization by Simulated Annealing. Science, New Series, Vol. 220, No. 4598, pp. 671-680 (May 13, 1983).
11. Kohl, N., Larsen, A., Larsen, J., Ross, A., and Tiourline, S.: Airline Disruption Management – Perspectives, Experiences and Outlook. Technical Report, CRTR-0407, Carmen Research (2004).
12. Liu, T., Jeng, C., and Chang, Y.: Disruption Management of an Inequality-Based Multi-Fleet Airline Schedule by a Multi-Objective Genetic Algorithm, Transportation Planning and Technology, Vol. 31, No. 6, pp. 613-639 (2008).
13. Ouelhadj, D.: A Multi-Agent System for the Integrated Dynamic Scheduling of Steel Production, Ph.D. dissertation, The University of Nottingham, School of Computer Science and Information Technology, England (2003).
14. Tumer, K., and Agogino, A.: Distributed Agent-Based Air Traffic Flow Management, 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii (2007).
15. Wolfe, S. R., Jarvis, P. A., Enomoto, F. Y., and Sierhuis, M.: Comparing Route Selection Strategies in Collaborative Traffic Flow Management., Proceedings of IAT 2007 (Intelligent Agent Technology Conference), Silicon Valley, California, USA 2-5 November 2007, IEEE Computer Society, ISBN: 0-7695-3027-3 (2007).
16. Yu, G., and Qi, X.: Disruption Management: Framework, Models and Applications. World Scientific Publishing Co. Pte. Ltd, ISBN: 981-256-017-3 (2004)

Prediction of Congested Traffic on the Critical Density Point using Machine Learning and Decentralised Collaborating Sensors

Wouter Labeeuw[1], Kurt Driessens[2], Danny Weyns[2], Tom Holvoet[2], Geert Deconinck[1]

[1] Departement of Electrical Engineering,
{wouter.labeeuw,geert.deconinck}@esat.kuleuven.be
[2] Department of Computer Science,
{kurt.driessens,danny.weyns,tom.holvoet}@cs.kuleuven.be
Katholieke Universiteit Leuven, Belgium

Abstract. In this paper we discuss short term traffic congestion prediction, more specifically, prediction of the sudden speed drop when traffic resides at the critical density point. We approach this problem using standard machine learning techniques combining information from multiple sensors measuring density and average velocity. The model used for prediction is learned offline. Our goal is to implement (and possibly update) the predictive model in a multi-agent system, where coupled with each sensor, there is an agent that monitors the condition of traffic, starts to collect data from other sensors located nearby when necessary and is able to predict local sudden speed drops so that drivers can be warned ahead of time. We evaluate Gaussian processes, support vector machines and decision trees not only limited to predictive accuracy, but also the suitability of the learned model in the setup as described above, i.e., keeping in mind that we want the warning system to be decentralized and want to ensure scalability and robustness.

Keywords: road traffic, short term velocity drop prediction, critical density, machine learning, multi-agent system

1 Introduction

The bulk of existing work on traffic prediction focuses on density [7, 1] and occupancy [16] prediction. The systems that use these predictions are employed to traffic management purposes [5].

In this paper, we focus on short-term predictions of traffic velocity instead. Our goal is to design a distributed system that can predict the sudden, local drop of speed that marks the start of congested traffic. If local predictions can be made with sufficient accuracy, we can warn oncoming traffic of the expected trouble ahead of time. Specifically, we are interested in the question: “Can we predict local future velocity drops at the critical density point in a decentralized way?”

The critical density point is the density at which the behavior of traffic is least predictable. Traffic flow can be distinguished into regimes [15]: free-flow, capacity-flow, congested, stop and go and jammed traffic. In free-flow traffic, vehicles can travel freely at their desired speed. When more and more vehicles join in, free-flow traffic gets denser until it reaches the capacity-flow, i.e., the maximum number of vehicles able to maintain free-flow traffic. The border between capacity-flow and congested traffic is the critical density point and is situated around 25 vehicles per kilometer per lane [20]. In this paper, a practical attempt to predict the future velocity at this transition point is made using standard machine learning algorithms. The experimental setup consists of sensors and agents that try to predict the future average velocity within their own range, using self collected data and data from the surrounding sensors. The experiments are build upon traffic simulation software [17] in which an intelligent driver model realistically directs the behavior of the individual vehicles [18] and use the machine learning suite Weka [21] for the machine learning components.

2 Predicting traffic congestion

Figure 1 represents the average velocity in meters per second as measured by a sensor over time and illustrates the prediction task we focus on in this work. The plot of the average velocity will be slightly different for sensors at different locations, but the sudden drops look similar in each plot. We consider the sudden drop to be the most useful information to predict, since it can be used to notify drivers of an upcoming dangerous situation. If drivers can be notified 1 km in advance that a strong reduction in speed is expected, it could significantly lower the probability of an accident happening due to distractions or a loss in concentration. Note that we focus on the region where the velocity stays more or less constant until the drop. The area indicated by B represents stop and go waves. Once traffic is in a stop and go wave, it will stay in a stop and go wave if the density does not drop drastically. The backpropagation of such a wave is easier to predict and has been handled in previous work [18]. In this work, we will focus on predicting the transition from A to B when, from the view of the driver, the congestion is hardest to anticipate.

The Dutch traffic information service divides traffic congestion into three subclasses: slow traffic (min 25 km/h and max 50 km/h for minimum 2km), stationary traffic (max 20 km/h) and a combination of both [9]. The chosen cut off point for the velocity is 7 m/s, which more or less equals 25 km/h. The vertical line in figure 1 indicates the time of this drop below 7 m/s.

We will approach this problem using supervised machine learning. Supervised machine learning builds predictive models using labelled training examples. This model can then be used to make predictions about the labels of previously unseen examples. The task of our learning problem is: "Given data from a local sensor and its surrounding sensors, learn a model that correctly predicts the velocity drop". For this feasibility study, we use the Weka tool [21], which implements a number of different machine learning algorithms.

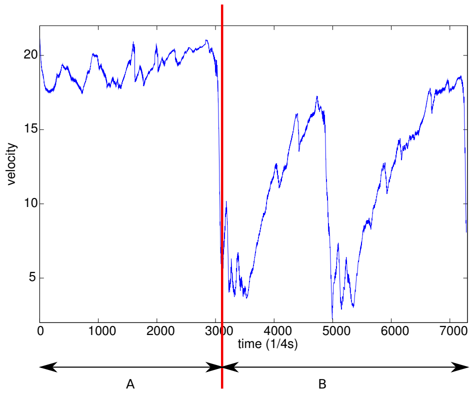


Fig. 1. Velocity drop

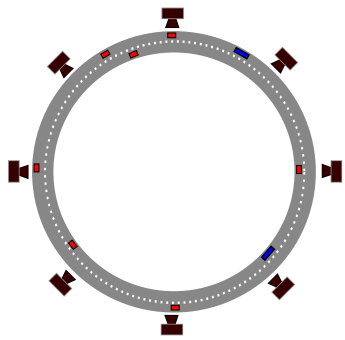


Fig. 2. Camera setup

2.1 Setup

The setup for the experimental study is reasonably simple. We used Treibers software [17] to simulate a ring road with an average speed and density sensor every kilometer. The range of each sensor was set to 350 m as is standard for traffic cameras [10]. A ring road was chosen, because it models a straight road with infinite history which is long enough to get congested traffic. Figure 2 illustrates the used setup. While this setup represents only an initial study, it allows us to illustrate a number of important issues. Learning data was collected by taking a snapshot of sensory information 4 times each second. To this snapshot we added the local average velocity as measured a fixed time t into the future. Different values of t were tried as will be discussed later.

2.2 Predicting numerical velocity

In a first step, we try to predict the future velocity using a regression algorithm, i.e., a learning algorithm that predicts a real value as an outcome. The learning algorithm we use was Gaussian processes with a radial basis function (RBF) kernel. Gaussian processes represent a strong baseline for regression [8]. Non-linear kernels, such as the RBF kernel, perform well when dealing with a large number of learning instances with a relatively low number of features [6, 3]. In short, Gaussian processes implement a non-parametric Bayesian technique. Bayesian regression techniques assume a prior distribution over the function hypothesis space (usually over the parameter vector defining the function) and calculate a posterior distribution using Bayes rule and the available learning data. Instead of assuming a prior over the parameter vectors, Gaussian processes assume a prior over the target function itself. We refer to [14] for a more elaborate discussion of the workings of Gaussian processes

As the covariance function required by Gaussian processes, we use a RBF kernel defined as $K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2)$ (illustrated as $K(x_i, x_j) =$

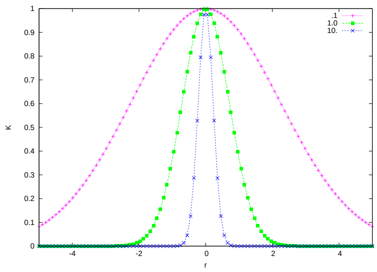


Fig. 3. Radial Basis Function

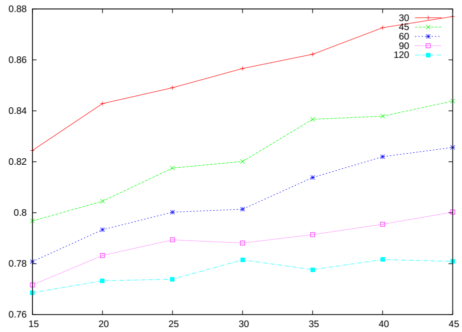


Fig. 4. Correlation: Change in time after drop

$exp(-\gamma r^2)$ in figure 3) [6]. The γ parameter controls the width of the kernel and thereby the amount of generalization used by the Gaussian processes. Higher γ values result in less generalisation. A small search over the γ of the RBF showed that γ equalling 1 gave better correlation results over other values. The correlation is an indication in how much two coeffients are related to each other. In this case how the predicted values and the actual values relate. All other learning experiments with Gaussian processes will use this value for the γ parameter.

Gaussian processes also allow specification of a measurement noise level. The way the sensors measure velocity as the average speed of passing traffic, gives rise to a natural measurement noise level, i.e., the standard deviation of the measured velocities.

Since we are trying to make predictions about the transition between free flow traffic and congested traffic, we need to collect data from both the A and the B region of Figure 1. Since we don't want to predict stop and go traffic, the amount of data from region B must be limited. Figure 4 shows how collecting more data after the transition influences the measured correlation. More data collected after the transition raises the prediction correlation. If data is collected for a time period longer then 30 seconds, the measured data originates from the B area which seems easier to predict. In this view, a collection time close to but below 30 seconds (the lowest studied prediction time) seems to be a good choice.

To test the need for a MAS approach in our sensor system, we first compared correlation results of using data from only the local sensor to data collected from a total of nine surrounding sensors (local, plus 4 sensors before and 4 sensors behind the target point). The training examples contain the velocity and the density from the participation sensors and the future velocity as the label to be predicted. Table 1 shows that (not surprisingly) using information from multiple sensors gives better correlation results than information from only the local sensor, which supports the need for multi-sensor cooperation in this type of prediction task. The gain of using multiple sensors ranges from small at a prediction time of 30 seconds, i.e. when predicting what the measured velocity will be 30 seconds in the future, to substantial when the prediction time increases.

Table 1. Correlation results

Pred. time	1 camera	9 cameras
30	0.81954	0.84855
60	0.68813	0.78329
120	0.43739	0.73997

Table 2. History

Pred. time	$\Delta t = 0$	$\Delta t \neq 0$
30	0.84855	0.85594
45	0.79709	0.82672
60	0.78329	0.80898
90	0.75415	0.79424
120	0.73997	0.78293

As measured velocity is sequential data where trends in the measurements might be highly informative we also investigated the predictive gain of past measurements or "sensory history". Using data from Δt time ago together with current measurements gives rise to better correlation results than using current information alone. Tests show that the exact value of Δt is of less importance. The correlation results are similar for Δt ranging from 5 to 30 (averaged in Table 2) and significantly better than the results of using no history ($\Delta t = 0$ in the table). The higher the prediction time, the bigger the gain from using a history.

Figure 5 shows a detailed plot of real future velocities (x-axis) versus predicted velocities (y-axis) for a prediction time of 60 seconds and $\Delta t = 10$. Data was collected until 25 seconds after the velocity drop. Area C at the top right shows a high correlation between real and predicted at high velocities. These are the recorded velocities before the transition, when traffic is still in free-flow. After the velocity suddenly drops, the model has more difficulties to make correct predictions. In area A and B, representing stationary and slow traffic respectively, future velocities are often predicted too high.

While using Gaussian processes delivers reasonable predictive results they do present a different difficulty. The learned model is large and consumes a large amount of memory and when making predictions, the model has to perform a lot of computations. This makes the model less suitable for agent purposes. Other regression techniques are expected to, at best, deliver only comparable results.

2.3 Predicting classified velocity

Since regression turns out to be hard, we study the use of classification for our congestion prediction problem. Classification is concerned with the prediction of nominal labels instead of numeric ones. In our problem, we consider three different types of traffic, and thus three possible labels: stationary, congested and normal traffic. These types are based upon the Dutch traffic information service. Stationary means speed below 7 m/s, congested is below 14 m/s, normal traffic is above 14 m/s. These are also the areas marked in figure 5.

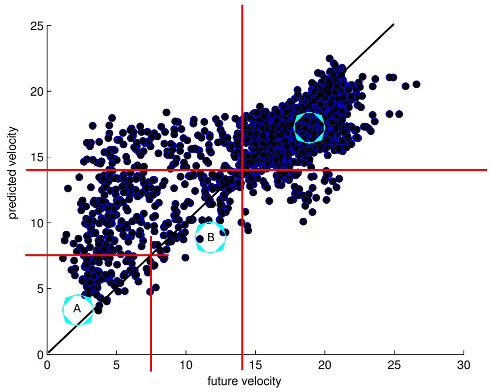


Fig. 5. Predicted velocity versus actual future velocity

The performance of the model will be examined by following quality evaluators: [21]

Accuracy is the overall probability that the model makes a correct prediction.

Recall is the percentage of actual positive instances that are predicted as positive, also called the *true positive rate*. In our traffic warning system, this will indicate the percentage of velocity drops that are predicted as such.

Precision is the percentage of positive predictions that are real positive instances. In our warning system, this gives an indication of the number of false alarms, i.e., precision = 1−(% of false alarms).

Since we are dealing with a 3-class prediction problem, we define a positive prediction to be a correctly classified instance, meaning e.g. '*normal*' has been classified as '*normal*' and a negative prediction as a wrongly classified prediction, meaning e.g. '*congested*' has been classified as either '*normal*' or '*slowdown*'.

Support Vector Machines In a first step we used a classification approach, very related to the Gaussian processes used for regression before: support vector machines (SVM). In short, SVM's classify examples using a linear decision boundary. In most cases, linear decision boundaries are not expressive enough to separate examples from different classes and therefor SVM's use a non-linear transformation on the input space to a new, high-dimensional, space. This way, a linear model in the new space can be a non-linear decision boundary in the original space [4]. The algorithm we used for training the SVM is Sequential Minimal Optimisation (SMO). SMO is very fast with linear kernels and reasonably fast with non-linear kernels. The memory footage grows linear with the training set, which means large training sets are possible [11]. For the input space transformation, we again used an RBF-kernel. To determine the best values for the parameters C (SMO) and γ (RBF) we again performed a grid search. $C = 10$

and $\gamma = 1$ gave the best results and all following experiments were preformed using these values.

Table 3 shows the influence of history on the evaluators. While recall is hardly influenced by adding history, it does make a difference for the accuracy and precision of the model. When no history is used ($\Delta t = 0$ in the Table), the results are somewhat worse than with the use of history. Since again the results for various values of $\Delta t \neq 0$ are similar, Table 3 shows averaged evaluation values.

Table 3. SVM: impact of the use of history

	$\Delta t = 0$			$\Delta t \neq 0$		
Pred. time	Accuracy	Recall	Precision	Accuracy	Recall	Precision
30	93.050	98.714	95.664	93.618	98.641	96.417
45	92.116	98.621	94.163	92.553	98.216	95.459
60	90.517	98.518	92.695	91.481	97.942	94.527
90	88.605	97.395	92.035	90.088	97.501	93.562
120	86.402	97.091	89.810	88.433	97.398	91.917

The model learned by SMO is small enough to be implemented into simple agents. The size ranges from 200 kB to 1 MB. One problem with using SVMs in an application as critical as traffic warning systems is that the learned model is pretty much a black box, and that it is next to impossible to interpret its decision strategies. More importantly however, the true positive rate on congested and slowdown are quite low. For example with a prediction time equal to 60 and history $\Delta t = 10$, the true positive rate for the prediction of normal traffic is 99.6 %, while the true positive rate for congested traffic is only 55.6 % and for slowdown only 17.6 %. While SMO is quite good at predicting normal traffic, it has significantly more difficulties with congested traffic and performs even worse on slowdown. Since these are exactly the conditions our warning system is looking for, this will not do.

Decision Trees To alleviate both problems indicated above, we tried decision trees on the same classification problem. Decision trees are a machine learning technique that employs a “Divide and Conquer” approach [12]. Decision Trees can easily be converted into rules and thus, the decisions they make can be interpreted and checked by a human, which is an advantage in critical applications such as traffic control.

We used the Weka variant of the C4.5 algorithm [13] as a decision tree learner. C4.5 has been a benchmark algorithm for a long time. In Weka, the algorithm is implemented as J48 [21]. Test results show that sensory history doesn’t influence the quality evaluators significantly. Table 4 shows the results for different prediction times. Not surprisingly, the predictive performance drops with increase prediction time.

Table 4. DT: results

Pred. time	Accuracy	Recall	Precision	Rules
30	91.648	97.625	95.368	68.235
45	90.129	96.346	94.532	87.520
60	88.928	95.755	93.796	100.24
90	86.316	94.435	92.283	124.52
120	83.970	91.195	89.112	144.82

The results are close to those obtained using the support vector machine. Using only a reasonably small number of rules (indicated in Table 4) the model performs comparable to the more computation intensive SVM model. However, with respect to the precision for the important classes, i.e. *congested* and *slowdown*, decision trees actually perform better. Using a prediction time equal to 60 and $\Delta_t = 10$ as as before, the true positive rate for '*normal*' is slightly worse than the results of SMO: 95.8% but those for '*congested*' traffic and '*slowdown*' actually improve significantly. Congested traffic has a true positive rate of 75.1% and slowdown of 53.4%. This means that fewer incorrect warnings will be given using decision trees than using the SVM model. An extract of the tree build for this test is shown below:

```
PrevVelocity4 <= 13.141108
|   PrevVelocity5 <= 17.9016
|   |   PrevVelocity3 <= 9.702755
|   |   |   PrevVelocity1 <= 15.582357
|   |   |   |   PrevVelocity1 <= 12.889484: slowdown
|   |   |   |   PrevVelocity1 > 12.889484: normal
|   |   |   PrevVelocity1 > 15.582357: congested
|   |   PrevVelocity3 > 9.702755
|   |   |   PrevVelocity4 <= 12.566312: congested
|   |   |   PrevVelocity4 > 12.566312
|   |   |   |   PrevDensity0 <= 85.714286
|   |   |   |   |   PrevDensity3 <= 45.714286: congested
|   |   |   |   |   PrevDensity3 > 45.714286: slowdown
...
```

3 Feasibility study

We want to make scalability and robustness important design criteria for our congestion warning system and therefor approach it as a distributed multi-agents system. In a first, simple design, each agent of the multi-agent system consists of a sensor, a repository and a decision maker. The sensor collects the necessary traffic information for prediction purposes and stores it into the agent's repository. The decision maker periodically checks the repository. When a certain density is encountered, the decision maker also collects information from other neighboring agents. The decision maker predicts the traffic speed state within the near future using the machine learned model, the data in the repository and the data received from the other agents. This decentralized approach ensures scalability. Adding more agents with the same machine learned model is easier than adjusting a central decision controller. There is no single point of failure. One failing agent will at most influence its eight neighboring agents.

3.1 Practical approach

We have evaluated the multi-agent system approach in a simulation setting [17]. In this simulation, the sensors which represent the sensory part of the agents are modeled as virtual camera's. The camera's are placed equidistant at 1 km from each other and their range is 350 m.

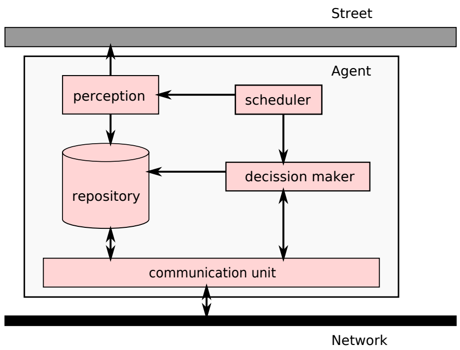


Fig. 6. Camera structure

The agent architecture can be found on Figure 6. The *perception* unit (the virtual camera) collects information from the street and calculates the density and intensity. This information is stored into the *repository*. The *scheduler* periodically stimulates the camera to take a snapshot, perform calculations and store the results. The *decision maker* checks the stored data in the repository. If the density is within the critical density, the decision maker will request the data from the neighboring agents. The collected data is then presented at the machine learned model, in order to predict the future velocity. The collection of data from the other agents, happens through the *communication unit*.

The repository stores and keeps history of the density and intensity measured by the perception unit. It also stores the neighboring agents. This neighbor information is useful for the decision maker. It ensures the data from the correct neighbors is collected. The neighboring agents are found using a bootstrap process. When the camera starts up, the camera broadcasts its existence and asks for other cameras within its range. The scheduler periodically stimulates the repository to check the correctness of the position of the relevant neighbors.

The number of agents is limited by the bandwidth of the communication network. If all agents would request specific neighbor information at the same time, it could overload the network. Alternatively, all information could be broadcasted and logged at each agent, but this would undo the local view of the agents.

3.2 Placing the model (and learner) inside the agent

The learned model from section 2 is part of the decision maker. The type of information requests to the neighboring agents will depend on the model, e.g., some models need historical data while others don't. For prediction purposes, Weka itself can be included. Switching models is easy in this approach, since it only requires the loading of a different model. The disadvantage is that Weka is somewhat big to be included inside an agent. Decision trees, for example, can easily be converted to rules and implemented inside the decision maker. Weka straightforwardly supports the generation of rules (java code) from the learned decision trees. The advantage of including Weka as part of the agent is that the prediction model could be adapted online as discussed in future work.

4 Related work

Related work on traffic and congestion prediction is quite extensive and a large scale overview is well beyond the extend of this paper. A lot of existing work focuses on long term predictions, such as the work by Yasdi [22], that predicts the density on weekly, daily and hourly basis through the use of a neural network. The goal of the predictions is use them to reroute traffic to keep the density on the roads below the critical point and avoid congestion formation completely. The decisions are made at a central point. In contrast, we focus on short term predictions of the average velocity on a short road segment in a distributed system.

Abdulhai et al. [1] try to predict the density within minimum 30 seconds an maximum 15 minutes, also using a neural network. The used densities all lie within free flow boundaries. The results show that the farther into the future one tries to predict, the more the neural network tends to predict the average density. The data is collected from 9 loop detector stations and the prediction is done centralized.

A very different approach for the prediction of congested traffic is an ant based system [2]. Every vehicle leaves a trail of pheromones, based on the traffic information. In this setup, vehicles themselves are able to predict congestion from the information of preceding cars. The density and velocity aren't measured directly but predicted through the accumulation of virtual pheromones.

Huisken et al. [7] compare time series analysis (ARMA) and neural networks (MLF) performance in predicting congestion through density estimation on a relatively short-term time scale (5 - 15 minutes) and conclude that neural networks gave the best results.

Taylor et al [16] try to predict the volume and the occupancy of traffic using a multilayer perception network. The prediction happens one minute in advance on data from 6am until 9am during weekday's. The performance of prediction is tested using the mean squared error. The neural network seems to perform quite well.

In contrast to all related work discussed above, we predict actual short term velocity drops when traffic is at the critical density point, instead of estimating density related values.

5 Conclusions

In this paper, we evaluated the use of machine learning techniques for the prediction of the average velocity of traffic at the critical density point. The machine learning models were trained using data from multiple cameras. The models were evaluated on their correctness and interpretability. Decision Trees gave the best results in both.

The warning system itself was designed as a multi-agent system. The agents contain the learned model, check their environment using a camera view for critical densities and when traffic resides at the critical density point, start to collect data from other cameras and use the learned model for short term velocity or congestion predictions. The model will trigger an alarm when congestion eminent and warn upcoming drivers that traffic is slowing down. These alarms and models can hopefully be used to lessen the occurrence of stop and go traffic.

In the current setup, failure of an agent influences the predictive capacities of the neighboring agents. To increase robustness, probability nodes instead of decision nodes can be used inside the decision tree when decision critical data is unavailable. Instead of deciding which path to take based on the (unavailable) data, the probability of the missing condition can be used to combine the decisions made by the child branches. This would mean that failure of one agent does not imply that eight other agents are also unable to make any predictions.

The proposed structure should be tested on real traffic data instead of simulation data. Considering that the simulator implements realistic driving behaviors, we expect at least some of our conclusion to carry to the real world.

Another interesting next step is letting the agents change their environment by warning drivers or implementing some speed limitations. Changing the environment will automatically cause the learned models to become wrong, raising the need for online learning. By integrating the machine learning algorithm within the agents, the model can be optimized online to account for local road specifics. Utgoff [19] presented an algorithm for incremental decision tree learning. Since agents collect data at the critical density point for prediction no extra communication overhead is caused to collect online training data.

References

1. Baher Abdulhai, Himanshu Porwal, and Will Recker. Short-term freeway traffic flow prediction using genetically optimized time-delay-based neural networks. In *Transportation Research Board*, Washington D.C, 1999.
2. Yasushi Ando, Yoshiaki Fukazawa, Osamu Masutani, Hirotoshi Iwasaki, and Shinichi Honiden. Performance of pheromone model for predicting traffic congestion. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 73–80, New York, NY, USA, 2006. ACM Press.
3. C. Campbell. An introduction to kernel methods. In Robert J. Howlett, Lakhmi C. Jain, and Janusz Kacprzyk, editors, *Radial basis function networks 1: recent developments in theory and applications*, pages 155–192. Physica Verlag Rudolf Liebing KG, Vienna, Austria, 2001.

4. N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2000.
5. Josefa Z. Hernández, Sascha Ossowski, and Ana García-Serrano. Multiagent architectures for intelligent traffic management systems. *Transportation Research Part C: Emerging Technologies*, 10(5-6):473 – 506, 2002.
6. C. W. Hsu, C. C. Chang, and C. J. Lin. A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 2003.
7. G. Huisken and A. Coffa. Short-term congestion prediction: comparing time series with neural networks. In *Road Transport Information and Control, 2000. Tenth International Conference on (Conf. Publ. No. 472)*, pages 66–69, 2000.
8. D. J. C. MacKay. Introduction to Gaussian processes. In C. M. Bishop, editor, *Neural Networks and Machine Learning*, NATO ASI Series, pages 133–166. Kluwer Academic Press, 1998.
9. VerkeersInformatieDienst Nederland. http://www.verkeersinformatiedienst.nl/fileproblematiek.html.
10. Traficon NV. Traficon camera specification: http://www.traficon.be/pagenode.jsp?id=7&type=ProductCategory.
11. John C. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods: support vector learning*, pages 185–208, Cambridge, MA, USA, 1999. MIT Press.
12. J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.
13. J. Ross Quinlan. *C4.5: programs for machine learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
14. Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning).* The MIT Press, 2005.
15. Maerivoet S. and De Moor B. Traffic flow theory. Technical report, ESAT-SCD (SISTA) K.U.Leuven, Leuven, Belgium, July 2005.
16. C. Taylor and D. Meldrum. Freeway traffic data prediction using neural networks. In *Vehicle Navigation and Information Systems Conference, 1995. Proceedings. In conjunction with the Pacific Rim TransTech Conference. 6th International VNIS. 'A Ride into the Future'*, pages 225–230, Jul-2Aug 1995.
17. Martin Treiber. http://www.vwi.tu-dresden.de/ treiber/MicroApplet/.
18. Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62:1805, 2000.
19. Paul E. Utgoff. Incremental induction of decision trees. *Machine Learning*, 4:161–186, 1989.
20. Verkeerscentrum Vlaanderen, Departement Leefmilieu en Infrastructuur, AdministratieWegen en Verkeer, Vuurkruisenplein 20, 2020 Antwerpen *MINDAT — Databank ruwe verkeersdata Vlaams snelwegennet — 2001 en 2003*, February 2003.
21. Ian Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, 2 edition, 2005.
22. R. Yasdi. Prediction of road traffic using a neural network approach. *Neural Computing & Applications*, 8(2):135–142, May 1999.

Intelligent Transportation Systems: a Ubiquitous Perspective

Lúcio Sanchez Passos, Rosaldo J. F. Rossetti

Department of Informatics Engineering
Artificial Intelligence and Computer Science Laboratory (LIACC)
Faculty of Engineering, University of Porto (FEUP)
Rua Dr. Roberto Frias, S/N • 4200-465 Porto • Portugal
{ext08101, rossetti}@fe.up.pt

Abstract. The concept of intelligent transportation has been devised about a couple of decades ago and still presents many challenging issues to be addressed so as it can be implemented to its full potential. In this paper we emphasise characteristics such as being user-centred and service oriented to support an important facet of future urban transportation: being ubiquitous. Instead of seeing ubiquitous transportation as a completely different paradigm, we discuss on the characteristics that actually turn ITS inherently ubiquitous.

Keywords: ubiquitous computing, ambient intelligence, intelligent transportation systems.

1 Introduction

With the growth of population in major urban areas and the accelerated increase in number of cars, traffic is becoming generically chaotic. The problem of congestions, differently from what many might think, not only affects the day-to-day life of citizens but also has a great impact on business and economic activities. These issues therefore generate less income, affecting the sustainable growth of cities throughout the world.

Considering current problems of traffic management, control and planning, especially fearing the consequences of their medium and long term effects, both practitioners and the scientific communities have strived to tackle congestion in large urban networks. Research has been carried out basically towards the design and specification of future transport solutions featuring autonomy, putting the user in the centre of all concerns and largely oriented to services. Such efforts were eventually to culminate in the emergence of the concept of Intelligent Transportation Systems (ITS), basically relying on a distributed and advanced communication infrastructure favouring interaction in virtually all level, from users to services, vehicles to vehicles, vehicles to infrastructure, and so forth. Interoperability and integration are crucial in this scenario. More futuristic though is the perspective in which users will play rather a passive role and be taken off the whole process, which will ultimately be managed by the system only, to which autonomous driving is expected to be an important ingredient. Although this view may seem quite hypothetical, it is capable to stimulate and foster much advancement in a wide spectrum of multidisciplinary fields, from engineering and computer science to sociology and urban planning and design.

In this work we basically recall many of the different aspects involved in the original definition of ITS and identify potential applications of the ubiquitous computing concept (sometimes hereafter referred to as ubicomp). Instead of defining a novel perspective for what has been recently coined Ubiquitous Transportation Systems (UTS), we prefer to see ITS from a ubiquitous perspective, emphasising those characteristics that actually turn ITS into ubiquitous systems. Therefore, ITS is inherently ubiquitous! Besides ubiquity, pervasiveness, ambient awareness and intelligence are equally addressed as complementary and conceptually related technologies. Many issues and interesting question marks rise in this context, stimulating different streams for further research, some of which are also discussed.

Following this brief motivation of the topic, the remaining of this paper is organised as follows. In the next section, we discuss on requirements for future urban transport so as it is possible for us to better understand why the whole bunch of technologies presented later on are important. Those are then presented and briefly discussed in the following section. Ubiquitous transportation is then presented in the fourth section, finally followed by some remarks, conclusions drawn and suggestions for future work.

2 Future Urban Transport

Persons and goods are transported for centuries, which have turned transport into an essential part of any economy based on trade. Basically, transport systems are intended to take both people and other goods from certain points to one or more destinations, accounting for users' needs, efficiency, and low cost as well. These systems have become rather complex and extremely large, being geographically and functionally distributed, both in that respecting structure and management.

Contemporary transportation systems have experienced three major revolutions throughout their existence. The first one came with the introduction of electromagnetic communication, allowing for a network of information that enabled greater integration of the system to exchange information more quickly and efficiently. The second great revolution started with the advent of digital systems delivering lower cost services. Then computers started playing an imperative role in transportation, improving efficiency of traffic control and coordination, as well as transport planning. Centralised traffic coordination started as well playing a major role in network design and management. Albeit centralised decision systems can be very efficient in theory, their performance is quite dependent on scale. In other words, they work proportionally to the number of unities processed by the system, becoming many times very slow and therefore lacking efficiency. Such a limitation poses serious problems and present undesirable consequences. As transportation systems are becoming very large, both in terms of structure and dimension, the whole process of acquiring information from all sources, processing the essential data and providing adequate responses timely is rather a very arduous task. This is especially more complex due to real-time constraints and the presence of heterogeneous participating entities.

Finally, the third great revolution in transport became evident with the advent of what has been coined Intelligent Transportation Systems (ITS). Now the user is a central aspect of transportation systems, forcing architectures to become adaptable and accessible by different means so as to meet different requirements and a wide range of purposes. Integration is then crucial! This novel scenario has been motivating and challenging practitioners and the scientific community and suggested a completely new decentralised perspective of processes. On the other hand, discussions are still fostered by current ambitions to the Future Urban Transport systems (FUT), even more conscious in terms of environment, accessibility, equality, security, and sustainability of resources. Some of the main features of today's intelligent transportation, to mention a few, are as follows.

Automated computation is an important requirement of ITS. Future transport systems must make decisions automatically, analysing input information and acting accordingly, triggering coordinated actions to improve system performance. Demand for *flexibility* and *freedom of choice* is another important aspect on the user's side. The current lack of flexibility in transportation systems limits their potential to users, especially in that concerning personalised services, which is a major target for criticism by many users. ITS then should be open to flexibility, different options and diver choices, as well as personalised services. Also, *accuracy* is another important aspect, i.e. precise and up-to-date information must be delivered on a timely basis. Indeed, in transport systems that are generally classified as dynamic, errors and other failure situations are proportional to the accuracy of system reactions and responses to requested services. Therefore, latency should be reduced through distributed architectures for reducing response time. As transportation systems are greatly dependent on the network topology and other characteristics, *intelligent infrastructures* become fundamental. New communication technologies, including mobile, wireless and *ad-hoc* networks are improving infrastructures a great deal, enabling it to become an active and interactive part of the system. This is especially important for the implementation of the ambient transportation intelligence concept. A *distributed architecture*, accounting for asynchronous, control algorithms, coordination and management autonomous elements is undoubtedly one of the major currently researched areas of ITS. There are several requirements that must be satisfied, from user-centred to service-based functionalities, turning intelligent transportation into a complex, heterogeneous and intricate artificial society. Current research already considers that ITS architecture must explore distributed algorithms using exogenous information from various sources, making greater use of parallelism and asynchronous capacities of pro-active entities. These characteristics are essential to ITS, and suggest an enormous application potential of Distributed Artificial Intelligence (DAI) based solutions, especially Multi-Agent Systems (MAS) [1].

Accounting for the characteristics aforementioned, multi-agent systems have been greatly studied and applied in developing the concept of future transportation. Especially in the past couple of decades, a great deal of research work has been carried out in this area. Despite ITS geographical and functional distributed dependency, other concepts are also being devised and improved in such as huge research live laboratory, namely pervasive and ubiquitous computing, ambient intelligence and service-oriented architectures, among many others. These concepts ultimately foster advance in many other areas so as to address issues that are flourishing around as ITS and FUT become a reality. Some of such issues include the identification and definition of heterogeneous autonomous entities, definition of the asynchronous nature of entities, as well as which aspect make them concurrent, definition of the adequate communication infrastructures, validation and proof of algorithms and models correctness, robustness, performance and stability. Of course this is far from being an exhaustive list and many other issues are still to arise in such a fascinating study area.

3 Related Concepts and Technologies

By the end of the last century, we have witnessed the emergence of a new vision for the role of electronic devices in people's lives. Such a vision was firstly pointed out by Mark Weiser [2] and basically relied on the use of computer resources and provision of information and services to people whenever and wherever they were wished and needed. This novel concept was widely accepted, and during the past decade researchers applied it to built devices of various sizes, which started to become part of our daily lives. There was an enormous growth of these types of devices such as hand-held personal digital assistants (PDA), digital tablets, laptops, and wall-sized electronic whiteboards for a wide range of purposes. Indeed, coupling these devices with the ability of communicating with people and the surrounding environment in diverse ways actually contributed to the definition of the Ubiquitous Computing, or ubicomp for short. Nonetheless, this concept is not limited to communication or interface. Ubicomp is intended to go further beyond and was designed to specify the adequate infrastructure to provide people with a "24-by-7" information services, meaning users can access service providers 24 hours a day, 7 days a week. This innovative perspective opens up a wide range of different interfaces yet unexplored, introducing a concept called everyday computing, which suggests that over time and the presence of ubiquitous environments, the system itself will create a routine of activities for people, without those who have program them.

There are several fields of study involved in the concept of ubiquitous computing, ranging from the ability to miniaturise processing devices for discrete use in our daily activities, as well as the development of applications that can function as networks of groups for each individual. All these study initiatives have the common goal to develop, explore and extend the concept of ubiquitous computing to its maximum potential. Today we are just starting to understand the implications of continuous immersion in computation. The future will hold much more than constant availability of tools to assist with traditional, computer-based tasks; in the near future, we are going to "wear" computers that are going to track our actions and health and allow us better perform our activities. Not surprisingly, one of the application potential of ubicomp is within intelligent transportation systems of the future. Both practitioners and the scientific community agree that ITS can profit a lot from ubicomp. Next we discuss a bit further on the different concepts and technologies that are likely to take part in such forthcoming ubiquitous transportation world.

As for *human-machine interfaces*, ubiquitous computing has brought into evidence a new concept and way of interaction between humans and machines, sometimes referred to as “off the desktop” interaction environment [9]. This view assumes that interactions between humans and machines will much less be based on traditional interfaces, such as the keyboard, mouse, and display paradigm, and will foster interactions that are more natural to humans, such as speech, gestures and the use of objects of easy manipulation. These natural actions can and should be used as explicit or implicit input to ubiquitous systems. This new concept is being studied for a long time already, and advances have been reported to the point that some of these natural interfaces have been implemented in commercial systems, sometimes replacing the conventional interfaces. One of the advantages of these interfaces is the easy learning and intuitive use, just by imitating the natural ways of human interactions, thus enabling an easy transition and adaptation of users.

The *context-aware computing* is derived from ubicomp in the sense it is one of the keys to ubiquitous environments, so that entities can interact with the environment in a comprehensive manner, obtaining information and adapting their behaviour to the current situation accordingly. According to [3], the importance of context to ubicomp is best explained through the so-called “five W’s”, as it is further discussed below.

Context-awareness to work properly should know *who* is gathered together in the same room. This is important because the exchange of people with the conditions of the environment changes as well. Knowing *what* is being carried out by people is also imperative for the system to perceiving and interpreting human activities which are needed for the interaction with the environment to be completed. The *where* component of contextual awareness has actually been explored more than the other components in many different ways. Of particular interest, however, is coupling notion of “where” with other contextual information, such as “when.” Time is definitely a very important variable in such a perspective, so the *when* component is imperative as well, especially because it may qualify information with regard to time and relationships between situations that occurred in different instants, allowing the system to produce better and timely responses. Even more challenging than perceiving “what” someone does, is understanding *why* a person is doing that and the reasons that actually triggered the action.

Service-oriented architectures (SOA) is a new in information systems and can be basically defined as a group of services that communicate with each other to better serve the end-user. To understand such a concept well, we need to define service, which can be seen as a specific implementation of a function and is able to access both data and resources [4]. So, the goal of SOA is to create an *ad-hoc* topology of application that users are allowed to string together pieces of the functionality, all this constructed on the base application provided by the system.

Basically, the potentials of SOA have been analysed according to two points of view. Firstly, as for technical aspects, SOA-based architectures have many advantages, but some advantages are equally identified. One major advantage is that services are relatively open and accessible by any user or other services, as long as they are able to understand each other. This makes SOA pretty interesting to support services over the Internet, for instance. Among disadvantages, however, security and lack of testing and validated service models are important issues still to be addressed. Secondly, from the economic perspective, the possibilities are great as SOA-based architectures provide the basis for new business opportunities and other similar investments, as well as for the customisation of current services making them adaptable to a new emergent dynamic demand [5].

In the last few decades, with the advent of the digital era and a change of orientation in systems architecture, users are brought to a central spot and services are devised to be autonomous and pro-active. In consequence, new paradigm called *Ambient Intelligence* (AmI) has emerged.

AmI is defined as the ability of the environment to sense, adapt and respond to actions of persons and objects that inhabit its vicinity [10, 11]. To accomplish that in full, AmI's philosophy requires some important characteristics. Devices and services must be *embedded*, in other words, they should be an inherent part to the environment itself. Indeed, in recent year, we have witnessed a trend towards the miniaturisation of electronic systems so as they can be easily embedded in other devices and spread out all over the environment. Also, systems must be *context-aware*, meaning they must be able to understand the presence of individuals, their interactions and objects that are around, and with that perception interpret their actions and needs. As users are central to this concept, services must be *personalised*, which feature assumes that the system can build up to converge with the immediate needs of a user. Such latter characteristic also implies that the system must be *adaptive*, in the sense it can adapt itself in response to users' behaviours. Finally, AmI is intrinsically *anticipatory*, using its context-awareness to anticipate users' preferences and intentions and to provide the adequate environment for users to enhance their expected outcomes.

Ubiquitous and pervasive computing are two concepts generally dealt with indistinguishably, and they refer to various computer systems that can be part of a whole system (or be the system itself) that allows connection and interaction between various devices without the direct knowledge of humans who are using them. The two concepts were initially suggested in [2], nonetheless they present small and subtle differences. Indeed, according to the definition of Oxford's Dictionary [6], ubiquitous means something which is present everywhere at anytime, and pervasive is an adjective denoting something that spreads itself on something. Thus, we can intuitively interpret these concepts differently in that a ubiquitous system is passive and expects users' initiative to access services and information in it. On the contrary, pervasive systems are those that autonomously interfere while interacting with users, being pro-active and adaptive most of the time [12, 13].

4 Ubiquitous Transportation

Several significant events that marked the history of humanity have been identified over time and are reflected in all activities of human societies. Of course, these include the evolution of transport means that have played a decisive role in suppressing frontiers between countries and bridging continents in what we call today a globalised world. At the end of the twentieth century, there were many revolutions in a very short period time, mainly due to electronics, information systems, and communication infrastructure. In the 60's, main-frames began to be used, which were known by their large size and difficulty in maintaining the system operational. One great achievement was, undoubtedly, the advent of the transistors age, with the increasing miniaturisation of electronic components and consequently the information systems and interfaces as well. With this new trend in computer infrastructure, new technologies started to appear in a very quick and increasing pace.

One such technology that revolutionized our daily lives was the personal computer, first introduced by IBM in the early 80’s. Also, the popularisation of domestic computer desktops has fostered the industry yet considerably more. One of several possibilities that PC has brought was to make it easier to implement digital system relatively powerful to cope with a bunch of different tasks and quite affordable. One typical example of practical use of PC is today’s traffic control systems, to coordinate traffic lights in a more efficient way. Nonetheless, mobile communication is also reaching drivers and passengers in their vehicles, increasing even more the potential application of computer systems to transportation. Indeed, with the increase use of computers in a scale never before imagined, communication technologies and infrastructure also benefited of much advance in recent years. Urban scenarios, for instance, are witnessing the advance of ad-hoc vehicular networks (coined VANET) that make interoperation among transport components (both travellers and infrastructure) even more interactive and efficient. The Internet is now present in this mobile world, fostering applications of unforeseeable potential, not only to transportation management and control systems enabling real-time monitoring of traffic activities, but as well to traveller either on an individual or collective basis.

Not surprisingly, the beginning of the new century brought with it the breaking of paradigms and the creation of new concepts, changing the main orientation of management systems, which typically were geared to operations. Today, however, we can identify a new focus on the clients (end-users), as well as their needs and well being. This has driven information system design to take into account user’s satisfaction as an imperative requirement that, on the other hand, demanded adaptive computer architectures. The concept of ubiquitous computing also shares such a view.

This was actually the very first ambition of what was called Intelligent Transportation Systems, in mid 90’s. Therefore, ITS is also ubiquitous in nature and, for some reason, this very characteristics of today’s transportation systems have remained latent for quite a long time. Quite recently, however, the scientific community is recalling former objectives in ITS and, with today’s available technology, Ubiquitous Transportation Systems (UTS) are becoming a reality. It is equally desirable to notice that UTS is not actually a new concept of transportation, but rather is intrinsic to ITS. Most works basically report on the influences of mobile computing and communication on current available transportation systems. Nevertheless, the application potential of UTS is far more promising. Figure 1 is merely a superficial illustration of what the concept of UTS would be. As shown in the figure, the concepts of ITS and ubicomp were created almost at the same time and also the research development in both fields has followed quite the same pace over the last years. So, with the co-existence of both ITS and ubicomp, all elements were present and favourable to the advent of ubiquitous transportation systems.

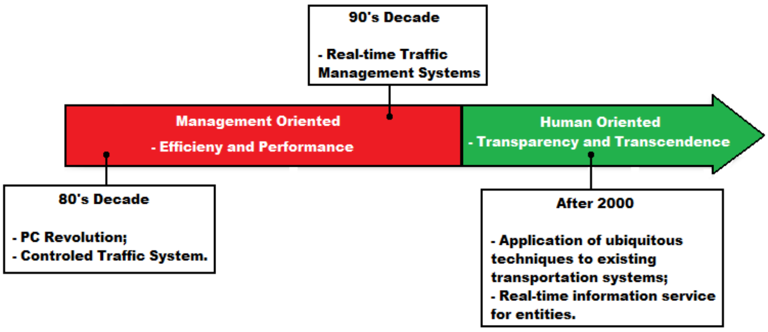


Fig. 1. Historical development of transportation systems

To illustrate this trend, Figure 2 depicts only the intersection areas in the framework of physical structure with the application of technologies for vehicle detection, data processing, communication and GIS/GPS, of ITS, as well as technologies related to wireless and mobile communication. However, if we apply the concept ubiquitous computing to the heart of ITS, there will be other things needing to be changed so that ITS is fully turned into UTS. One effect of such a natural evolution is the adaptation of services in ITS, which are currently process-based. In UTS, however, service-oriented architecture will be fully dynamic adaptable to the users' needs. Thus, it is quite acceptable that there exist lots of aspects that must be studied for the perfect fusion of such two complimentary concepts. If they can be effectively coupled together, then a very powerful transportation system will emerge, with many possibilities of services, not only to transport companies and the market but also to the individual user.

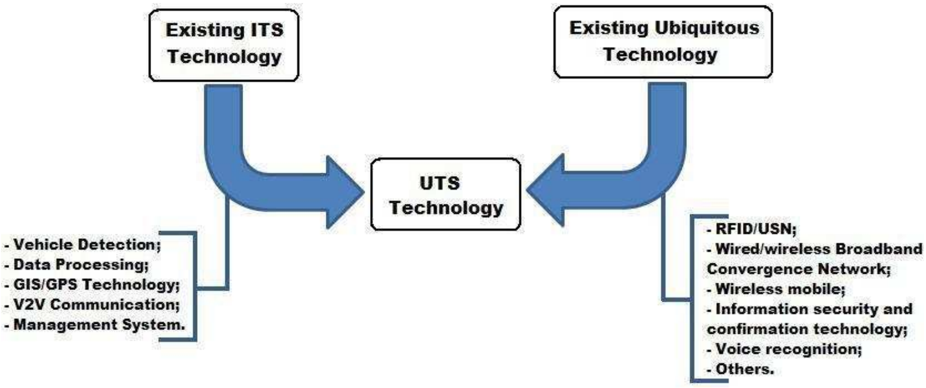


Fig. 2. Explaining how ITS and ubicomp are related to each other

As initially stated in this paper, our goal is to view and analyse intelligent transportation systems from a ubiquitous computing point of view rather than proposing or working on a novel concept of ubiquitous transportation as implied in [7]. Indeed, origins of ITS suggest an infrastructure on the basis of all technological aspects discussed in the previous section, meaning users, services and infrastructure can now interact as peers. Therefore, intelligent transportation systems are ubiquitous in nature. For the sake of simplicity, however, we refer to such perspective as ubiquitous transportation or UTS for short. The terminology is borrowed from [8], but instead of explaining a novel concept we are rather referring to an aspect inherent to ITS.

Thus, in this section we establish a relationship between the concepts of ubiquitous computing and intelligent transportation. To understand such a correlation, we need to look back at the primordiums of ITS, when new guidelines were first drawn for the future transportation systems. Initially, it should be noted that forthcoming transportation systems would be drastically based on distributed systems and thus ITS and ubicomp shared common characteristics.

Another important aspect endorsed by ITS is the full integration of users in all processes within the system, via static or mobile devices, prior to a certain journey or en-route, inside vehicles. For such an interaction to be effective and efficient, there should be a fully immersion of devices supported by an appropriate communication infrastructure, which would makes ITS naturally ubiquitous. This confluence, however, has been seen for some time as the emergence of new transport paradigm, namely the ubiquitous transportation systems (UTS). Nevertheless, there is no sharp boundaries between them two and both ITS and UTS can be dealt with indistinguishably.

UTS can be better understood in the light of some ubicomp's prominent features, which according to [8] has four key aspects. First, services must be present everywhere and anytime. Instead of carrying a device wherever someone goes, the device is either physically or virtually available everywhere to anyone. Second, it is not the device and its capabilities that actually matter but the environment. The real value of ubicomp lies in the fact that it is a comprehensive environment rather than a collection of services supplied by individual devices. Third, users are not conscious of devices being used. Thus, using a service does not require conscious awareness of the device, which allows the user to concentrate on the task at hand. Finally, services must be TPO-based (time, place, and occasion). This way, available services should match the prevailing situation and needs of users.

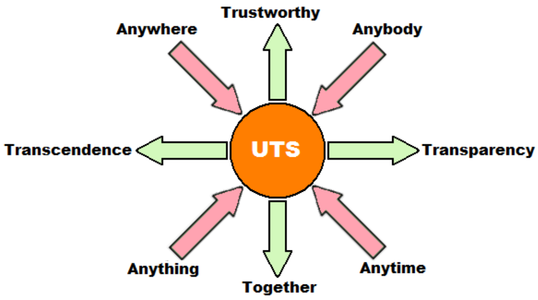


Fig. 3. UTS properties

Therefore, according to the features discussed above [8] it is possible to identify eight important properties of UTS, as depicted in Figure 3. Inward arrows illustrate UTS accessibility properties, such as anything, anybody, anytime, and anywhere. On the other hand, outward arrows illustrate properties UTS should provide to users, such as transparency, togetherness, transcendence, and trustworthiness. These properties are also known as the 4A and the 4T of ubiquitous computing, and will be discussed a bit further in the context of intelligent transportation as follows.

According to ITS original principles, transportation services should be accessible to *anybody* or *anything*, which might be any entity within the system, such as other services that carry out their tasks on autonomous bases. The system must also be accessible *anytime* and *anywhere*, meaning entities can find a service whenever and wherever it is deemed necessary.

In that concerning ITS environment, services quality and integration, transportation systems should be *trustworthy*, i.e. maintain the privacy of users' data at the same time accuracy and timing of services are preserved. Also, systems must be *transparent* to the users, meaning people should not need to be aware of the presence of components while using them. Boundaries between services should also be attenuated to the point users experiment a collective intelligence, with various components working *together* on a collaborative basis. As for *transcendence*, it represents the ability of the system to sense, diagnose and respond not only to the humanly recognisable transportation environment, but also to the humanly unrecognisable one, usually coined as extended reality.

Albeit ubiquitous and pervasive computing are sometimes seen as synonyms, as new technologies and studies are quickly emerging autonomy is even more a reality turning ubiquitous transportation into pervasive systems in the sense previously mentioned. Rather than being simply available, pervasive transportation systems (PTS) play a more active role, influencing transit and seeking better solutions to quickly respond to situations that require an overview of the environment in an autonomous way.

Ambient intelligence (AmI) is top autonomy level of intelligent transportation systems, encompassing both ubiquitous and pervasive properties. Thus, AmI implies a fully autonomous system, managing and controlling vehicles (and travellers, in a general sense) by sensing all parts of the transit environment. The path to this scenario is still a long way ahead, requiring multidisciplinary development, not restrict to devices but also in understanding human behaviour and applying advanced Artificial Intelligence techniques. Undoubtedly, this field emerges with great interest, challenging both scientific and practitioners' communities. To better understand how these concepts have evolved and interacted over time, some basic scenarios are discussed below, as follows.

As for ubiquitous transportation, it is perhaps the very next step towards the implementation of the original concept of ITS. Indeed, today's technology already allows us to experience new properties and services that enhance traditional transportation. For instance, mobile communication allows a user to access and consult traffic conditions while she is still on the way to the car. Also, pedestrians may use multi-modal travel planners to compose better itineraries, encompassing buses, trams, suburban trains, walking and even cabs booked in advance. This is performed on demand, and the system is able to offer the best alternatives according to specific requirements, such as cost, expected travel time, or presence of given points of interest (PoI), e.g., drugstores, restaurants and gas stations.

In pervasive transportation systems, scenarios suggest a much more pro-active set of abilities. Some examples are already in use, whereas much of the benefits expected from such systems are still dreams to come true. For the former, we can mention situations in which emergency vehicles, e.g. ambulances, need to get to a particular place in a congested urban area. In these situations, control systems, aware of the emergency, collaborate by setting all traffic lights across the due itinerary to green, so as to avoid excessive waste of time at junctions. In the latter case, information systems are expected to play a more pro-active role, anticipating users' needs and improving their daily schedule. For instance, as soon as a driver gets in the vehicle, the system might check the driver's agenda and prepare the best itinerary accordingly, while setting the appropriate radio stations accounting for the user's preferences. Many mobile information media would be integrated to deliver the necessary piece of information no matter the user is onboard, in the vehicle, or walking on the street, or at her desk at work.

Transportation's ambient intelligence, on the other hand, would certainly profit from a vast and effective sensors network. Albeit infrastructure and vehicular *ad-hoc* wireless networks (coined VANET) have been evolving steadily over the last few years, a fully intelligent transportation environment still poses many issues to be addressed. Such kind of system, for instance, would be able to handle accidents more effectively. In the event of an accident and if the environment itself were able to perceive and characterise it (e.g. identifying the degree and number of vehicles involved), the adequate means to address the situation would be deployed immediately without the need of one single person calling the emergency services, while re-routing other vehicles to avoid the accident site. Another potential application in urban public transport would include the automatic identification of demand. Let us suppose that for some reason the number of passengers waiting a bus service at a specific stop increased considerably. If the environment, featured with proper sensors and communication abilities, were able to identify the reason for such a transient increase in demand as well as to quantify the exceeding number of passengers, alternative transportation means or contingence services, by despatching additional vehicles into the active fleet, could be delivered accordingly. These scenarios might seem quite futuristic at first glance; nonetheless, they are quite feasible with today's technology and will very likely make part of everyone's daily lives in the near future.

5 Conclusions

Future urban transportation, regardless whether technology is available, motivates much research and development in a wide range of multidisciplinary fields. Safer, greener, sustainable and accessible-to-all transportation has never been closer to reality as today. Indeed, ITS as originally devised about a couple of decades ago put the user in the central spot, surrounded by a wide variety of services, some of which are already currently available. However, to become effectively ubiquitous, some efforts are needed to further develop concepts such as ambient intelligence, pervasiveness and autonomy of services. Whereas first discussions have suggested the creation of a ubiquitous transportation paradigm, a deeper analysis of original ITS ideas actually corroborate our perspective that ITS is in fact inherently ubiquitous. Technology is becoming available day after day and it is just a matter of time for the full potential of ITS-based solutions to be recognised and effectively deployed. The next steps to follow include the specification of fully multi-agent based architecture of a ubiquitous transportation system, using the Gaia methodology [14], as well as a transportation ontology to support services specification.

References

1. Schleiffer, R.: Intelligent agents in traffic and transportation. In: Transportation Research 10C, no.5-6 (2002)
2. Weiser, M.: The Computer for the 21st Century. In: IEEE Pervasive Computing, pp.19-25 (2002)
3. Abowd, G.D., Mynatt, E.D.: Charting Past, Present, and Future Research in Ubiquitous Computing. In: ACM Transac. Computer-Human Interaction, vol.7, no.1, pp.29-58 (2000)
4. Lublinsky, B.: Defining SOA as an Architectural Style: Align Your Business Model with Technology. In: IBM Systems Journal (2007)
5. Krämer, B.J., Halang, W.A. (Eds.): Contributions to Ubiquitous Computing. SCI, vol.42. Springer, Heidelberg (2007)
6. Compact Oxford English Dictionary, http://www.askoxford.com/ (Last accessed on June 30, 2009)
7. Weiser, M.: The Coming Age of Calm Technology. In: PowerGrid Journal, vol.1.01, pp.1-8 (1996)
8. Lee, E., Ryu, K., Paik, I.: A Concept for Ubiquitous Transportation Systems and Related Development Methodology. In: International IEEE Conference on Intelligent Transportation Systems, pp.37-42 (2008)
9. Weiser, M., Gold, R., Brown, J.S.: The World is not a Desktop. In: ACM Interactions (1993)
10. Aarts, E.: Ambient Intelligence: A Multimedia Perspective. In: IEEE Multimedia, vol.1, pp. 12-19 (2004)
11. Boekhorst, F.: Ambient Intelligence, the Next Paradigm for Consumer Electronics: How will it Affect Silicon? In: IEEE International Solid-State Circuits Conference (ISSCC), vol. 2, pp.16-20 (2002)
12. Miao, Z., Yuan, B.: Discussion on Pervasive Computing Paradigm. In: IEEE Region 10 TENCON, pp.1-6 (Nov., 2005)
13. Kurkovsky, S.: Pervasive Computing: Past, Present and Future. In: 5th International Conference on Information and Communications Technology, pp.65-71 (2007)
14. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing multiagent systems: The Gaia methodology. In: ACM Transac. Software Engineering and Methodology, vol.12, no.3 pp.317-370 (2003)

Evaluating Policies for Reservation-based Intersection Control*

Matteo Vasirani and Sascha Ossowski

Centre for Intelligent Information Technology
University Rey Juan Carlos, Madrid, Spain
{matteo.vasirani,sascha.ossowski}@urjc.es

Abstract. Autonomous vehicle navigation in urban road networks will be possible in the future, making traveling yet another activity that does not need the human intervention. In this paper we empirically evaluate different policies to manage a reservation-based intersection, an infrastructure facility proposed by Dresner and Stone to regulate the transit of vehicles through intersections. We evaluate two different scenarios, a single intersection and a network of intersections, comparing the original policy employed by Dresner and Stone in their work with 4 policies, inspired by the adversarial queueing theory (AQT).

Keywords: autonomous vehicle, reservation-based intersection, control policies, adversarial queueing theory

1 Introduction

The recent and promising advances in artificial intelligence and, particularly, in multiagent system technology suggest that autonomous vehicle navigation in urban road networks will be possible in the future. Safe and efficient urban automated guided vehicles (AGV) could make driving yet another activity that does not need the human intervention. At the present time, cars can be equipped with features such as cruise control [3] and autonomous steering [5]. Furthermore, there exist small-scale systems of AGVs, for example in factory transport systems. If this trend holds, one day fully autonomous vehicles will populate our road networks. In this case, given that the system will have a variable (and possibly huge) number of vehicles and an open infrastructure, central control such in today's AGV systems [4] will be impossible.

To this respect, Dresner and Stone [2] introduced a minimally centralized infrastructure facility that allows for the control of intersections. In their model, an intersection is regulated by an intelligent agent (intersection manager) that assigns reservations of space slots inside the intersection to each automated guided vehicle intending to transit through the intersection. Such an approach

* This research was partially supported by the Spanish Ministry of Science and Education through projects "THOMAS" (TIN2006-14630-C03-02) and "AT" (CONSOLIDER CSD2007-0022, INGENIO 2010).

has shown, in a simulated environment, several advantages, because it may drastically reduce delays with respect to traffic light and it makes possible the use of fine graned, vehicle-centric, control policies.

In the Dresner and Stone's original work, the intersection manager applies a simple first-come-first-served policy, evaluating the reservation requests in the same order they are received by the intersection manager. Nevertheless, allowing the intersection manager to evaluate a set of requests at the same time, it may make more informed decisions so as to optimize the intersection throughput. In this work, we make an empirical evaluation of different policies, inspired by the adversarial queueing theory (AQT) [1], comparing them with the first-come-first-served policy proposed by Dresner and Stone.

This paper is structured as follows: in section 2 we detail the Dresner and Stone's protocol that rules the interaction between the vehicles that want to cross the intersection and the control facility; section 3 introduces the different policies that can be applied for the processing of the reservation requests, evaluating them in section 4; we discuss the experimental results in section 5; finally we conclude in section 6.

2 Protocol

The reservation-based system proposed in [2] assumes the existence of two different kind of agents: *intersection managers* and *driver agents*. The intersection manager controls an intersection and schedules the transit of each vehicle. The driver agent is the entity that autonomously operates the vehicle.

Each driver agent, when approaching the intersection, contacts the intersection manager, sending a REQUEST message. The message contains the vehicle's ID, the arrival time, the arrival speed, the lane occupied by the vehicle in the road link before the intersection and the type of turn. The intersection manager simulates the vehicle's transit through the intersection and informs the driver agent whether its request has conflicts with the already confirmed reservations or not. If the transit does not have conflicts with the confirmed reservations, the intersection manager replies with a CONFIRMATION message, which implies that the driver agent implicitly accepts the reservation parameters. On the other hand, if the transit is not feasible, the intersection manager replies with a REJECTION message.

The intersection manager also uses the *reservation distance* as a criterion for filtering out reservation requests that could generate deadlock situations [2]. The reservation distance is approximated as $v_a \cdot (t_a - t)$, where v_a is the arrival speed of the vehicle, t_a is the arrival time of the vehicle (both contained in the REQUEST message), and t is the current time. For each lane i, the intersection manager has a variable d_i, initialized to ∞. For each reservation request r in lane i, the intersection manager computes the reservation distance, $d(r)$. If $d(r) > d_i$, r is rejected. If, on the other hand, $d(r) \leq d_i$, r is processed as normal. If r is rejected after being processed as normal, $d_i \leftarrow min(\ d_i, d(r)\ )$. Otherwise, $d_i \leftarrow \infty$. Although the use of the reservation distance does not guarantee that a vehicle only gets a reservation if all the vehicles in front of it already have a reservation, it makes it more probable.

3 Reservation-based policies

In the intersection control mechanism proposed originally by Dresner and Stone, the intersection manager processes the incoming requests with a first-come-first-served policy (FCFS). This means that if two vehicles send requests that require the same space-time in the intersection, the vehicle that sends the request first will obtain the reservation. This policy in extreme case could result being quite inefficient. Consider the case in which a set of n vehicles, v_1, v_2, ..., v_n, such that v_1's request has conflicts with every other vehicle, but that v_2, ..., v_n do not have conflicts with one another. If v_1 sends its request first, its reservation will be granted and all other vehicles' requests will be rejected. On the other hand, if it sends its request last, the other $n-1$ vehicles will have their requests confirmed, whilst only v_1 will have to wait.

Still, other policies for processing the requests, inspired by the research on adversarial queueing theory (AQT), can be employed. The AQT [1] model has been used in the latest years to study the stability and performance of packet-switched networks. In this model, the arrival of packets to the network (i.e., the traffic pattern) is controlled by an adversary that defines, for each packet, the place and time in which the packet joins the system. Each node in the network has a reception buffer for every incoming edge, an output queue for every outgoing edge, and a packet dispatcher that dispatches each incoming packet into the corresponding output queue (or removed, if this is the final node of the packet), using a specific policy. Under these assumptions, the stability of the network system is studied, where stability is the property that at any time the maximum number of packets present in the system is bounded by a constant that may depend on system parameters.

The AQT model and the request processing of the reservation-based control mechanism share some similarities. In the same way a packet dispatcher decides which packet from the reception buffer will be dispatched to the corresponding output queue, so an intersection manager may decide in which order to process a set ot reservation requests, assigning priorities to requests accordingly with its scheduling policy. Taking inspiration from the AQT model, we compared the FCFS policy with 4 universally stable policies, namely *longest-in-system* (LIS), *shortest-in-system* (SIS), *farthest-to-go* (FTG) and *nearest-to-source* (NTS). The LIS policy gives priority to the request of the vehicle which earliest joined the system. The SIS policy gives priority to the request of the vehicle which latest joined the system. The FTG policy gives priority to the request of the vehicle which still has to traverse the longest path until reaching its destination. The NTS policy gives priority to the he request of the vehicle which is closest to its origin, i.e., which has traversed the less portion of its whole route.

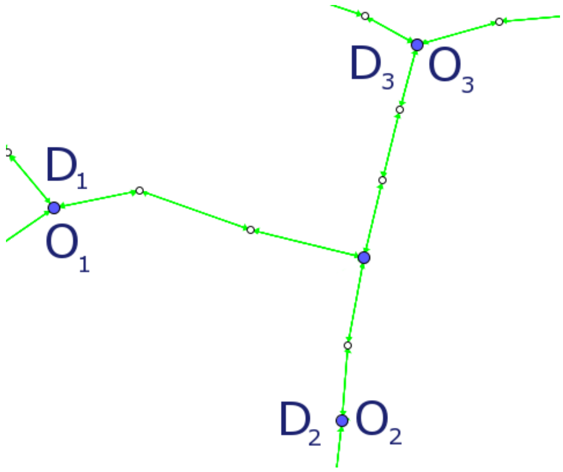


Fig. 1. Scenario 1: single intersection

4 Experimental results

In order to implement the 4 policies, the REQUEST message sent to the intersection manager must contain the necessary additional information: the time stamp when the vehicle joined the system, i.e., when it started to travel, an identifier of the origin location, and an identifier of the destination location.

We evaluated two different scenarios: i) a single intersection scenario (section 4.1) and ii) a network of intersections scenario (section 4.2).

As baseline, we used an intersection regulated by traffic lights with 3 phases (one per incoming road link) of 30 seconds each.

The experiments have been done using a custom, discrete-time, mesoscopic-microscopic simulator. This simulator models the traffic flow on the roads at mesoscopic level [7], where the dynamic of a vehicle is governed by the average traffic density on the link it traverses rather than the behaviour of other vehicles in the immediate neighbourhood as in microscopic models.

Since the mesoscopic model does not offer the necessary level of detail to model a reservation-based intersection, when a vehicle enters an intersection its dynamic switches into a microscopic, cellular-based, simulator, whose update rules follow the Nagel-Schreckenberg [6] model. The cell size is set to 5 meters, and for simplicity we assume that the vehicles cross the intersection at a constant speed, so that any additional tuning of parameters, such as slowdown probability or acceleration/deceleration factors, is not necessary.

4.1 Scenario 1: single intersection

In this scenario we simulate a single intersection with three road links of three lanes each, that connect the origin set $\mathcal{O} = \{O_1, O_2, O_3\}$ with the destination

Table 1. Traffic demands for scenario 1

	λ								
	1	5	10	15	20	25	30	35	40
total spawned vehicles	29	136	285	438	633	716	832	1063	1183

set $\mathcal{D} = \{D_1, D_2, D_3\}$ (see figure 1). When a vehicle is spawned, we assign it an origin $o_i \in \mathcal{O}$ and a destination $d_j \in \mathcal{D}$, with $i \neq j$. We simulate different traffic demands by varying the expected number of vehicles (λ) that, for every O-D pair, are spawned in an interval of 60 seconds. We spawned vehicles for a total time of 10 minutes. Table 1 summarizes the global traffic demands for different values of λ.

The metrics we used to evaluate the performance of the different policies were the *average delay* (sec.), the *average queue time* (sec.) and the *average rejected requests* (% of sent requests).

The average delay measures the increase in travel time due to the presence of the intersection (be it reservation-based or regulated by traffic lights). It is measured running two types of simulation: in the first one, the intersection is regulated by the control mechanism under evaluation and the vehicles must obey the norms that the control mechanism imposes; in the second one, the vehicles travel as if they could transit through the intersection unhindered. The difference between the two average travel times gives us the average delay. Formally:

$$\frac{\displaystyle\sum_{i \in \mathcal{V}} (t_f^i - t_0^i) \;-\; \sum_{i \in \mathcal{V}} (\widehat{t}_f^i - \widehat{t}_0^i)}{N}$$

where $\mathcal{V}$ is the set of vehicles, N is the number of vehicles, t_f^i and t_0^i are respectively the time when vehicle i arrives at its destination and when it leaves its origin in the simulation with the intersection regulated by a control mechanism, while $\widehat{t}_f^i$ and $\widehat{t}_0^i$ are respectively the time when vehicle i arrives at its destination and when it leaves its origin if we make the vehicles transit through the intersection unhindered.

The average queue time is the time spent by the vehicles at the intersection queue. Formally:

$$\frac{\displaystyle\sum_{i \in \mathcal{V}} (t_{q_f}^i - t_{q_0}^i)}{N}$$

where $t_{q_f}^i$ is the time when vehicle i leaves the queue of the intersection and $t_{q_0}^i$ is the time when it enters the queue of the intersection.

The average rejected requests is a metric that applies only to the reservation-based policies, and is measured as the ratio between the rejected requests and the sent requests. Formally:

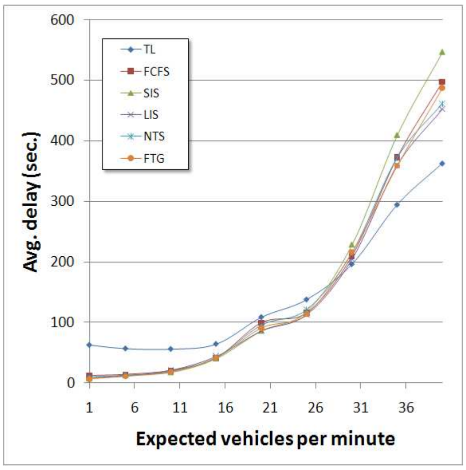


Fig. 2. Average delay

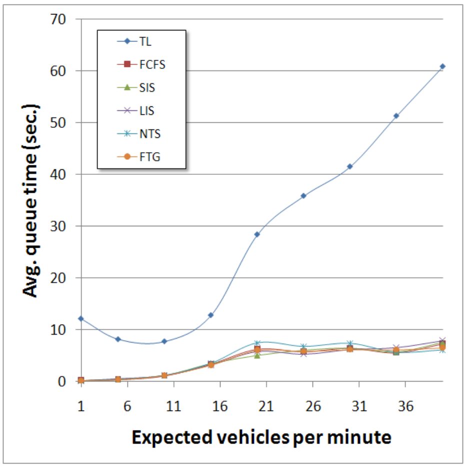


Fig. 3. Average time in queue

$$\frac{\sum_{i \in \mathcal{V}} r^i / s^i}{N}$$

where r^i is the number of rejected requests of vehicle i and s^i is the number of requests sent by vehicle i.

Figure 2 plots the average delay for different traffic demands ($\lambda \in [1, 40]$). When the traffic demand is low ($\lambda \in [1, 15]$), all the reservation-based policies (FCFS, LIS, SIS, FTG, NTS) tend to behave in the same manner, reducing the average delay of about the 65% with respect to traffic lights (TL). Still, when the traffic demand reach a critical value (around $\lambda = 30$), the reservation-based intersection performs worse than the traffic light intersection, with an increase of the average delay between 24% and 50%. Among the reservation-based policies, with high traffic densities the LIS policy is the best one (the increase of the average delay with respect to traffic lights is "only" 24.65%), while the SIS policy is the worst one (with an increase of 50.57%). The reservation-based intersection outperforms a traffic light intersection particularly when the traffic demand is below a certain threshold, because few requests are rejected and the majority of vehicles can transit through the intersection without waiting for a green phase such in traffic light intersections.

Although the experiments have been performed with a custom simulator that is different from that used in [2], the above results seem consistent with the results that we can find in [2]. In fact, in the original work by Dresner and Stone, the reservation-based intersection with FCFS policy outperforms the traffic-light intersection when the traffic density is in the range $[0, 1]$ $vehicles/sec$, while the authors didn't gave any results for higher traffic demands. In our experiments, the reservation-based intersection starts to perform worse than the traffic-light

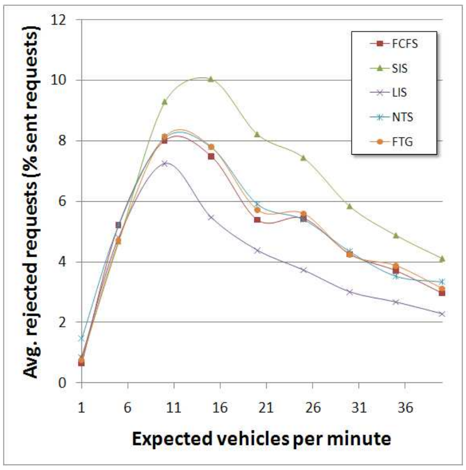


Fig. 4. Average rejected requests

when the expected number of vehicles (λ) is 30, or, given table 1, when the traffic demand is 1.38 *vehicles/sec*, beyond the maximum value evaluated in [2].

Figure 3 plots the average time spent at the intersection queue. Here are noticeable two very distinct dynamics. With a traffic light intersection, the time spent by the vehicles at the intersection queue grows linearly and constantly with the traffic demand. On the other hand, with a reservation-based intersection, the queue time settles around about 7 seconds, independently of the policy in use. This plot gives us an idea of the vehicle's behaviour when approaching the two different types of intersection. If the intersection is regulated by traffic lights, the vehicle proceeds at the speed permitted by the traffic conditions and, once it reaches the intersection, if the traffic light is red it enters the intersection queue. In this way, the more the vehicles approaching the intersection, the longer the waiting time at the intersection queue. With a reservation-based intersection the dynamic of the vehicle approaching the intersection is different. If the vehicle holds a valid reservation, it maintains its speed because it has safety guarantees about its transit through the intersection. On the other hand, if it does not have such reservation, it reduces its speed for safety reasons, and it keeps requesting a reservation to the intersection manager. Thus the collective behaviour is a slower, smoother, traffic flow through the intersection, with few time spent stuck at the intersection.

Finally, we evaluated the reservation-based policies in terms of average rejected requests (as a percentage of the sent requests). Here with rejected request we refer to a request that cannot be granted due to *conflicts* with the already confirmed reservations. Figure 4 plots the results for the different traffic demands under evaluation. With low traffic density, all the policies perform quite similarly, and the rejected requests increase linearly with the number of vehicles

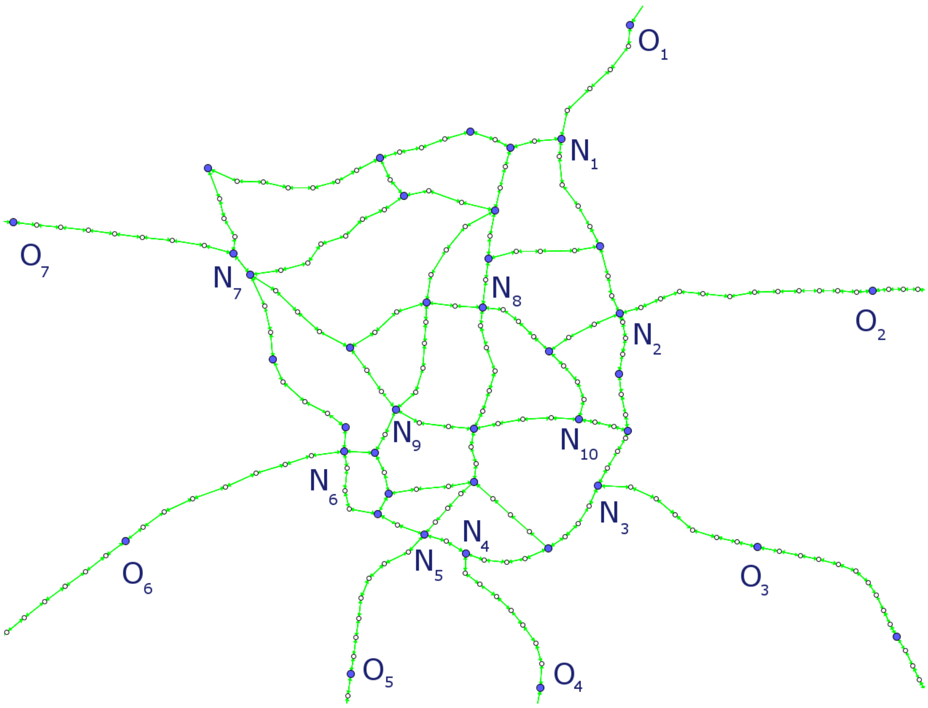


Fig. 5. Scenario 2: network of intersections

approaching the intersection. When the traffic demand reaches a critical point (around $\lambda = 15$), the rejected requests tend to decrease with the traffic demand. The reason of this counterintuitive dynamic is the effect of the *reservation distance*. As the number of rejected requests increases, the reservation distance tends to become smaller, because it is updated with the distance of the closest vehicle whose request has been rejected. With high traffic density, the effect of the reservation distance becomes predominant, filtering out the majority of the reservation requests and processing only those of the nearest vehicles. Since less requests are processed, less conflicts are detected, so that the average rejected requests decrease.

4.2 Scenario 2: network of intersections

In this scenario we simulate an entire network of intersections (figure 5). We defined several locations that serve as origins and destinations for the traffic demand. The vehicles that commute from/to locations $\in \mathcal{O} = \{O_1, O_2, O_3, O_4, O_5, O_6, O_7\}$ form the traffic under evaluation. The vehicles that commute from/to locations $\in \mathcal{N} = \{N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8, N_9, N_{10}\}$ serve to add "noise" and to populate the network more realistically.

We aimed at recreating a typical morning peak, with 2 different traffic demands, namely *low* (3986 vehicles) and *high* (11601 vehicles).

The metrics we used to evaluate the performance of the different policies for a given O-D pair (o, d), with $o, d \in \mathcal{O}$, was the average delay per crossed intersection (sec/intersection):

$$\frac{\sum_{(o,d)\in\mathcal{OD}} avgDelay(o,d)/intersections(o,d)}{||\mathcal{OD}||}$$

where $\mathcal{OD}$ is the set of all the O-D pairs (o, d), $avgDelay(o, d)$ is the average delay of the given O-D pair, $intersections(o, d)$ is the number of intersections of the route from o to d, and $||\mathcal{OD}||$ is the number of O-D pairs in the set $\mathcal{OD}$.

Low traffic demand. Table 2 shows the average delay per crossed intersection and the relative delay (made 100 the best policy). At first sight, it seems confirmed that, as for scenario 1, the traffic light intersection is the intersection policy that causes more delay for low traffic demand. This confirms the results of scenario 1: the reservation-based intersection with a good policy takes advantage of low traffic demands, reducing drastically the delay with respect to a traffic lights intersection. With traffic lights, vehicles stop at the intersection even if there are no vehicles on the road link with the green phase. On the other hand, with a reservation-based intersection, few vehicles mean few reservations that are rejected, so that the transit through the intersection speeds up.

The LIS policy is the policy that causes less delay, 6.68 seconds per intersection, about the 21% less than the FCFS. The traffic light intersection is the worst one: a vehicle is delayed 38 seconds when it transits through each intersection it finds on its route, with respect to the $7-8$ seconds of a reservation-based intersection.

Still, it is noticeable how the traffic light intersection has a low standard deviation with respect to its average delay. This is a desirable property from the point of view of the quality of service of the system, which in this way does not penalize nor favour excessively a specific O-D pair, introducing the same delay for every trip through the network.

Table 2. Average delay per crossed intersection (low traffic demand)

	Average delay (sec/intersection)	**stdev.**	**Relative delay**
TL	38.00	8.39	568.77
FCFS	8.13	7.43	121.64
LIS	6.68	5.63	100.00
SIS	7.63	5.68	114.24
FTG	7.12	5.58	106.56
NTS	7.65	5.35	114.44

High traffic demand. Finally, we evaluated the reservation-based policies and the traffic light intersection using a high traffic demand, with a total of 11601 vehicles traveling through the road network. To assess which policy is the best one in reducing delays, we rely again on the average delay per crossed intersection (table 3). When the traffic demand is high, the traffic light intersection (TL) turns out to be the best policy, with 194.38 seconds of delay per crossed intersection. The reservation-based policies perform all slightly worse than the traffic light intersection, with about 230 seconds of delay per crossed intersection. It is interesting to notice that also in this case the TL has the lowest standard deviation, almost the half of the standard deviation of the best performing reservation-based policy. This is a hint that as the demand increases, the performance of a reservation-based intersection becomes more volatile: in some part of the network it could speed up the transit through an intersection, while in other parts it may slow down the transit even more than a traffic light, which behaviour is, on the other hand, more predictable and stable.

Table 3. Average delay per crossed intersection (high traffic demand)

	Average delay (sec/intersection)	**stdev.**	**Relative delay**
TL	194.38	95.72	100.00
FCFS	226.48	154.06	116.51
LIS	229.69	162.04	118.16
SIS	230.33	161.36	118.49
FTG	233.63	159.14	120.19
NTS	225.08	160.67	115.79

5 Discussion

In the experiments described in section 4.1 and 4.2 we evaluated the performance of different policies that a reservation-based intersection manager can employ to process the reservation requests that it receives. One (FCFS) is the policy used in the original work by Dresner and Stone, the others (SIS, LIS, NTS, FTG) are policies inspired by the theory of adversarial queueing. As baseline, we used an intersection regulated by traffic lights (TL) with n phases (one per incoming road link) of 30 seconds each. From the experimental results of scenario 1 (single intersection) and scenario 2 (network of intersections) we can conclude the following:

The reservation-based intersection is suited for low-load situations. As seen in the experimental results, the reservation-based intersection reduces drastically the average delay when the traffic demand is low. The vehicles are able to transit through the intersection unhindered, they almost do not stop at the intersections, and the allocation capacity of the intersection is maximized. Still, when the demand increases, the throughput of the reservation-based intersection becomes closer to that of a traffic light intersection, and for high traffic demand it performs even worse than a traffic light intersection. This is because a reservation-based intersection is less robust than a traffic light and its performance is very sensitive to the traffic demand. With many vehicles approaching the intersection, the correct arrival time at the intersection becomes harder to estimate and more sensitive to traffic variations, so that many confirmed requests are cancelled by the vehicles, thus reducing the intersection throughput.

The reservation-based intersection produces a smoother traffic flow. From the analysis done for scenario 1, we can conclude that a reservation-based intersection affects the pattern of the traffic flow. Although the average delay increases with the traffic density, the reservation-based intersection reduces drastically the time spent by the vehicles at the intersection queue, especially in worst case situations: the queue time with high traffic demand ($\lambda \in [20, 40]$) is reduced up to a 80% with the reservation-based intersection. These two metrics suggest that a reservation-based intersection producing a slower, smoother, flow through the intersection, with the vehicles spending less time stuck at the intersection.

FCFS is the simplest policy but it is quite efficient. In spite of its simple behaviour, the FCFS does not perform much worse than the other, more complex, policies. Thus, it can be considered the best choice, since it needs less information: SIS and LIS need to know when the vehicle joined the system, while NTS and FTG need the information about where the vehicles is coming from and where is going to. This fact suggests also that probably we cannot expect great improvements of the efficiency of a single intersection using even more sophisticated policies than those evaluated in this paper. Still, we remark the performnce of the LIS policy. As as shown in the above experiments, it performs better than the FCFS in reducing delays and queue time, especially in low-load situations, when the reservation-based intersection outperforms the traffic light intersection. Furthermore it can be implemented with no much more effort with respect to the FCFS and the extra information it needs cannot be manipulated by the driver agent. For example, when the driver agent starts up, it has no time stamp, so the very first reservation that it will request won't have any time stamp. The intersection manager, detecting that the request has no time stamp, could manage the request with a default time stamp (e.g., the actual time) and then it could "stamp" the driver agent with the actual time so that, for the rest of its trip, it will provide a good approximation of the time when it joined the system.

6 Conclusions

Autonomous vehicle navigation in urban road networks will be possible in the future. To this respect, Dresner and Stone introduced an infrastructure facility that allows reservation-based control of intersection to regulate the transit of vehicles. In this paper we evaluated different policies, inspired by the adversarial queueing theory, that can be employed by this kind of facility, comparing them with the first-come-first-served policy used in the original work.

We showed that the reservation-based intersection is best suited for low-load situations, compared with a traffic light intersection, generating a smoother traffic flow with less time spent at the intersection queue. Furthermore, by empirical evaluation, we showed that, albeit simple, the first-come-first-served (FCFS) policy is quite efficient. Still, some improvements can be obtained using the longest-in-system (LIS) policy, which reduces delays and queue time more effectively than the FCFS, especially in low-load situations.

References

1. M. Blesa, D. Calzada, A. Fernández, L. López, A. Martínez, A. Santos, and M. Serna. Adversarial queueing model for continuous network dynamics. In *Proceeding of Mathematical Foundations of Computer Science*, volume 3618 of *Lecture Notes in Computer Science*, pages 144–155. Springer, 2005.
2. K. Dresner and P. Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31:591–656, 2008.
3. P. Ioannou and C. C. Chien. Autonomous intelligent cruise control. *IEEE Transactions on Vehicular Technology*, 42(4):657–672, 1993.
4. C. W. Kim and J. M. A. Tanchoco. Conflict-free shortest-time bidirectional agv routeing. *International Journal of Production Research*, 29(12):2377–2391, 1991.
5. B. Krogh and C. Thorpe. Integrated path planning and dynamic steering control for autonomous vehicles. *Proceedings of the IEEE International Conference on Robotics and Automation*, 3:1664–1669, Apr 1986.
6. K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I France 2*, pages 2221–2229, 1992.
7. T. Schwerdtfeger. Dynemo: A model for the simulation of traffic flow in motorway networks. In *Proceedings of the 9th International Symposium on Transportation and Traffic Theory*, pages 65–87. VNU Science Press, 1984.

Chapter 2

ALEA - Artificial Life and Evolutionary Algorithms

Genetic Algorithms using Populations based on Multisets

António Manso[1], Luís Correia[1]

[1] LabMAg - Laboratório de Modelação de Agentes
Faculdade de Ciências da Universidade de Lisboa
Edifício C6, Piso 3, Campo Grande,
1749-016 Lisboa - Portugal
manso@ipt.pt , Luis.Correia@di.fc.ul.pt

Abstract. The traditional representation of the populations used in evolutionary algorithms raises two types of problems: the loss of genetic diversity during the evolutionary process and evaluation of redundant individuals. In [11, 12] the authors propose a new formal model (PLATO) for multiset representation of individuals and their populations which applied to heuristic algorithms, minimizes the problems identified above. This paper presents a computational representation of populations based in multisets, and the adaptation of the genetic algorithm to deal with this type of representation, the Multiset Genetic Algorithm (MGA). A new operator called rescaling is developed as well as a metric to measure genetic diversity. The standard genetic algorithm is applied to some types of problems using the standard and the new type of populations and empirical results shows the genetic diversity is increased and the number of individuals evaluated is decreased as expected.

Keywords: Multiset, Genetic Algorithms, Genetic Diversity, Population-Based Algorithms

1 Introduction

Evolutionary algorithms (EA) are stochastic methods that mimic the process of biological evolution and have become popular tools for search, optimization, machine learning and design problems [1]. When applying these algorithms to complex problems where the search space is complex the most frequent difficulty is the premature convergence of the algorithm due to the lack of genetic variety. Individuals with better fitness propagate their genes in successive generations leading to a premature convergence [2]. Genetic diversity is essential to the evolutionary process, and without it the EA stops at least good solutions especially in problems that have more than one solution (multi-modal optimization) or involve simultaneous objectives, multi-objective optimization.

Several strategies have been proposed to maintain genetic diversity at different stages in the evolutionary process: Selection with weak pressure to fitness like uniform or tournament selection methods; Reproduction promoting the attraction between elements of different individuals or using other reproductive intelligent operators; Mutation with several adaptive operators; Substitution: promoting the replacement of individuals with similar genotypes but better fitness [2, 3, 4, 5, 6, 7, 21]. EA have revealed potential to reach good balances in order to find a good set of solutions with limited computational power. The maintenance of many good solutions in parallel is desirable and good algorithms have been developed for the clearing, clustering, crowding, fitness sharing and speciation [19].

The efficacy of the EA is directly related to the size of the population. However, in some applications, the evaluation of a large number of individuals is computationally expensive and delays the evolutionary process. To minimize this problem we find in literature some solutions: saving the objective value of individuals in the databases, estimating the ability of individuals using similar systems and prediction of fitness [9,10].

All strategies above are improvements to the original algorithm, which require the redefinition of the operators and all have an increased computational cost. In [11, 12], the authors introduce a new formal model using the concept of multiset for the representation of populations. This model allows the resolution of two problems listed above: maintaining the genetic diversity and avoiding superfluous evaluations. In [15] the authors show a software prototype, based in these ideas and applied to Genetic Algorithms (GA). This paper describes the adaptation of the GA to a multiset based population, also introducing a new operator called rescaling. This variant is called Multiset Genetic Algorithm (MGA). Empirical results show that the MGA produces higher genetic diversity and smaller number of evaluations than the same GA using simple population (SP). The diversity was calculated by two measures of genetic diversity.

2 Multisets and Multipopulations

A multiset is a collection of elements which may appear repeated. The number of times an element occurs in a multiset is called its multiplicity. The cardinality of a multiset is the sum of the multiplicities of its elements [13]. We can define a multiset as a set of ordered pairs <n,e> where n is the cardinality of the element e. In this definition the set {a,a,a,b,b,c} has an equivalent representation in multiset {<3,a>,<2,b>,<1,c>}.

EAs are based on populations of individuals in the form of collections. In the traditional representation it is common to have repeated individuals within the population (Table 1). This can be efficiently represented by a multiset (Table 2). Multipopulations (MP) are populations where the individuals are represented by ordered pairs <n,g> where n is the number of copies of the genome g. To this ordered pair we call multiindividual (MI), and a MP is a set of MIs with number of copies greater than zero. The set of g in a MP is called the support of the MP. In case of Table 2 we have a support set with four elements (notice that all of them have different genotypes).

Populations are dynamic collections where individuals are inserted, removed and searched. To implement MP we must redefine these three operations in the traditional data structure to support MI.

- **Search** - an MP is a set of individuals, grouped in MI, which can be indexed. The index of an individual is defined as a range from the sum of the multiplicity of all previous MI to this value added of the own multiplicity (Table 3). Searching an individual is searching one index in the population. This index is very important to maintain the equivalence between the MPs and the normal populations (SP) necessary for performance comparisons.

Table 1. Population with 8 individuals of the problem MaxOnes.

Genotype	Fitness
11111110	7
11111110	7
11111110	7
11110000	4
11110000	4
00001110	3
00001110	3
00000010	1

Table 2. Multi-Population equivalent of the population represented in Table 1.

Copies	Genotype	Fitness
3	11111110	7
2	11110000	4
2	00001110	3
1	00000010	1

Table 3. Indexing Multiindividuals in the Multipopulation of Table 2.

Copies	Genotype	Fitness	Indexes
3	11111110	7	0,1,2
2	11110000	4	3,4
2	00001110	3	5,6
1	00000010	1	7

- **Insert** - when an individual is appended to the MP, the first operation is to check if there is already a MI with the same genotype; if true this operation increases the number of copies of the MI to incorporate the new individual, if false the individual is inserted in the population and the number of copies is one (Table 4).

Table 4. Append the individual “11111110” and “00000000” in the MP of the Table 3.

Copies	Genotype	Fitness	Indexes
4	11111110	7	0,1,2,3
2	11110000	4	4,5
2	00001110	3	6,7
1	00000010	1	8
1	00000000	0	9

- **Remove** – the elimination of an individual of one MP is done decrementing the number of copies of the MI. If the number of copies decays to zero, the MI is removed from the data structure (Table 5).

Table 5. Remove the individual at the index 7 (“00001110”) and the individual “00000010” (index 8) in the MP of the table 4.

Copies	Genotype	Fitness	Indexes
4	11111110	7	0,1,2,3
2	11110000	4	4,5
1	00001110	3	6
1	00000000	0	7

Appending and removing individuals in the MP produces one search for the individual genotype into the MP, incrementing the computational complexity to the evolutionary process. To minimize this issue the data structures to support MP must be efficient in search, not only to select individuals, but also to add and remove them from the MP. Random access is another important aspect for the implementation of some genetic operators.

Genetic Diversity

Genetic diversity is based on the Hamming distance between individual genotypes. This measure between two bit strings returns the number of bits that are different in the genotype representations. By applying the same principle to the entire population we can calculate the Hamming distance of an individual in relation to the population as the sum of Hamming distances between it and all the other elements of the population. The genetic diversity of the population is given by the sum of Hamming distances of the individuals that compose it. We define a measure called the genetic diversity consisting on a normalized form of the Hamming distance for binary strings. It can be efficiently computed as

$$genetic\_diversity = \frac{4}{n^2 l}\sum_{i=0}^{l} k_i\left(n-k_i\right), \quad (1)$$

where l is the number of bits of each individual, k_i the number of ones of the allele i in the population and n the population size. The maximum diversity is 1.0 when the percentage of alleles for each gene is 50%, and the minimum diversity is 0.0 when all genes have the same value. This measure produces similar results to *pop_diversity* [20] but is computationally more efficient and is normalized to interval [0, 1].

Different alleles

The number of different alleles in the population is another diversity measure, which calculates the amount of genes that are not fixed in the population [14]. This metric can measure the ability of a population to explore the search space through the recombination operator.

The maximum diversity is obtained when the population has different alleles in all the genes and the minimum value is obtained when all the genes have the same value overall the population, which means all individuals are identical.

Equation (2) shows the normalized form of the measure

$$different\_allels = \frac{\sum_{i=0}^{l} a_i}{l}, \quad (2)$$

where l is the number of bits of each individual and a_i is zero if all the alleles of the gene i have the same value and one otherwise.

3 Adaptation of the Genetic Algorithm to Multi-Populations

The simple GA (SGA), when using the MP and the traditional operators, benefits automatically of the reduced number of evaluations and increased genetic diversity. The number of evaluations is decreased because they can evaluate many individuals at once (one multiindividual), and the genetic diversity of the population is increased because all the MI in the support set have different genotypes.

The individuals with best fitness increase their number of copies in successive generations. If this number of copies is not controlled the benefits of the MP may be decreased due to the huge number of copies of the best individuals. We can adapt the common EA operators to this new representation, but a lighter intervention can be considered by just introducing a new operator. This will allow us to compare results of the SGA with the Multiset Genetic Algorithm (MGA).

Rescaling

To control the number of copies of the individuals we adapted the schema defined in [11, 12] and introduced a new operator in the genetic process: rescaling (Fig. 1). Rescaling is a population operator that changes the number of copies of a MI preventing it from taking over population. We divide the number of copies of each individual by a factor *r* and assign the num of copies to the smallest integer that is greater than the division. Fig. 2 shows the effect of the operator in a population of 128 individuals of the problem Maxones over 250 generations.

With the factor *r* equal to 1.0 the number of individuals grows in successive iteration. This is due to keep constant the number of MI in the population where the fittest individuals accumulate copies in the evolutionary process. Higher values of *r* stabilize the number of copies of good individuals. Experimental results show that the value of 1.5 to factor *r* it’s a good compromise between number of individuals and number of MI in the population.

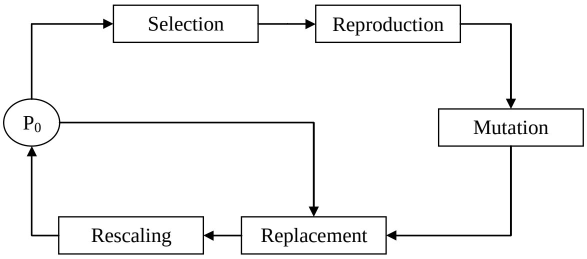


Fig. 1 . Optimization Algorithms Operators

In the first steps of the evolution the number of individuals increases more slowly because there is no individual with greater fitness than the others, and the number of copies is small.

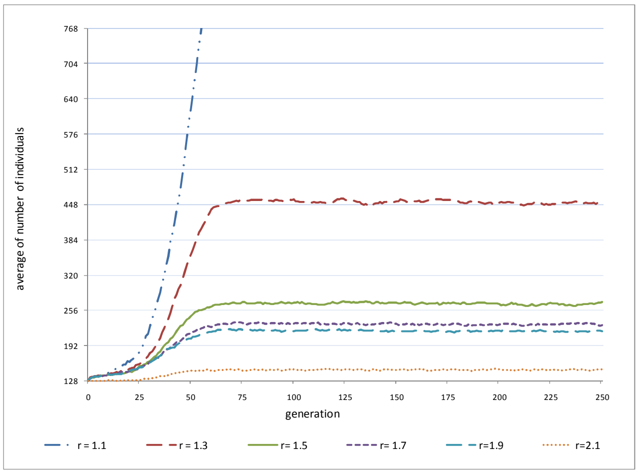


Fig. 2 Effect of rescaling in the number of individuals in the population**.**

4 Empirical study

In our empirical results we use the MUGA simulator [15] to show the effect of MP in the evolutionary process in comparison to SP. To measure the effect of the MP in the evolutionary process we use standard GA operators to evolve the two types of populations: SGA – GA that evolve simple population and MGA – GA that evolve multipopulations. The parameters of the GA are equal for the two types of models and are displayed in Table 8.

Table 8. Parameters used for SGA and MGA except rescaling (only in MGA).

Population size	128 individuals or multiindividuals
Mate population size	128 individuals
Selection	Binary tournament
Combination	Crossover 1-cut point, probability =75%
Mutation	Bitwise mutation with probability =1%
Replacement	Binary tournament with no reposition
Rescaling	Factor = 1.5
Iterations	2500 generations

We use three functions to investigate the effect of MP: Royal Road Function R1 [16], Knapsack [20] and MZ1 [18]. We perform 64 runs for each problem. In each one a new random population is generated and assigned to both MGA and SGA. The results presented below are the arithmetic means of the runs performed.

Royal Road Function R1

This function is designed to investigate, in detail, schema processing and recombination [16]. It uses a 64 bit string. It is unimodal and the search space is organized in steps with constant size.

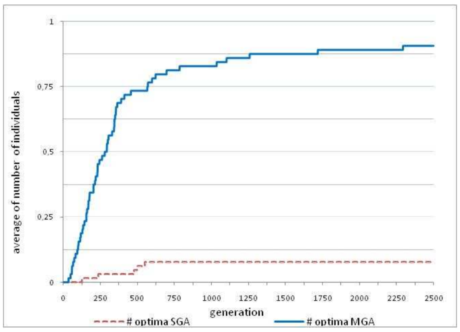


Fig. 3 - Average of number of optima found by generation.

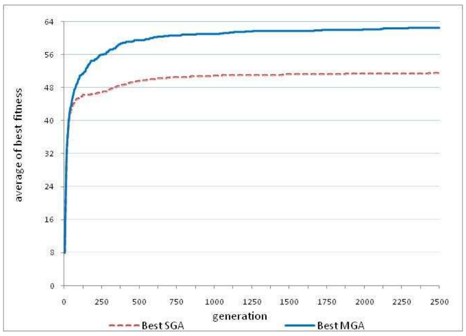


Fig. 4 - Average of value of best individual found by generation.

This function has one maximum and it is hard to discover with no genetic diversity. MGA found the solution 90.6% of the runs and SGA found the optimal solution 7.8% of the runs (Fig. 3, Fig. 4). Fig. 5 and Fig. 6 show the population diversity (different alleles and genetic diversity). The diversity of the SGA is good because the population does not converge and there are many local maxima. MGA has a higher diversity, explained by the constant size of the support set. This genetic diversity in SP eventually generates the best solution but more iterations are necessary. If we take all tests up to 2500 generations we notice the number of evaluations in MGA (22700) is lower than in SGA (29500) (Fig. 7). The number of individuals in MGA stabilizes in 166 after a few generations (Fig. 8). In the beginning of evolution the number of individuals increases because there are many local maxima with the same fitness. In each higher fitness plateau the number of local maxima decreases which leads to the stabilization of the number of individuals in a smaller value.

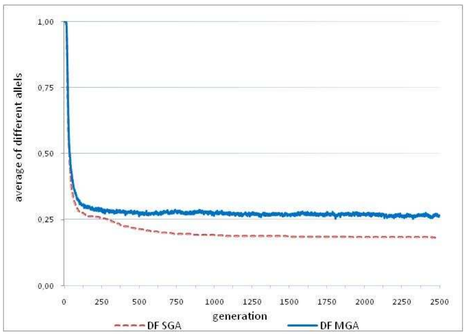


Fig. 5 - Average of number of different alells of the population by generation.

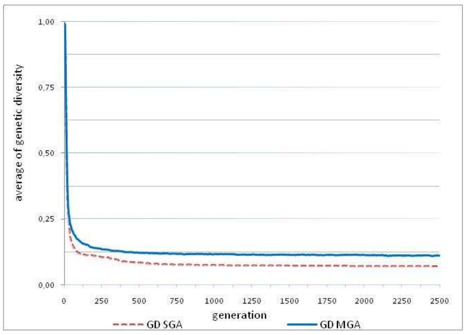


Fig. 6 - Average of genetic diversity by generation.

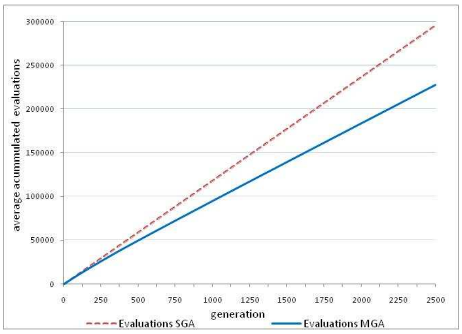


Fig. 7 - Average of accumulated evaluations per generation.

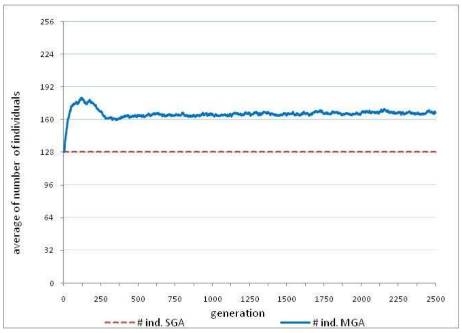


Fig. 8 - Average number of individuals by generation.

Knapsack

Knapsack is a well know NP-complete combinatorial optimization and we use it to show the effects in combinatorial optimization problems. We implement the problem presented in [20]. The maximum capacity is 50% of the total height and penalization is done by the linear function of [18].

Experimental results show that the problem has at least four maxima with the best value known of 1920:

000001111101001011001001011110100110101011111110111

0000011111101001011001001011110100010101011111110111
0000011101101001011001001011111101110100011111110111
0000011111101001011001001011110101110101011110110111

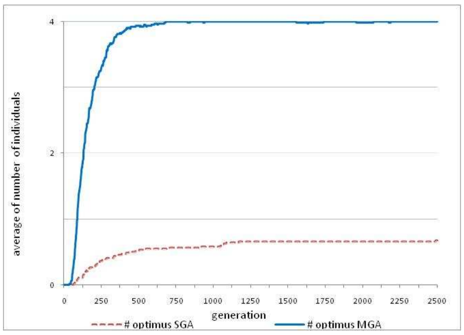


Fig. 9 - Average of number of optima found by generation.

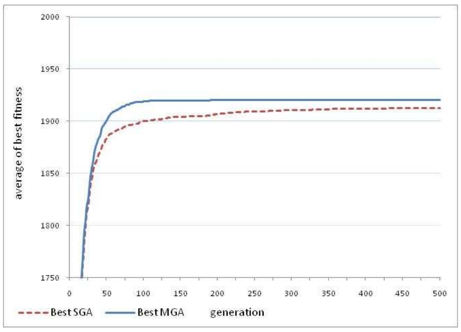


Fig. 10 - Average of value of best individual found by generation.

The MGA always found four optimum solutions (Fig. 9) and the SGA found one optimum 67.1% of the runs. In both simulations a good solution is found. The mean of the best values for SGA is 1915.4 and for MGA is 1920 (the best known) (Fig. 10).

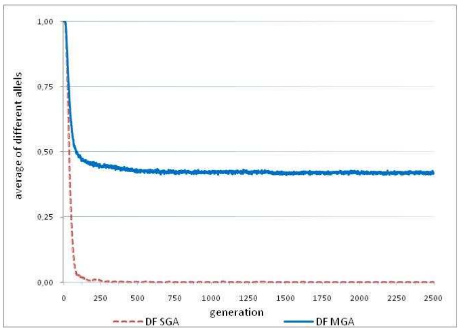


Fig. 11 - Average of number of different alleles of the population by generation.

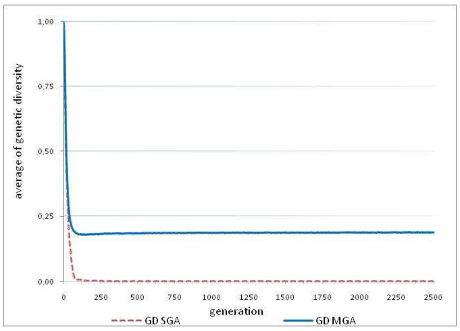


Fig. 12 - Average of genetic diversity by generation.

The number of different alleles (Fig. 11) and genetic diversity (Fig. 12) in SGA converge to zero while in MGA they converge to a good ratio: 41.9% of different alleles and 18.7% of genetic diversity. This lack of genetic diversity explains the difficulty of SGA to find optimum solutions. The number of evaluations is 10.8% higher in SGA and the number of individuals in MGA stabilizes in 226. The higher number of individuals is explained by the number of the optima found by the MGA.

MZ1

MZ1 [18, pp. 36] is a bidimensional function defined in a continuous space and we use it to show the effect of MP in numerical optimization based in real numbers coded in binary strings. The fitness function is described by equation 3. Variable x_1 is defined in the interval $-3.0 \leq x_1 \leq 12.1$ and coded by 18 bits and variable x_2 is defined in the interval $4.1 \leq x_2 \leq 5.8$ and coded by 15 bits.

$$MZ1(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_2) + x_2 \sin(20\pi x_2) . \quad (3)$$

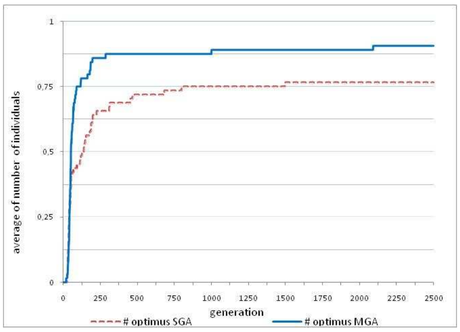


Fig. 13 - Average of number of optima found by generation.

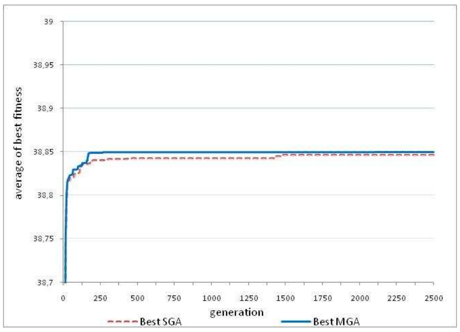


Fig. 14 - Average of value of best individual found by generation.

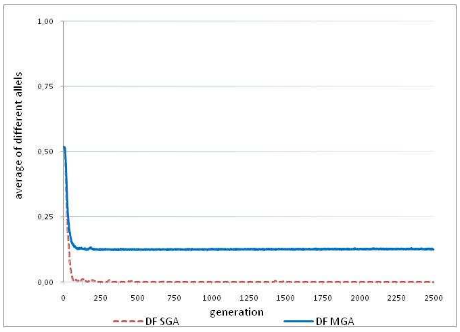


Fig. 15 - Average of number of different alleles of the population by generation

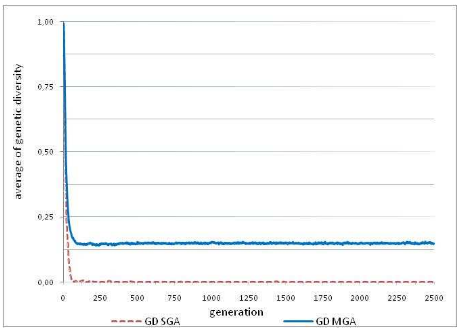


Fig. 16 - Average of genetic diversity by generation

We notice that the results of MGA in this problem are not as good as in the previous ones. Optimization of complex functions is difficult for SGA due to premature convergence of the population. In MGA we also notice premature convergence (Fig. 13 and Fig. 14), in spite of the constant population diversity (Fig. 15 and Fig. 16). This is due to the fact that MGA converges to a local maximum and uses the genetic diversity to form a cluster around it. The number of individuals in the

MP stabilizes in 338, which is a high number resulting from many MI with similar high fitness.

5 Conclusion

In this paper we present an implementation of GA with populations represented by multisets, named Multiset Genetic Algorithm (MGA). This model introduces a new operator called Rescaling, to allow a comparison between MGA and the Simple Genetic Algorithm (SGA). We performed experiments to determine an adequate value for the division factor parameter of the new operator. A metric called genetic diversity, implementing in an efficient way a normalized Hamming distance was proposed to evaluate the performance of the different models.

We conducted a series of tests in different problems, to determine the behavior of the MGA during the evolutionary process and to compare the results with the SGA. The number of evaluations in MGA is lower than in SGA in all the tests. MGA has the largest genetic diversity in all situations. This helped to reach the optimal solution in Royal Road Function R1 and to find and maintain different optimal solutions in the knapsack problem. MGA in MZ1 problem did not get results as significant as in the other problems, but showed some directions to follow in the future. The cluster around a local maximum takes the entire genetic diversity through bits with little significance to the global optimum. The concept of multiindividual (MI) can be redefined with the incorporation of metrics that allow a single MI to represents similar genotypes instead of a single one. Some metrics are being study and will be subject to investigation in the near future.

This work presents the impact of MGA in a optimization process using conventional genetic operators. The standard genetic operators could be redefined, and new ones can be developed take to advantage of the number of copies of the MI. Some operators have already been adapted and are present in MUGA[15]. In the future they will be subjected to analytical treatment to determine their efficiency.

The use of multipopulations involves a computational effort grouping individuals in a MI. Efficient data structures that support efficient search and random access to the individuals will be researched to minimize the computational effort.

The results of the experiences reinforce our conviction that a multiset representation of the population is a powerful way to preserve genetic diversity, to avoid superfluous evaluations and, therefore, to significantly improve the performance of GA.

References

1. Whitley, D.:An Overview of Evolutionary Algorithms: Pratical Issues and Common Pitfalls. Journal of Information and Software , 43:817-831. (2001)
2. Theoretical analysis of diversity mechanisms for global exploration. Proceedings of the 10th annual conference on Genetic and evolutionary computation, pp 945-952 (2008)

3. Amor, B.H.,Rettinger, A.:Intelligent Exploration for genetic Algorithms: using self-organizing maps in evolutionary computation. Proceedings of the 2005 conference on Genetic and evolutionary computation, pp1531-1538. (2005)
4. Cervantes, J., Stephens,C.R. : Rank based variation operators for genetic algorithms. Proceedings of the 10th annual conference on Genetic and evolutionary computation, pp 905-912 (2008)
5. Cherba, M.D., Punch, W.: Crossover gene selection by spatial location. Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp 1111 - 1116 (2006)
6. Hutter, M.:Fitness Uniform Selection to Preserve Genetic Diversity. Proceedings of the 2002 Congress on Evolutionary Computation, pp 783-788 (2002)
7. Lima, C.F., Sastry K., Goldberg, D.E., Lobo, F.G.: Combining competent crossover and mutation operators: A probabilistic model building approach. Proceedings of the 2005 conference on Genetic and evolutionary computation, pp 735 – 742 (2005)
8. Deb, K.: Current trends in evolutionary multi-objective optimization. International Journal for Simulation and Multidisciplinary Design Optimization, pp1-8 (2007)
9. Schmidt, M.D., Lipson H.:Co-evolution of Fitness Maximizers and Fitness Predictors. GECCO Late Breaking Paper, (2005)
10. Kim,E., Cho, S. , An efficient genetic algorithm with less fitness evaluation by clustering.Proceedings of the 2001 Congress on Evolutionary Computation, volume 2, pp 887–894 (2001)
11. Aparício J.N., Correia, L., Moura-Pires, F.: Populations are Multisets-PLATO. GECCO 1999, pp 1845-1850 (1999)
12. Aparício J.N., Correia, L., Moura-Pires, F.: Expressing Population Based OptimizationHeuristics Using PLATO. EPIA 1999, pp 367-383 (1999)
13. Blizard, W. Multiset Theory. Notre Dame Journal of Formal LogicVolume 30, Number 1, Winter (1989).
14. Wagner,S.,Affenzeller,M:The Allele Meta-Model - Developing a Common Language for Genetic Algorithms. Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, Lecture Notes in Computer Science 3562, pp. 202-211. Springer-Verlag (2005)
15. Manso,A. ,Correia,L.: MUGA (MultiPopulations Genetic Algorithm) - Intelligent Systems Demonstrations Event, EPIA (2005).
16. Mitchell, M.: Introduction to Genetic Algorithms, 5ª ed. MIT Press, pp 94-98 (1998)
17. Jong, Keneth A. de: Evolutionary Computation - A unified approach, The MIT Press, pp 163 (2006)
18 Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 2rd extend Edition Springer-Verlag pp. 36 ; pp 80 (1996)
19 Singh, G, Deb, K.: Comparison of multi-modal optimization algorithms based on evolutionary algorithms, Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp 1305 – 1312 (2006)
20 Evolutionary Computation Benchmark Repository- Fifty item Knapsack Problem, http://www.cs.bham.ac.uk/research/projects/ecb/ (July , 2009)
21 Ochoa, Gabriela. Error Threashold in Genetic Algorithms, Evolutionary Computation Journal, MIT Press; pp. 157-182 (2006)

How to Build the Network of Contacts

Selecting the Cooperative Partners

Pedro Mariano[1], Luís Correia[2], and Carlos Grilo[2,3]

[1] Transverse Activity on Intelligent Robotics – IEETA – DETI, Universidade de Aveiro, Portugal
[2] LabMAg – Dep. de Informática, Faculdade de Ciências, Universidade de Lisboa, Portugal
[3] Dep. Eng. Informática, Escola Superior de Tecnologia e Gestão Instituto Politécnico de Leiria, Portugal

Abstract. We address the problem of finding the correct agents to interact with from a general standpoint. We take the payoff obtained by agents in any game with dilemma as an input to our model. Its output is a probability distribution used in the partner selection that increasingly favours cooperative agents. Our approach contrasts with others designed for specific games without concerns of generality. We show both theoretically and experimentally that the major factor affecting cooperators selecting only themselves is the agents' strategies. This result does not depend on game nature or the initial probability distribution.

1 Introduction

Models of reputation or the chance of punishing are typically models analysed in situations where a dilemma is present. Reputation is used by agents as a measure of how well one behaves, and thus leads to one agent being favoured in detriment of others. Punishment is used as a method to discourage the proliferation of non-cooperators [7]. These models are often used with concrete, but representative, games [13, 2].

Such models are put forward to explain the prevaillance of cooperation in a population of interacting agents. Typically such models assume that the agents are laid in some lattice that restricts with whom an agent may interact [20]. Graphs are often used to describe such restrictions. Such graphs are also called networks of contacts. There are also models that focus on which type of network favours the development of cooperation [12].

The model that we present focuses on partner selection by agents. Our agents are able to select with whom they wish to interact. This approach is often taken on models of reputation [9, 18, 19, 7], where an agent maintains some vector that classifies its potential partners. This vector is then used to pick up the most promising partners.

We will consider that the only information an agent gets from an interaction is its payoff. It does not know which actions were taken by its partners. This feature allows an agent to select its partners, only requiring knowledge about the payoffs of whichever game is used for interaction.

The motivation of the model presented in this paper is to address the problem of how cooperative agents are able to survive in a population subject to interaction constraints. There are several approaches to instill cooperative behaviour in a population of interacting agents. This is a topic of much research in Game Theory (see for instance [6, 8, 5, 21, 1, 4, 8]), Sociology, Biology (see [16, 13, 3]), and Artificial Intelligence (see [9, 19, 22]), to name a few areas.

As we use Game Theory to model the interactions between agents, our model makes direct use of the payoffs of a game in order to select the cooperative partners. The model consists in a probability vector maintained by each agent that in each position represents the probability of selecting an agent as a partner to play a game.

Since the agent has to find the best partner, the algorithm can be compared to a Cournot adjustment process [10] were players iteratively adjust their strategies to their partner responses. In this paper, an agent strategy remains constant but it adjusts its preferences towards more profitable or cooperative partners. Similar approaches to partner selection have been tackled in [14] but they focused on a specific game such as Prisoner's Dilemma (PD). With the approach we now propose, we are able to do partner selection in any situation capable of being described as a game, and we are not limited to any particular game.

2 Definitions

We will use Game Theory as the tool to model interaction between agents. To this end, we will consider that a population $\mathbb{P}$ of agents interacts accordingly to the rules of some n-player game $\mathcal{G}$. The game describes the strategies available to players and the payoffs they obtain as a function of the strategies used. The game has a n-dimensions strategy space $\mathbf{S} = \mathbb{S}_1 \times \mathbb{S}_2 \times \ldots \times \mathbb{S}_n$ where agents can draw a strategy $s_i \in \mathbb{S}_i$ to play a game. The vector $\mathbf{s} = (s_1, \ldots s_n)$ represents a strategy profile of the n players involved in the game. The game also has n payoff functions, $u_i : \mathbf{S} \to \mathbb{R}$, with $i \in \{1, 2, \ldots, n\}$. The payoff values are bounded and belong to $\mathbb{R}$. Let $\underline{u}$ be the lowest payoff and $\overline{u}$ be the highest payoff in game $\mathcal{G}$.

Agents are composed of a strategy $s_i \in \mathbb{S}_i$. This means agents are characterised by the role they play in game $\mathcal{G}$. As an example, a game may model a buyer/seller scenario. Generally speaking, the game is asymmetric. This means the following condition is true:

$$\exists i \exists j : i \neq j \implies \mathbb{S}_i \neq \mathbb{S}_j.$$

An agent to play a game $\mathcal{G}$ must select $n-1$ agents of the appropriate role, i.e. each one with a strategy s_j with $j \neq i$. If the game is symmetric, then all the strategy sets $\mathbb{S}_i$ are equal.

We also aim at reaching a position where cooperative agents only interact between themselves. As cooperative agents we define those that form a strategy profile that maximises the average payoff of the players. We define the payoff obtained by such Pareto Optimal profile as follows:

$$u_P = \max_{\mathbf{s}} \sum_i \frac{u_i(\mathbf{s})}{n}.$$

As an example, in the Iterated Prisoner's Dilemma (IPD) game [11], a cooperative agent is one that does not defect, and in a Public Good Game (PGG) [11] a cooperative agent is one that contributes to the common good.

Table 1 summarises the nomenclature used throughout this document.

α, β, γ	agents
$\mathbb{P}$	the population of agents
$\mathcal{G}$	a n-player game
$\overline{u}$	highest payoff in game $\mathcal{G}$
$\underline{u}$	lowest payoff in game $\mathcal{G}$
u_P	payoff obtained by an agent in a Pareto Optimal profile
$w_{\alpha,\beta}$	label of the edge from agent α to agent β

Table 1. Nomenclature used in this document

3 Model Description

A population $\mathbb{P}$ of agents is represented by a directed simple graph where a vertex represents an agent α and a labelled edge from α to β represents the probability of agent α interacting with β.

$$\mathbb{P} = (V, E)$$

with:

$$\begin{aligned} V &= \{\alpha, \beta, \gamma, \ldots\}, \\ E &= \{(\alpha, \beta, w_{\alpha,\beta}), \ldots\}, \\ w_{\alpha,\beta} &\geq 0, \\ \sum_\beta w_{\alpha,\beta} &= 1. \end{aligned}$$

This definition allows us to represent the directed simple graph by a matrix. If there is no link from agent α to β, then it is assumed that probability $w_{\alpha,\beta}$ is zero.

Each agent performs the following algorithm:

1. partner selection,
2. play the game,

3. update $w_{\alpha,\beta}$ probabilities.

In the first iterations of the algorithm, the agent must find the best partners and increase their probability of being selected, while decreasing the chance of selecting bad partners. Our model must strike a balance between exploring new neighbours and exploiting the best agents it has found so far.

The algorithm can be run in a distributed fashion by each agent. No communication between agents is necessary, nor any central control is required. The $w_{\alpha,\beta}$ matrix could be split in vectors, each one maintained by the corresponding agent.

3.1 Update policy

The edge weight update policy for agent α is a function defined as follows:

$$w_{\alpha,\beta}^{t+1} = \zeta(w_{\alpha,\beta}^{t}, u_{\alpha}^{t})$$

where $w_{\alpha,\beta}^{t}$ is the edge weight before the game and u_{α}^{t} is the payoff agent α obtains in the game.

The main focus of the work presented in this paper is the analysis of an update policy that meets the following two conditions:

Cooperative aggregation Cooperative agents are mostly connected to each other. If α and a neighbour β are part of a Pareto Optimal profile, then in the limit the sum of the probability of selecting all β should be 1:

$$\sum_{\beta} \lim_{t\to\infty} w_{\alpha,\beta}^{t} = 1 \quad \forall_{\beta}\, u_{\alpha}(\ldots, s_{\beta}, \ldots) = u_P. \tag{1}$$

Stability The update policy must be robust in order to resist perturbations in the population and to be applicable to any n-player game. In the long run and in the absence of perturbations, weights must stabilise:

$$\lim_{t\to\infty} (w_{\alpha,\beta}^{t+1} - w_{\alpha,\beta}^{t}) = 0. \tag{2}$$

We are assuming that the number of cooperative agents is equal or higher than the number of players and partner selection is done without replacement. Otherwise, a cooperative agent does not have enough partners to play a game.

The edge weight update policy is divided in two cases depending on whether an agent played the game with agent α or not.

Agent β played the game A simple policy is to multiply the old weight by a factor that is proportional to the distance between the payoff u obtained by agent α and the Pareto Optimal payoff u_P:

$$w_{\alpha,\beta}^{t+1} = \begin{cases} w_{\alpha,\beta}^{t} \dfrac{u - \underline{u}}{u_P - \underline{u}} & u < u_P \\ w_{\alpha,\beta}^{t} & u = u_P \\ w_{\alpha,\beta}^{t} \dfrac{\overline{u} - u}{\overline{u} - u_P} & u > u_P . \end{cases}$$

This rule by itself does not guarantee the condition stated by equation (1). Only combined with the rule for the case of agents that were not selected we achieve it. Regarding the stability, this rule will keep weights unchanged if the payoff is equal to u_P. Otherwise they will tend to zero as in the first and third cases $w^t_{\alpha,\beta}$ is multiplied by a factor always less than 1. Either way, equation (2) is met.

This policy from the viewpoint of an uncooperative agent is irrational. Such agents defect in the PD, do not provide the good in PGG, keep all the money in the Ultimatum. Their payoff is often greater than u_P. They should keep selecting the partner in order to exploit him. Instead of a single peak, the update policy should have a cutoff threshold:

$$w^{t+1}_{\alpha,\beta} = \begin{cases} w^t_{\alpha,\beta} \dfrac{u - \underline{u}}{u_P - \underline{u}} & u < u_P \\ w^t_{\alpha,\beta} & u \geq u_P. \end{cases} \tag{3}$$

This rule also verifies the stability condition. Another characteristic is that if $u \geq u_P$ the agent will not change its probabilities and thus it will probabilistically play with the same agents. Since it is rational for all agents, we will consider it for the remainder of the paper.

Agent γ did not play the game The multiplicative factor used for agents that played the game implies that the weight of all agents that played will either stay the same or decrease. If they decrease, the difference must be distributed among the other edge weights. A simple solution is to distribute equally:

$$w^{t+1}_{\alpha,\gamma} = w^t_{\alpha,\gamma} + \frac{s}{x} \tag{4}$$

where s is the sum of the difference of all link values, egressing node α, that have played the game in the current round,

$$s = \sum_\beta (w^{t+1}_{\alpha,\beta} - w^t_{\alpha,\beta})$$

and variable x is the number of neighbours of agent α that were not selected.

This equation shows that this policy explores alternative partners if $u < u_P$, since the probability of others being selected in the next iteration is increased.

Equation (4) combined with equation(3) is able to achieve the condition expressed by equation (1). If a cooperative agent selects an uncooperative, the corresponding weight will decrease towards zero. The difference is distributed among the weights of players that were not selected. However, the weight of a second uncooperative partner also increases, but not by much. Even if this second partner is selected, its weight is reduced and distributed among all the partners. The point is that, in the long run, weights of uncooperative agents decrease while weights of cooperative agents will absorb the distributed differences.

4 Experimental Analysis

The purpose of the experiments reported in this paper is to assess the time cooperative agents take to only select themselves as partners of interaction. To this end, simulations with different proportion and quantities of cooperative agents were performed. Other conditions were also analysed such as initial probability values, different games and, in one case, different game parameters.

4.1 The Games

The selected games were the Give-Take game and IPD. These games have been chosen because they pose a dilemma. The first one is a 2-player game with several Pareto Optimal strategy profiles [15], and the second is commonly used.

The Give-Take game is played between two agents that must share a single resource. Only one agent can hold it and benefit from it at each iteration. That agent can give the resource to its partner. The partner can take (grab) the resource or do nothing. Their roles will remain the same, provided both players do nothing. Agents change role whenever the resource ownership is changed. Whenever an agent gives the resource away, it can receive a bonus, b_g. An agent that takes (grabs) the resource (from its partner) pays a penalty as its partner.

Two sets of parameters were used differing on the value of parameter b_g: possession of the resource per iteration was set to 1, giving the resource, b_g, yields 0.5 or 0 units, taking the resource costs 2 units to the performer and 1 unit to the subject of the action. The game had at least 100 iterations, and after the 100^{th} the probability of occurring one more iteration was set to 0.5. Due to implementation decisions, game length was limited to 1000 iterations.

The strategy used by the agents is deterministic and has two parameters: t_g number of iterations the agent waits (after obtaining the resource possession) before giving it to its partner; t_t number of iterations the agent waits (after loosing the resource possession) before taking it from its partner. These parameters allow us to have strategies that never give the resource ($t_g = \infty, t_t = x$) or that do nothing ($t_g = \infty, t_t = \infty$).

The payoffs of IPD were: temptation to defect 5; cooperate 3; suckers 0; penalty 1. Each game lasted at least 10 iterations, and after the 10^{th} the probability of occurring one more iteration was set to 0.5. The strategy used by agents is stochastic and has one parameter: the probability to cooperate.

4.2 The Population

Regarding the agents, different strategies where used, which can be roughly classified in how cooperative they are. The strategies used in the Give-Take game are:

S1, S2, S3 These strategies always give the resource, but do so at different times: 1, 10 and 20 iterations. They never take the resource. Therefore, if both agents use the same strategy, their form a Pareto Optimal profile.

However, when agents use different strategies, the one that gives the resource sooner is explored;

S4 An uncooperative and aggressive strategy, as it never gives the resource, and takes it immediately after it looses its possession.

The strategies used in IPD are:

S1 A cooperative strategy that always cooperates;
S2, S3 Two strategies that cooperate with probabilities 0.7 and 0.3;
S4 One is uncooperative as it always defects.

The number of strategies in the population, for each type, varied between 2, 4, 8, 16 or 32 agents. This allows us to study the time taken for different proportions and quantities of cooperative strategies to mostly select themselves as partners. For instance 32 **S1** versus 2 of each **S2**, **S3** and **S4** is a scenario that favours cooperative agents. A scenario with 8 of each **S1**, **S2**, **S3** and **S4** is balanced for all strategies. On the contrary, a scenario with 2 **S1** versus 32 of **S4** and any number of **S2** and **S3** is very unfavourable for the cooperative agents as they will take longer in finding the correct partners.

Total population varied among $\{8, 10, \ldots, 112, 128\}$. This allows us to study how cooperative agents select themselves as interaction partners in different scenarios. We opted for a majority of cases with a small number of agents because we are interested in analysing the edge weights and viewing the resulting graph. For bigger populations we have 128^2 edges to examine, therefore, in this case, we only analyse global results. As for the initial edge weights, two options were used: identical values so that every agent has the same chance of being selected; random values.

4.3 Other Parameters

Each simulation consisted of 1000 rounds of games. In each round all agents played at least one game, since the following steps were performed per round for every agent: select $n-1$ partners proportionally to the edge weights, play the game, update the edge weights of the agent that selected partners. After each round the edge weights were recorded.

The simulation was developed in JDK 1.6. Random numbers were produced using an instance of the `cern.jet.random.engine.MersenneTwister` class [17], a pseudo-random number generator that has a large period of $2^{19937} - 1$.

Table 2 shows all the conditions tested in the experiments. For each combination of conditions 10 runs were performed and results averaged.

4.4 Results

We have plotted the average probability of a group of agents (all with the same strategy) of selecting a group of agents (again all equal in terms of strategy). We have also calculated the standard deviation.

games	Give-Take $b_g = 0.5$ Give-Take $b_g = 0$ IPD
initial edge weight	random identical
# **S1**	2, 4, 8, 16, 32
# **S2**	2, 4, 8, 16, 32
# **S3**	2, 4, 8, 16, 32
# **S4**	2, 4, 8, 16, 32

Table 2. Summary of the conditions tested in the simulations.

Figure 1 shows the results from the simulations with Give-Take and $b_g = 0$. In this figure we additionally show the average probability at the 10000^{th} round. Figure 2 shows the probability graph obtained in a simulation with 8 agents.

Figure 3 shows some results from the simulations with Give-Take and $b_g = 0.5$ while figure 4 shows the results obtained with IPD. Since results are similar, there are fewer plots in these figures.

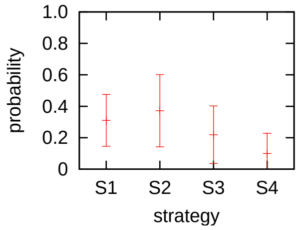
5 Discussion

In the following discussion we will use the term *aggregation time* to refer to the time taken for a group of strategies to mostly select themselves. That is to say, how many rounds of edge weight updating must occur in order to observe an approach to the condition stated by equation (1).

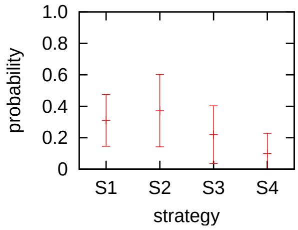
Initial edge weights do not influence aggregation time. This can be observed in figures 1(a) and 1(b). Both refer to Give-Take with $b_g = 0$ and two **S1** agents. Other numbers of **S1** agents confirm these findings (results not shown).

Game nature does not influence the fact that in the long run cooperative agents aggregate. However, aggregation time depends on game nature as well as on agent strategies. When we compare figure 3(a) with figure 4(b), both have the same number of cooperative strategies, but aggregation time differs. The same line of reasoning can be established between figure 1(a) and figure 4(a), although the first only includes simulations with initial random edge weight and the second includes both identical and random initial edge weights. All other number of **S1** agents (see table 2) produced similar results (not shown).

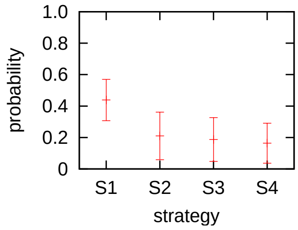
In the Give-Take with $b_g = 0$ there can be three groups of cooperative strategies. However, whenever a game is played, the number of iterations may vary, and it may not be a multiple of the time each strategy keeps the resource before giving it. This means that in a game between strategies **S3**, one of them may get a payoff lower than u_P and thus decrease the probability of selecting its peer. In figure 2, taken from a simulation with 2 agents per strategy, we show the edge weight graph with weights lower than 0.15 omitted. It can be seen that **S3** agents prefer **S2** and **S1** agents.



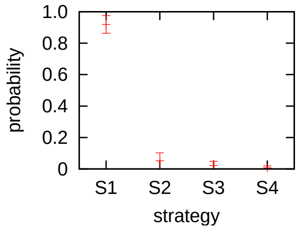
(a) Initial random edge weights and 2 agents with **S1** strategy.



(b) Initial identical edge weights and 2 agents with **S1** strategy.



(c) Initial random edge weights and 2 agents with **S1** strategy. Edge weights taken at $10,000^{\text{th}}$ round.



(d) Simulations with 16 **S1** strategies.

Fig. 1. Results of Give-Take with $b_g = 0$. Vertical axis represents the probability of strategy **S1** choosing a strategy in the horizontal axis. In cases 1(a) to 1(c) the results are averages of all possible combinations of strategies **S2** to **S4** (see table 2). In case 1(d) we also varied the initial edge weights.

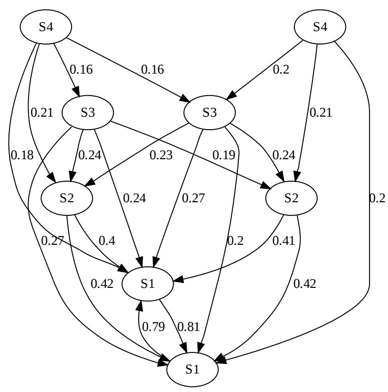
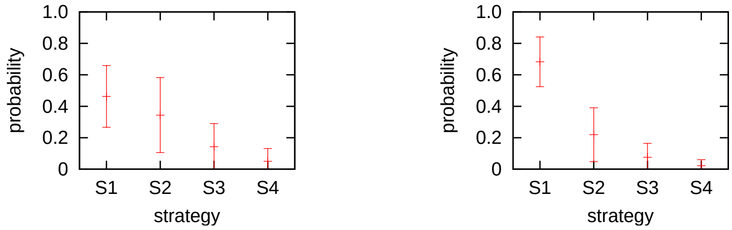
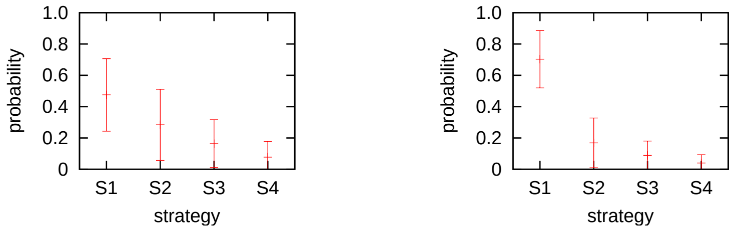


Fig. 2. Results of Give-Take with $b_g = 0$. Probability graph in the last round of a single simulation run with 2 agents per strategy type. Edge weights lower than 0.15 were omitted.



(a) 4 agents with strategy **S1**. (b) 8 agents with strategy **S1**.

Fig. 3. Give-Take with $b_g = 0.5$. Results are averages of all combinations of parameters not fixed (see table 2).



(a) 2 agents with strategy **S1**. (b) 4 agents with strategy **S1**.

Fig. 4. IPD results - averages of all combinations of parameters not fixed (see table 2).

Results show that the number of cooperative strategies influences the aggregation time. The higher is their number the shorter is aggregation time. This can be observed in figure 1(d), which refers to Give-Take with $b_g = 0$ and 16 agents with strategy **S1**, that has the highest probability of this strategy selecting a peer. Comparing figures 3(a) and 3(b), which refer to Give-Take with $b_g = 0.5$ and have, respectively, 4 and 8 agents with strategy **S1**, we observe that the second figure has the highest probability. If we increase the number of rounds, the probability of cooperative strategies to select themselves increases, as can be seen by comparing figure 1(a) taken at the 1000$^{\text{th}}$ round and figure 1(c) taken at the 10000$^{\text{th}}$ round.

Deterministic and stochastic strategies do not impinge on aggregation time. Instead, strategies' cooperative nature has an influence on it. Agents with cooperative strategies that are part of a Pareto Optimal profile will only select themselves.

6 Conclusions and Future Work

We have proposed and analysed theoretically a model for partner selection in a Multi-Agent System interacting through a game. We have complemented this analysis with an experimental simulations. The results confirm the theoretical analysis. In particular, game nature or initial edge weights do not influence the final aggregation result, although the former affects aggregation time while the latter does not. In the long run, cooperative agents will only select themselves as partners of interaction. The number of cooperative agents strongly influences their aggregation time. The more they are, the faster they aggregate.

This model requires a memory size that grows linearly with the number of partners. It also requires that agents are uniquely identifiable. The agent must know in advance the population size.

Despite some limitations, the edge weight update policy does not suffer from the problem of a period of learning followed by a period of reaping the benefits of education. If an agent does not get a good payoff from the current players, it will raise the edge weights of all the other agents, giving them a chance of being selected. However, further confirmation of this property should be tested with simulations where new agents are periodically introduced in the population, and the games and agents that select them should be monitored.

If agents select their partners, they can also refuse to play with certain agents. This could lead to situations where an agent cannot play because all its neighbours refuse to play. However, the possibility of refusing brings the problem of collecting experience back. During the period in which an agent is testing its neighbours, it may be explored by non-cooperators. In addition, after an agent closes its learning window, new agents will never play with it, unless we allow the reopening of the learning window.

Agent strategies are fixed. We should analyse the effects of adaptation, and apply the edge weight update policy to a framework of evolutionary games. An agent that cannot find a suitable partner can overcome this if it changes its strategy.

Acknowledgements

This work is partially supported by FCT/MCTES grant SFRH/BD/37650/2007.

References

1. James Andreoni, Paul M. Brown, and Lise Vesterlund. What makes an allocation fair? Some experimental evidence. *Games and Economic Behaviour*, 40:1–24, 2002.
2. Pat Barclay. Reputational benefits for altruistic punishment. *Evolution of Human Behaviour*, 27:325–344, 2006.
3. R Boyd and P J Richerson. Punishment allows the evolution of cooperation (or anything else) in sizable groups. *Ethology and Sociobiology*, 13:171–195, 1992.
4. Jordi Brandts and Neus Figueras. An exploration of reputation formation in experimental games. *Journal of Economic Behavior & Organization*, 50:89–115, 2003.
5. David V. Budescu, Amnon Rapoport, and Ramzi Suleiman. Common pool resource dilemmas under uncertainty: Qualitative tests of equilibrium solutions. *Games and Economic Behaviour*, 10(1):171–201, 1995.

6. Terence C. Burnham. Engineering altruism: a theoretical and experimental investigation of anonymity and gift giving. *Journal of Economic Behavior & Organization*, 50(1):133–144, 2003.
7. F.A.C.C. Chalub, F.C. Santos, and J.M. Pacheco. The evolution of norms. *Journal of Theoretical Biology*, 241(2):233–240, 2006.
8. David J. Cooper and Carol Kraker Stockman. Fairness and learning: an experimental examination. *Games and Economic Behaviour*, 41(1):26–45, 2002.
9. Chrysanthos Dellarocas. Goodwill hunting: An economically efficient online feedback mechanism in environments with variable product quality. In Rino Falcone and Larry Korba, editors, *5th Workshop on Deception, Fraud and Trust in Agent Societies*, pages 26–40, 2002.
10. Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, 1991.
11. Herbert Gintis. *Game Theory Evolving - A problem-centered introduction to modeling strategic interaction*. Princeton University Press, 2000.
12. Christoph Hauert. Spatial effects in social dilemmas. *Journal of Theoretical Biology*, 2006.
13. Joseph Henrich and Robert Boyd. Why people punish defectors: Weak conformist transmission can stabilize costly enforcement of norms in cooperative dilemmas. *Journal of Theoretical Biology*, 208:79–89, 2001.
14. Leslie Luthi, Mario Giacobini, and Marco Tomassini. Synchronous and asynchronous network evolution in a population of stubborn prisoners. In *CIG*. IEEE, 2005.
15. Pedro Mariano and Luís Correia. A resource sharing model to study social behaviours. In Fernando Moura Pires and Salvador Abreu, editors, *Progress in Artificial Intelligence - 11th Portuguese Conference on Artificial Intelligence, EPIA 2003*, pages 84–88. Springer-Verlag, 2003.
16. John Maynard Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.
17. cern.jet.random.engine.mersennetwister class part of colt library. http://acs.lbl.gov/ hoschek/colt/. last checked March 2009.
18. Sarvapali D. Ramchurn, Dong Huynh, and Nicholas R. Jennings. Trust in multiagent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.
19. Jordi Sabater and Carles Sierra. Review on computational trust and reputation models. *Artif. Intell. Rev.*, 24(1):33–60, 2005.
20. Francisco C. Santos, Marta D. Santos, and Jorge M. Pacheco. Social diversity promotes the emergence of cooperation in public goods games. *Nature*, 454:213–216, July 2008.
21. Frans van Dijk, Joep Sonnemans, and Frans van Winden. Social ties in a public good experiment. *Journal of Public Economics*, 85:275–299, 2002.
22. Bin Yu and Munindar P. Singh. Detecting deception in reputation management. In *AAMAS'03*. ACM, 2003.

A Tool for Automatic Routing of Auxiliary Circuits in Ships

Paulo Triunfante Martins[1], Victor J.A.S. Lobo [1,2]
[1] Portuguese Naval Academy
[2] ISEGI - Universidade Nova de Lisboa
vlobo@isegi.unl.pt

Abstract. This work proposes a method and software for the automatic layout design of auxiliary circuits within a compartment of a ship, including pipes and cable trays. This task is normally part of the detailed design phase and so it is expected that the ship's structure, general arrangement and equipment layout has already been decided. Accordingly, knowing the start and goal points for each circuit path, all of them are routed in a 3D space taking into consideration, obstacles, ramifications and several kinds of constraints. The method is implemented using routing algorithms and genetic algorithms in order to find the best possible group of paths considering all the above.

Keywords: Genetic Algorithms, Piping, Routing.

1 Introduction

The ship design and construction process goes through several stages, starting by its concept design, basic design, detail design and design for production. Typically, it is on the basic design stage that the ship structure, piping and electrical circuits are defined, and the various equipments are selected. Then, in the detail design stage, everything has to come together in a unique design, taking into account the specific set of solutions found.

Amongst the various tasks of the detailed design stage, the pipe layout design (PLD) is one of the more time-consuming and prone to error, and thus can originate more reworks. This comes to be even more important for medium size ships, such as small tankers, research and military vessels, due to lack of space. In fact, bad routing can affect the ship's local structure, as well as her operation and maintenance.

This problem is similar to the routing problem in circuit design, but there are significant differences: the PLD, unlike circuit design, is truly 3-dimensional, and the type of restrictions and desirable characteristics are quite different.

Nowadays, there are several software tools available, that integrate the design and manufacturing systems. In most cases, they have a design module for 3D layout, piping and HVAC (heating, ventilation and air conditioning), complemented with a tool for checking collisions between different objects.

However, we do not know any tool that provides a fully automated system to solve the PLD. This problem consists on finding the layout of pipe and cable routes minimizing the length of the path connecting two or more objects (equipments), using only one path or a main path with ramifications (branch handling), while avoiding obstacles such as ships structure, equipments and other circuits, in accordance with several shape and location constraints (maintenance, ergonomics, structural integrity, etc). Fig. 1 shows the pipe layout of a vessel, as taken from NUPAS Cadmatic home page (http://www.nupas-cadmatic.com/)[1].

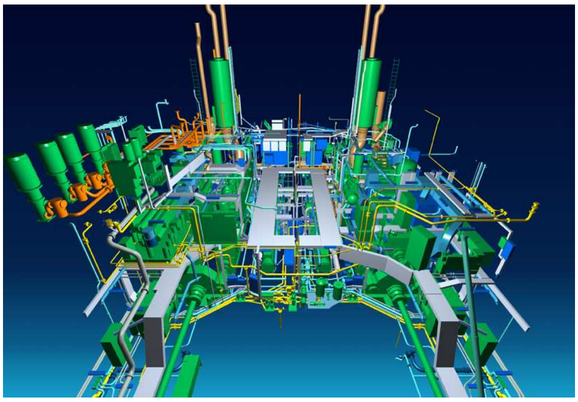


Fig. 1. Pipe layout example (taken from NUPAS Cadmatic internet home page).

Without such tool, the outcome of the overall process relies on the designer's knowledge and experience. Further, it is not possible for any person to define all pipe and cable routes at the same time, which raises integration and optimization problems. This is in fact a serious problem for a large number of new constructions.

This text describes a pipe and cable routing tool that can be used in the detail design stage in order to reduce man-hours or during basic design stage to identify potential problems that usually are only detected during construction. The aims of such tools have been defined previously by several authors [2] as follows:

(a) to minimize user input and decision;
(b) to make the system easy to use;
(c) to incorporate both pipe and cable routes;
(d) to be used in real shipyard design process.

2. Literature review

The need for a tool to help design the piping layout has been well-known for a long time. Back in the 80's decade shipyards usually built small models where straws were placed to check for collisions between different system components. With the 90's the first software tools that provide 3D visualization and collision check between objects started to appear, integrating a large amount of corporate knowledge and a way to automatically produce material lists and several documentation such as isometric drawings for production. Nevertheless, this is a different problem from the one this paper is concerned with, since no commercial fully automated system to support the designers in the routing selection of the different circuits is known to exist.

Some of the first research on PLD, as defined previously, was dedicated to the design of power plants and chemical refineries, as well as ships and submarines, though the problems are slightly different. During the last two decades several studies have been presented about this subject, trying to solve 2D and 3D problems, with and without pipe ramifications, using different algorithms and constraints.

Asmara and Nienhuis [3], divide the approaches into: roadmap search, cell decomposition approaches, potential field methods and mathematical programming methods. Probably, the most common one is the cell decomposition approach, consisting in the workspace division into different cells, complemented with mathematical programming methods, which deal with the path layout as a standard optimization problem with constraints.

This approach was followed by Zhu and Latombe [4], where the 2D and 3D problems were solved considering that each path had only one start point and one goal, though several paths had to coexist in the same workspace. In a first stage, the workspace was divided into rectangular cells taking into consideration existing obstacles. Afterwards, a search for the shortest paths connecting all start and goal points was performed using the A* (Branch and Bound) algorithm [5]. The order by which the paths are calculated is critical, and some of branches of A* will fail to find any solution at all. In order to solve this problem and to implement shape and location constraints, a path evaluation technique was implemented to decide the order in which the different routes should be defined, and then obtain sequentially a possible solution for all paths.

Ito [6] uses genetic algorithms to define and evaluate possible paths in a uniform cell decomposed space. His procedure, though containing many constraints, introduces the spatial potential energy concept, attributing a score to each cell depending on whether the path is intended to go through it. As an example, if it is established that pipes should go along walls, a lower value of potential energy is attributed to the cells near them, while the cells containing objects have a very high potential energy. In the end of the process, the possible paths are evaluated using a fitness function, choosing the solution where the sum of the potential energy for all routes in the problem is the smallest, which in fact already includes the shortest path requirement.

Park and Storch [7] differ from the previous references by proposing a cell decomposition method of non-uniform cells, whose shape and size is defined by grouping different pipes into common routes and considering pipe ramifications. This idea stems from the analysis of fabrication costs and operabibility, which are also used in the evaluating process of candidate paths, implemented by a decision tree.

More recently, [3, 8] presented the DelftPipe tool that aims to solve the PDL for pipes in a ship. The tool consists of an interface with commercial CAD systems, a pipe routing tool and an optimizer tool. As far as the pipe routing tool is concerned, it starts by implementing a non-uniform cell decomposition method, and then implements Djikkstra’s shortest path algorithm to select the different candidate routes taking into account pipe ramifications and the order in which each circuit is routed. The third stage uses the discrete particle swarm optimization algorithm to find a solution that complies with the performance criteria of reducing the path length and number of bends, as well as the standard criteria imposed by international rules or by the rules of the classification society according to which it is being built.

3. AISROUTE – decision support tool for pipe and cabling layout design

The AISROUTE is the decision support tool that aims to help the designer in the task of routing pipe circuits and electrical cabling, free from collisions, along a new vessel, once its general arrangement, structure and equipments locations have been defined.

This tool was built using Matlab and is similar to the one presented by the Delft University group, in the sense that it is modular; however the algorithms and solutions found are quite different from the ones presented in [3, 8].

Figure 2 shows the decision support tool block diagram, where all the different stages of the process can be seen:

(1) The "User Interface" imports the geometry from a commercial CAD program, where the different objects (equipments and structure) are placed. This action defines the workspace and some of the location constraints to paths;

(2) Next, the designer must specify all start and goal points (requirements) for the different paths, including pipe, cable routing and accessibility paths. On the other hand, for each path the different constraints (shape and location) should also be set, in order for them to be implemented afterwards;

(3) Routing begins by finding several "shortest path" solutions that fulfill the start and goal points' requirements and the workspace constraints, by changing the order of the paths to define first;

(4) The different solutions are then used as the initial population for a genetic algorithm implementation that evaluates each solution as far as all requirements and constraints are met. This evaluation is done by a fitness function that attributes penalties when any constraint is not fulfilled and makes use of the spatial energy concept of Ito [6] to reward the best paths;

(5) Finally, once a good solution is found, the "User interface" allows for its visualization and its export.

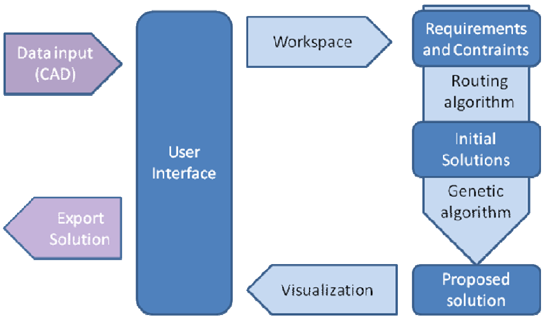


Fig. 2. AISROUTE block diagram

3.1 User Interface and workspace definition

The user interface allows the user to introduce the necessary data for routing, including the workspace geometry where the paths have to be defined, as well as their requirements (start and goal points) and different constraints. It was built in a windows environment using Matlab and Java objects, and it is able to import ASCII STL files that almost all CAD software is able to export.

The first step to work with AISROUTE is to define the space geometry of a "compartment" containing all obstacles (equipment and structure), that will stand for the location restrictions for the different paths. Each one of the obstacle must then be imported from CAD software. Figure 3 shows the user interface window, where some solids and three routes have been added.

Internally, the space is decomposed into cubic cells, whose faces have the same area as the smallest section area of the circuits to be routed. The coordinates of each cell form a matrix of X × Y × Z (longitudinal/ transverse/ vertical positions). Afterwards, when the obstacles are imported, the 3D matrix cells related to their position are marked as occupied.

This matrix is named the workspace, and it will be the platform over which the routing process is going to take place. During this process, the routes connecting the different starts and goals are searched through the matrix elements and the ones selected as part of each path are filled up. Once the process is completed, a new solution matrix, that includes not only the workspace but also the routes, is found. This procedure is going to be described in the next chapters.

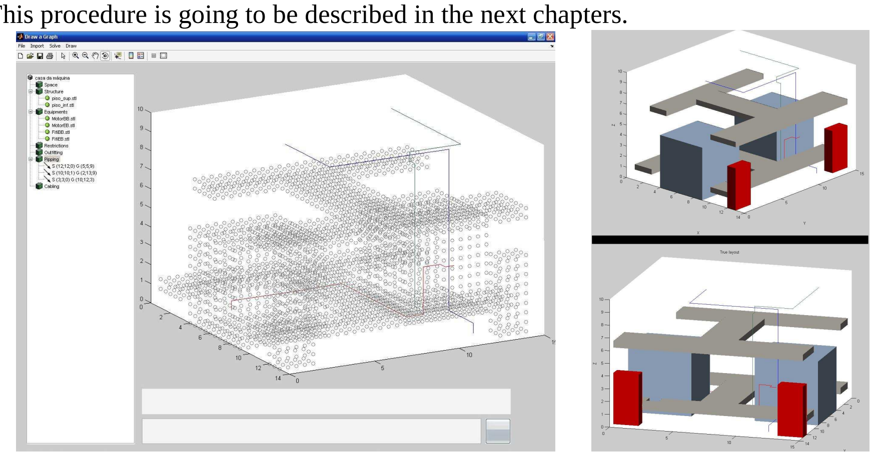


Fig. 3. AISROUTE user interface window (a), and visualization window (b)

Though this cell-decomposition procedure produces a large number of cells and causes an extra computer effort, these disadvantages may be overlooked in comparison with the advantages that they bring, namely:

(a) there is a direct relation between the spatial coordinates and the cells of the workspace and solution matrices;
(b) obstacles positions are marked in the workspace matrix and so the routes found will be collision free without need for subsequent collision check;

(c) it enables the use of maze solving algorithms for circuits' routing with and without ramification;

(d) it enables the implementation of different kinds of constraints and to select areas where there are advantages for the paths to go through;

3.2 Requirements and constraints analysis

Once the workspace is known there is the need to identify the start and goal points for each route, to identify the cells which can't be crossed by the path, and good areas for the path to go through. In other words, it is necessary to define the requirements and constraints for each path.

The constraints can be divided according to their nature into shape and location constraints [4], but to be able to implement them they were grouped into three different types:

(a) Type 1 – common location and shape constraints to all circuits, related to the obstacles of the workspace, including other paths, and to the fact that only 90º curves are allowed by the algorithm;

(b) Type 2 – shape and location constraints that exist only for a specific route, such as the case of gravity flow pipes, or the impossibility of going through some areas of the workspace (e.g. cables running through the floor);

(c) Type 3 – location constraints originated by the proximity to other routes. These ones are quite difficult to address since they are dynamic, and dependent upon previous path choices. As an example, it is intended that most pipes run together to be able to use common supports, however there are fluid systems that have to keep some distance between them due to heat sensibility, or electromagnetic interference between electric cables.

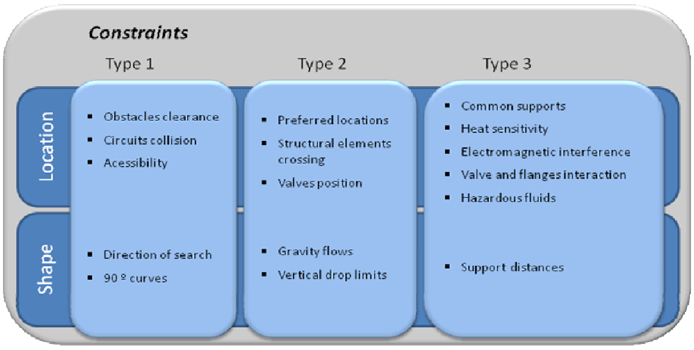


Fig. 4. Different kinds of constraints divided by their nature and type

The type 1 constraints will be present in all stages of the process, i.e. in both routing algorithm and genetic algorithm applications. On the contrary, the type two and three constraints are only going to be considered in the genetic algorithm implementation.

As far as the type two constraints are concerned, they are dependent upon the workspace and the obstacles within and though some are pre-defined within the program, they can be easily changed or new ones can be set by the user.

Nevertheless, the quality of the routing process is dependent upon the type three constraints. These are the ones that will allow for an easier operation of the ship and a more reasonable layout, though they are the most difficult to implement and to define. Unfortunately these are also the ones that are not yet fully developed in this work. Some of the constraints that should be considered are shown in **Fig. 4**.

3.3 Routing implementation using shortest path algorithms

The first approach to finding a good layout for all circuits is made using a shortest path algorithm presented by Lee [9] who, in its definition, is able to deal with the type one constraints.

To explain it briefly, it starts by flooding the neighbor cells of the start point in all six main directions (3 dimensional) up to when the goal point is reached, taking into account if the cells are already occupied by any obstacle. Then, the path is backtracked to the start point following a pre-defined search pattern. Although this method requires a high level of computation time, it was selected because it is easily changed to cope with different pipe/ cable tray sections (more than one cell) and main path ramifications. Additionally, the method always finds the shortest path, though it can find more than one, as presented in figure 5 (a).

The algorithm change to search for routes of circuits with several sections' shapes and sizes is trivial, once it is considered that to belong to the path, not only the target cell has to be free, but also her neighbors.

To implement the ramification case, it was assumed that besides the two points to be connected, let's say S1-G1, there is an extra one G2 that will be considered as a new start point. From G2 the flooding process is done up to the time when any of the cells of the path S1-G1 is reached (figure 5(b)).

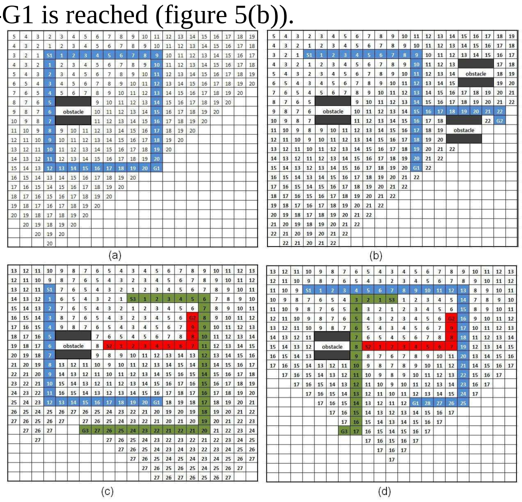


Fig. 5. Lee Algorithm routing examples for two similar shortest paths (a), path ramification (b), and different solutions for multiple routing dependent upon the order of paths (c) and (d)

In the figure 5 (c) and 5(d) examples, it can also be seen that when routing more than one circuit (let say S1-G1, S2-G2, S3-G3 – blue, red, green), there are several different path combinations that meet all requirements and that are found using the same Lee algorithm but changing the order in which the path search is done. As an example the paths found when starting to route S1-G1 and S2-G2 (figure 5(c)) are very different from the ones found starting by S3-G3 (figure 5(d)).

This is the way how the first set of solution matrices are found, taking in account for the definition of the routing order the following:

(a) the first route that should be considered is the "accessibility" that connects all different equipments/ objects defined in the workspace;
(b) the paths with larger sections should be routed first, in order to have less curves and a more basic routing style [7];

3.4 Genetic algorithm implementation

The genetic algorithm is used to evaluate an initial set of solutions found by the previous step and then to change them in accordance with the types two and three constraints. Their implementation in this area, as mentioned before, was first presented by Ito [6].

Genetic algorithms are inspired by the principles of the evolution of species, as described in [10] or [11]. The procedure begins by selecting a set of possible solutions named parents, whose characteristics are their chromosomes. These solutions are then used to obtain a group of children (new set of possible solutions), in an iterative form, using techniques called reproduction, which may follow three different processes:

(a) elitism – the best parents chromosomes are fully transmitted to their children;
(b) crossover – a new child is obtained mixing the chromosomes of two parents, using some of the genes of each parent;
(c) mutation – change of some of the genes of one parent in a repeated way through the iteration process.

The evaluation of each parent/ child is done by a fitness function that incorporates all requirements and constraints, using a penalty system for the ones that do not complete the requirements or do not fulfill the constraints. The aim is to have the less penalties as possible, i.e. the smallest fitness value.

In this work, each 3D solution matrix is considered to be the chromosomes with each matrix element (corresponding to a cell) being the gene that can be occupied or not. Accordingly the initial set of solution matrices found using the shortest path algorithm by changing the routing order are the first parents (initial population).

The process starts by attributing a spatial potential energy value, dependent upon the constraints and the circuit that is being routed, to each one of the matrix elements/ cells/ genes [6]. Afterwards, using the reproduction techniques new solution candidates are found and evaluated, aiming to find the paths with the lowest potential energy, found by adding up all values of the cells belonging to the path. Further, if the start and goal point are unconnected an extra very large value of energy is added (penalty).

As an example, figure 6 represents this procedure in 2D. Let it be assumed that the ramified path (blue path) has already been defined. It is intended that the red path does not cross any obstacle (type one constrain - black), stays away from the space limits (type two constraint - brown) and its path should be close to the blue path to use the same supports (type three constraints – light blue). A possible potential energy distribution is presented in figure 6(b) which may result in the solution presented in figure 6(a).

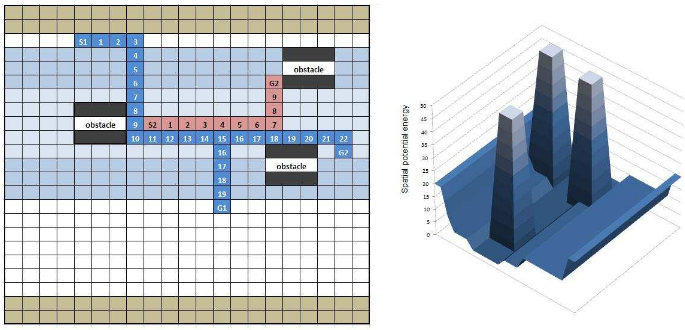


Fig. 6. Use of spatial potential energy concept in the routing of two pipes

So far, a number of preliminary tests have been performed on different configurations of compartments. The results have been very satisfactory, achieving good quality results in reasonable time. We are currently applying this software in a real case, and evaluating the results against the standard available solutions.

4. Conclusion

A method for the pipe and cable tray layout design was presented using a combination of routing algorithms [9] and genetic algorithms [10, 11], making use of the spatial potential energy concept [6].

This is thought to be a reasonable method as long as the means to define the potential energy values for each cell depending upon the circuit are well defined. In fact, this is the most difficult and yet less completed part of the work.

All in all, a decision support tool the help designers in defining the auxiliary circuits within a compartment is working and available to be tested, waiting to find all unavoidable errors in the implementation.

References

1. NUPAS, Nupas Cadmatic home page - *http://www.nupas-cadmatic.com/*. 2009, Numeriek Centrum Groningen B.V. and Cadmatic Oy.
2. Kang, S., S. Myung, and S. Han, A Design expert system for auto-routing of ship pipes. Journal of Ship Production, 1999. **15**(1).

3. Asmara, A. and U. Nienhuis. Automatic piping system in ship. in COMPIT - 5th Int. Conf. on Computer and IT Application Mar. Ind. 2006. Leiden.
4. Zhu, D. and J. Latombe. Pipe Routing = Path Planning (with many constraints). in IEEE International Conference on Robotics and Automation. 1991. Sacramento, California.
5. Hart, P.E., N.J. Nilsson, and B. Raphael, Correction to "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". SIGART Newsletter, 1972(37): p. 28-29.
6. Ito, T., A genetic algorithm approach to piping route path planning. Journal of Intelligent Manufacturing, 1999. **10**(1).
7. Storch, R. and J.-H. Park. Pipe-routing expert system. in International Conference on Computer Applications in Shipbuilding. 2002
8. Asmara, A. and U. Nienhuis. Automatic Piping System Implementation: A Real Case. in COMPIT - 6th Int. Conf. on Computer and IT Application Mar. Ind. 2007.
9. Lee, C.Y., An Algorithm for Path Connections and Its Applications. IRE Transactions on Electronic Computers, 1961. **10**(2): p. 364-365.
10. Holland, J., Adaptation in Natural and Artificial Systems: an introductory analysis with applications to biology, control, and artificial intelligence. 1975, Ann Arbor: University of Michigan Press.
11. Fogel, D.B., Evolutionary Computation : Towards a New Philosophy of Machine Intelligence. 2nd ed. 1999: IEEE Press.

Predicting the Outcome of Mutation in Genetic Algorithms

Sandeep Rajoria[1], Carlos Soares[2], Jorge Pinho de Sousa[3], and Joydip Dhar[4]

[1] ABV-Indian Institute of Information Technology and Management, Gwalior, India
[2] LIAAD-INESC Porto LA/Faculdade de Economia, Universidade do Porto, Porto, Portugal
[3] INESC Porto LA/Faculdade de Engenharia, Universidade do Porto, Porto, Portugal
[4] Faculty of Applied Sciences, ABV-Indian Institute of Information Technology and Management, Gwalior, India
sandeep.rajoria@iiitm.ac.in, csoares@fep.up.pt, jsousa@inescporto.pt, jdhar@iiitm.ac.in

Abstract. The general goal of our work is to develop *smart operators* that incorporate Machine Learning methods to improve the search of GAs (or EC, in general). As a first step in that direction, this paper investigates if is it possible to learn to predict the behavior of the swap mutation operator on Job-Shop Scheduling problems. For that purpose, we generate all possible operations and assess their effect on the quality of the solutions. We apply a learning algorithm to obtain a mapping between a description of the operations and that effect. We obtain good results even using relatively small training sets.

1 Introduction

Genetic Algorithms (GA) or, more generally, Evolutionary Computation (EC) methods, are widely used in Optimization [2]. Within the EC framework, the user may tackle an optimization problem using off-the-shelf methods, such as a binary GA. In this case, the user must design methods to translate problem solutions into binary strings (encode) and to translate these back into solutions (decode). Any implementation of a binary GA can then be used to carry out the optimization process. On the other hand, EC also enables a radically different approach, by tailoring part or the whole optimization process to the problem. In this case, the user may work with any kind of representation of the solutions (ranging from simple numerical vectors to very complex structures, such as graphs and trees) but appropriate operators must then be developed (or adapted) to carry out the search.

The flexibility of the EC framework encourages the development of new operators. These can be general operators, that depend only on the representation, or more specific ones, that are only suitable for a small set of problem types. This flexibility creates an appealing opportunity to employ Machine Learning methods [5].

The general goal behind this research is to develop *smart operators* that incorporate Machine Learning methods to improve the search of GAs (or EC, in general) so as to make it more efficient and faster. These operators should be able to decide how to behave based on previous experience. But before developing such an operator it is necessary to investigate the following question: *is it possible to learn to predict the behavior of GA operators?*.

In this paper, we perform one such study. Our goal is to test whether it is possible to predict the outcome of the swap mutation operator of a GA applied to the problem of Job-Shop Scheduling. We systematically generate mutation operations and evaluate their effect on the quality of the solutions. Then, we characterize operations using a set of measures and, finally, we use learning methods to obtain a mapping between the characteristics of operations and their effect.

We start by describing the Job-Shop Scheduling problem and the mutation operation which is investigated in this work (Section 2). Next, we describe our approach in detail, focusing on the process of generating the data to be used for learning (Section 3). In Section 4, we present some experimental results. Finally, we present some conclusions and future work, namely how a *smart operator* can be built based on the learning approach described in this paper (Section 5).

2 Job-Shop Scheduling Problem and Swap Mutation

We start by informally describing the Job-Shop Scheduling problem. More detailed descriptions can be easily found in the literature (e.g., [6]). Then, we describe the swap mutation operator, which is the object of this analysis.

2.1 Job Shop Scheduling problems

In summary, the Job Shop Scheduling (JSS) problem can be defined as follows:

- A finite set of n jobs
- Each job consists of a chain of operations
- A finite set of m machines
- Each machine can handle at most one operation at a time
- Each operation needs to be processed during an uninterrupted period of a given length on each machine

The goal is to find a *schedule*, that is, an allocation of the operations to time intervals on machines, that optimizes some criterion. In our experiments, the evaluation criterion is the *makespan*, i.e., the end time of the job finishing the latest.

Thus from the above definition we can figure that the most basic and simple JSS problem consists of:

Precedence matrix This basically consists of the precedence constraints, representing the order of the machines in which the jobs have to be processed.

Duration matrix The time required for the jobs to be processed in the machines.

Table 1 contains an example of a 3x3 JSS problem, i.e., 3 machines and 3 jobs.

Table 1. An example of a 3x3 problem

	Duration			Precedence		
	Machine1	Machine2	Machine3	Machine1	Machine2	Machine3
Job1	10	24	11	3	2	1
Job2	64	31	95	2	3	1
Job3	9	12	30	1	3	2

2.2 Representation

A chromosome is an unpartitioned permutation with m repetitions of the n jobs [1]. An example of a solution for a 3x3 (3 Machines and 3 Jobs) JSS problem is

1,3,3,2,1,2,3,1,2

which actually will mean that:

- the first operation to be scheduled is the first of job 1 (which, according the precedence matrix, turns out to be on machine 3);
- the second operation to be scheduled is the first of job 3;
- the third operation to be scheduled is the second operation of job 3, as 3 appears for the second time;
- and so on...

A schedule is generated by decoding a solution. We use a simple algorithm which allocates time slots on the machines to operations, strictly following the order in the chromosome. The steps used in the whole process of decoding can be summarised as follows. For $i \in 1 \ldots n \times m$:

1. $j \leftarrow i^{\text{th}}$ element of the chomosome. The value of j represents the job to which the operation that we are scheduling belongs.
2. $o \leftarrow$ number of operations of j that have been already scheduled, i.e., the number of times j is in the chromosome before the i^{th}.
3. $p \leftarrow$ machine in which the o^{th} operation of job j should be executed (from the precedence matrix).
4. $d \leftarrow$ execution time required by the operation of j to be executed oon machine p (from the duration matrix).
5. $s \leftarrow$ max(end time of the latest operation of job j that was already scheduled, end time of the latest scheduled operation on machine p).
6. Schedule the operation of job j on machine p from time s to $s + d$.

This algorithm generates valid schedules, i.e., schedules which do not violate any of the constraints. However, it may generate schedules which unnecessarily delay operations. For instance, let us consider two operations on the same machine, where operation i follows operation j in the chromosome. Using this algorithm, i will be scheduled after j, even if it the corresponding machine has a free slot before operation j which is long enough to schedule i (and assuming that the remaining constraints are not violated). Other algorithms exist that do not have this shortcoming (e.g., generate *active schedules* [3]). However, the choice of decoding algorithm is not essential for the purpose of this work.

2.3 Swap Mutation

Our work is focused on the swap mutation operator, which is commonly used in permutation-based representations [2]. It simply consists of generating a new chromosome by randomly swapping two elements from an existing one. An example of the swap mutation operator is given in Table 2.

Table 2. An example of the swap mutation. The selected positions are 2 and 4

Chromosome	1,3,3,2,1,2,2,1,3
Mutated chromosome	1,**2**,3,**3**,1,2,2,1,3

3 Prediction of the Outcome of Swap Mutation

As stated earlier, the goal of this work is to test the hypothesis of whether it is possible to learn models that are able to predict the behavior of GA operators and, in particular, of swap mutation. Two important characteristics of this problem are:

- the function we are trying to learn is deterministic, unlike most learning problems. The outcome of an operation is always the same.
- the universe of examples is finite and can be systematically generated. The universe of solutions of a $n \times m$ problem is the set of all unpartitioned permutations of n elements with m repetitions. The universe of all operations is obtained by testing all possible pairwise swaps of elements for each solution in the universe.

The size of the universe of operations is, naturally, very large, except for very small n and m. But, in the latter case, it is possible to generate all the possible operations, and, thus, investigate our hypothesis thoroughly, as will be described.

The methodology proposed is the following:

1. Generate all possible operations and estimate their effect on the quality of the corresponding solutions, which is the *target variable* (Section 3.1).

2. Generate a description of each of the operations (i.e., a set of *features*) (Section 3.2).
3. Use learning methods to find a mapping between these features and the target value.

Here the target variable we are dealing with is the effect of the mutation operator on the makespan of the solution. So the function that we are looking for maps the inputs, i.e., the features of the operation (x_i) to the output, i.e., the variation in the makespan of the chromosome ($\Delta makespan$). It can be represented as:

$$\Delta makespan = f\left(x_1, \ldots, x_k\right) \tag{1}$$

where k is the number of attributes. The target variable is numeric, which means that this is a regression problem.

For the development of smart operators, it may be sufficient to know simply if the operation is going to improve the quality of the solution or not. In this case, the problem can be addressed as a classification problem. The class of each operation can be determined simply by determining the sign of the variation in makespan:

$$class = sign\left(\Delta makespan\right) \tag{2}$$

Only if it is possible to generate such a mapping successfully, can we move on to build a smart swap mutation, which can guide its operation based on a model of its past performance.

Experimental results of this approach are presented in Section 4. Before that, we describe in mode detail the process of generating the target values.

3.1 Target Values

Examples are generated as follows:

1. Take one of the instances and generate all the possible chromosomes, i.e., all unpartitioned permutations of n elements with m repetitions. In a 3x3 problem, this amounts to a total of $\frac{(3*3)!}{3!*3!*3!} = 1680$ chromosomes. Then, we compute the fitness of each of the chromosomes.
2. For each one of the chromosomes, find out all the operations that is possible to carry out with the swap mutation. For a 3x3 problem, there are $18 + 9 + 0 = 27$ possible swap mutations for each solution that exchange alleles with different values.
3. Determine the target value for each operation. This is computed as the difference in the fitness of the chromosome that is generated by the operation and the fitness of the original chromosome. In a 3x3 problem, we have a total of $1680 * 27 = 45360$ operations and, thus, of fitness variations. Table 3 illustrates the generation of the target value.

Table 3. An example of the process of generating the target value

	Chromosome	Fitness
Original	1,2,2,3,2,3,1,1,3	208
Mutated	**2**,2,2,3,**1**,3,1,1,3	129
Variation		-79

3.2 Features

To be able to predict the effect of an operation on the quality of the corresponding solution, we need to describe those operations using predictive features. This means that the features must contain information that is useful to determine that effect which we are trying to predict. Additionally, the features can only use information that is available before the operation is executed. Otherwise, there is no need to predict its outcome. The information that is available to be used in the development of the features is the following:

Problem :
- Precedence of the machine in the jobs (i.e., the precedence matrix)
- Execution time of the jobs on the machine (i.e., the duration matrix)

Individual :
- Order of the jobs for scheduling (i.e., the chromosome)
- Fitness of the individual

Operation : In the case of swap mutation, the information is can be used is
- Position of the swaps

Many features can be generated based one of these types of information or combinations. In this study we have generated the following set of features:

- J1 & J2 - Jobs to which the operations being swapped belong.
- D1 & D2 - Execution time of all the operations in the job before the current one, for J1 and J2.
- O1 & O1 - Order of the operation in the job, for J1 and J2.
- M1 & M2 - Machines in which the operation should be executed, for J1 and J2.
- MOJ & MAOJ - The job which is numerically smaller and bigger, respectively.
- NTJAB1 & NTJAB2 - Number of jobs that have occurred before the job, for J1 and J2.
- TMJ1 & TMJ2 - Total execution time of all jobs on the machine used by the operation, for J1 and J2.
- DBP - Difference in the position of the operations on their respective machines.
- TTBMTMJ1 & TTBMTMJ2 - Ratio of the total execution time of all jobs on the machine used by the operation (TMJ1 or TMJ2) and the maximum total execution time of all the machines, for J1 and J2.
- SM - Whether the operations share the same machine before and after swap.

- NM1 & NM2 - New machines in which the operation should be executed after the swap, for J1 and J2.
- SNM - Whether the new machines are same.
- TTOMBTTNMJ1 & TTOMBTTNMJ2 - Ratio of the total execution times of all jobs on the old (before the swap, TMJ1 or TMJ2) and new (after the swap) machine used by the operation, for J1 and J2.
- RJMPJ1 & RJMPJ2 - Ratio of the number of jobs on the machine used by the operation, before and after it, for J1 and J2.
- NSOB1 & NSOB2 - Number of operations on the same machine of the operation in between the swap positions, for J1 and J2.
- DPM - Difference between the positions of the jobs J1 and J2 on the machines of the corresponding positions.
- TPJMJ1 & TPJMJ2 - Total execution time of the jobs executed on the same machine used by the job, that are scheduled before it, for J1 and J2.
- MC - The mean correlation between the execution time of the jobs on the machines, for all the jobs.

4 Experimental Results

Given that we generate all possible operations, we need to work with small problems. We have generated ten problems with three jobs and three machines (3x3). We implemented a problem generation method which is commonly used in Operations Research [6].

The smart operators we plan to develop will generate models based on operations that were previously executed. The number of operations that is available for this purpose is necessarily small when compared to the universe of all possible operations. Although our goal at this stage of the work is simply to investigate whether it is possible to predict the effect of an operation, we should use a relatively small number of examples to be able to obtain more reliable conclusions.

Additionally, we wish to assess how small a dataset we can use and still obtain satisfactory models. Therefore, we did experiments using 20%, 10%, 9%, 8%, 7%, 6%, 5%, 4% and 3% percent of the data for training and the remaining data to estimate the generalization error (i.e., as test set).

The algorithms that we used for the purpose of model generation are given in Table 4. More information about these algorithms can be found in textbooks on Machine Learning and Statistics (e.g., [5, 4]).

Table 4. Learning algorithms

Regression		Classification	
LR	Linear Regression	LD	Linear Discriminant
RT	Regression Tree	DT	Decision Tree
SVM-R	Support Vector Machines	SVM-C	Support Vector Machines
RF-R	Random Forest	RF-C	Random Forest

The results are presented in Table 5 and Figure 1. Several interesting observations can be made. Firstly, these results indicate that the hypothesis underlying this study is true: it is possible to predict the effect of swap mutation. The best result in the regression problem is an NMSE of 0.16 and in the classification problem is an error rate 5%.

Table 5. Results of predicting the effect of swap mutation, varying the amount of training data. The evaluation measure for classification is the error rate and for regression is NMSE

	Regression				Classification			
%	LR	RT	SVM-R	RF-R	LD	DT	SVM-C	RF-C
20	0.59	0.55	0.29	0.16	0.18	0.16	0.10	0.05
10	0.59	0.56	0.33	0.23	0.19	0.16	0.12	0.07
9	0.59	0.55	0.35	0.24	0.19	0.16	0.12	0.08
8	0.59	0.57	0.35	0.26	0.19	0.16	0.12	0.08
7	0.59	0.55	0.36	0.27	0.19	0.16	0.13	0.09
6	0.59	0.56	0.38	0.29	0.19	0.17	0.13	0.09
5	0.59	0.56	0.39	0.31	0.19	0.16	0.13	0.10
4	0.60	0.57	0.41	0.33	0.19	0.16	0.14	0.10
3	0.60	0.57	0.44	0.36	0.19	0.17	0.15	0.11

Although the numbers cannot be compared directly, we observe that the results obtained in the classification problem seem to be much better than the ones obtained in the regression problem. This means that we are better able to predict whether an operation is going to improve the quality of the solution or not, than the value of the variation (i.e., by how much the solution improves or worsens).

Additionally, the best results are obtained with the Random Forest algorithm. The results clearly show that Support Vector Machines are also doing pretty well. However, we note that both of these algorithms are computational more expensive than the other ones. This cost may be relevant in the context of the optimization algorithm, although we have not investigated this issue yet.

The lines in the plot are almost horizontal, which means that the effect of reducing the size of the training set is not very big. However, on closer inspection, we observe that it is stronger on the algorithms that achieve the best results. In fact, for the Random Forests the error doubles both in the classification and the regression problems. However, it is still low (10% and 0.36, respectively). One interesting result is that the performance of the Discriminant Analysis methods and simple Tree-based models are almost not affected by the training set size.

The variance of the results is also quite low, which indicates that the approach is robust (Figure 2).

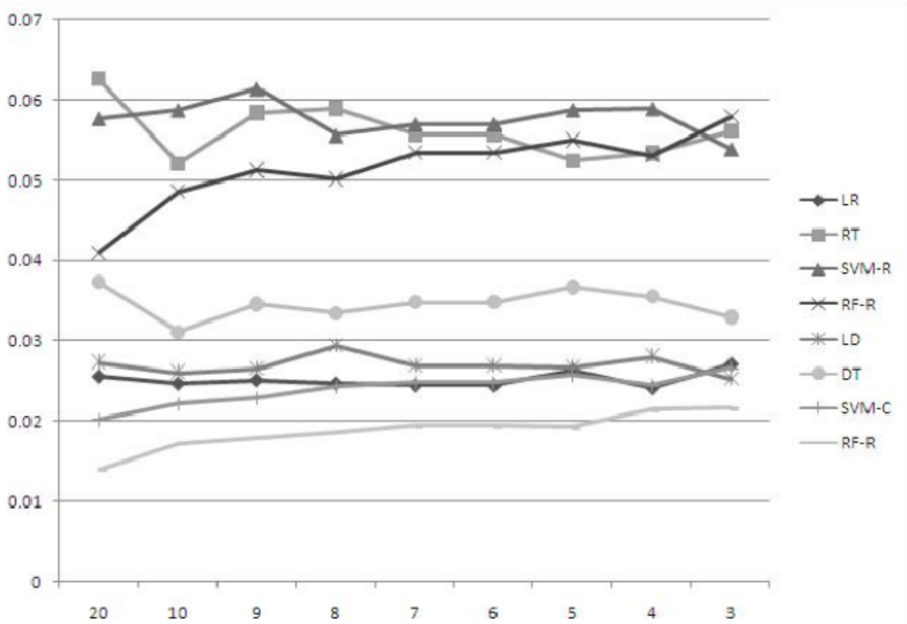


Fig. 1. Results of predicting the effect of swap mutation, varying the amount of training data

5 Conclusions

In this paper, we present a first set of experiments to test whether it is possible to predict the effect of the swap mutation operator in optimization with Genetic Algorithms. The optimization problem addressed is the Job-Shop Scheduling problem.

Our results show that the problem is learnable, even using relatively small training sets. This indicates that it is possible to develop a smart swap mutation operator that decides how to mutate a chromosome based on the effect of previous operations. This smart operator must start with a model based on a small set of operations but it could periodically (say, every 100 operations) update the model. Here we have used larger samples, so it remains to be evaluated the effect of using such small training sets on the quality of the results.

Such a smart operator can be regarded as a greedy operator. Therefore, we note that although it is possible to build it, this does not necessarily mean, when integrated in the search process of a GA, it will contribute to better solutions. Additionally, it must be taken into account that the smart operator described will only be useful if it is not computationally very expensive relative to the optimization process. These issues must be verified empirically. Additionally, it is possible to develop hybrid operators, which sometimes make decisions based on the learning model and sometimes will choose randomly.

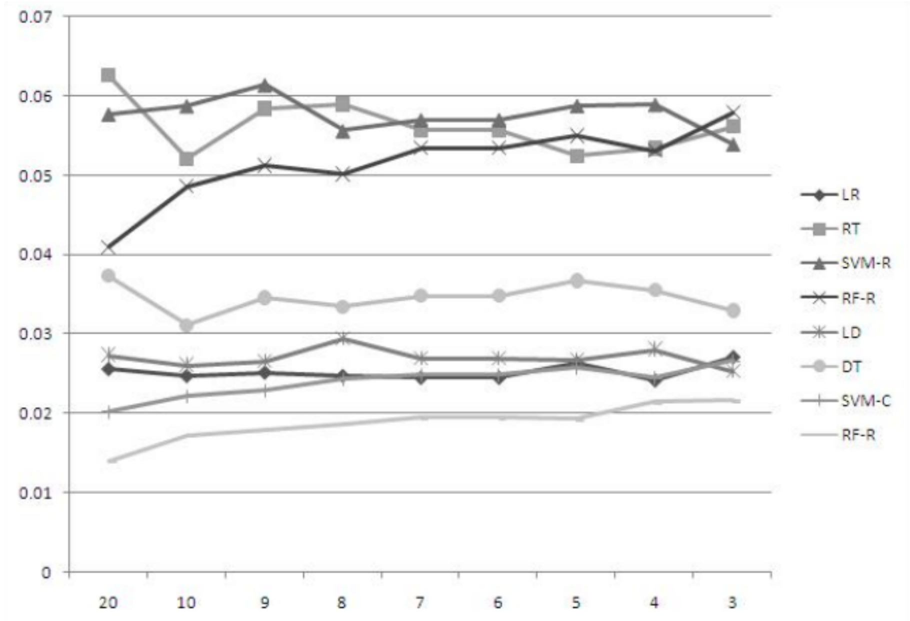


Fig. 2. Variance of the results of predicting the effect of swap mutation, varying the amount of training data

Despite the good results obtained, we believe that it is necessary to make further tests before using this approach to develop smart operators. One important issue is the size of the optimization problems. Our experiments so far were on very small problems. We need to test this approach on larger problems. Naturally, it will not be possible to generate all solutions, so we will work with samples.

Finally, the simplicity of the swap mutation makes it a good candidate to test our hypothesis. It would be interesting to test whether it is possible to use this approach to other operators, in particular to more complex ones, such as crossover. The challenge lies in the design of the features used to describe operations.

References

1. Christian Bierwirth, Dirk C. Mattfeld, and Herbert Kopfer. On permutation representations for scheduling problems. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature – PPSN IV*, pages 310–318, Berlin, 1996. Springer.
2. A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing (Natural Computing Series).* Springer, November 2003.
3. B. Giffler and G.L. Thompson. Algorithms for solving production-scheduling problems. *Operations Research*, 8(4):487–503, 1960.

4. Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning*. Springer, August 2001.
5. T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
6. E. Taillard. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64:278–285, 1993.

Chapter 3

CMBSB - Computational Methods in Bioinformatics and Systems Biology

Approximate 3D Motif Search in Proteins with Domain Specific Knowledge

Arno Formella[1], Thorsten Pöschel[2], and Cástor Sánchez Chao[1]

[1] Universidad de Vigo, Spain
Departamento de Informática
formella@uvigo.es
[2] Universität Erlangen–Nürnberg, Germany
Excellence Cluster: Engineering of Advanced Materials
thorsten.poeschel@eam.uni-erlangen.de

Abstract. We present three heuristics including the usage of domain specific knowledge to improve a general purpose algorithm for the 3D approximate point set match problem and its application to the task of finding 3D motifs (like surface patterns or binding sites) in proteins. The domain specific knowledge and further heuristics are used, under certain conditions, to reduce the run time for the search and to adapt the number of reported matches to the expectations of the user. Compared to the general purpose algorithm, the new version is twice as fast, and can be further improved especially for small tolerances in the matches by means of analyzing the distance distributions of the atoms.

1 Introduction

In recent years proteins are compared and analyzed in all of their representations: as primary structures, i.e., their bare sequences of amino acids, as secondary structures, i.e., their principal folds mostly in α–helices and β–sheets, and as ternary structures, i.e., their complete three–dimensional appearances. The definition of similarity in the molecular context always depends on the scientific question under investigation, especially when one takes into account that proteins are not really rigid structures, rather they exhibit certain amount of flexibility due to temperature, solvents, and other factors. In many research areas, such as docking and functional analysis, it becomes more and more important to compare the 3D structure of the molecules [14].

There is quite a range of algorithms and programs available to the scientific community, both commercial and free, that cover different aspects of protein comparison, analysis, and fold prediction, see [6, 10, 12] for comprehensive overviews of different approaches. They include backbone alignment (based on the C_α–chains), secondary structure elements alignment, and sequence–based alignments, which either perform pairwise alignment or multiple structure alignment. Many of them are hosted at publicly accessible web–servers such as [1, 7, 11] and many more.

In this article, we describe an extension of the program **psm** [4]—a program being already included in a public web–based search facility [1] and especially useful to identify small components in large arrangements, for instance, similar surface or binding site structures, loops and hinges etc. within large proteins—with some new features that, under certain conditions, improve its performance and allow for a more adaptive search. The core algorithm of **psm** searches for all occurrences of a small 3D point set, e.g., atoms, in a large 3D point set, e.g., the macro–molecule. Other algorithms dedicated to a similar task include PINTS [13] and needle–haystack [8].

The rest of the article is organized as follows. Section 2 briefly describes the general purpose approximate point set search algorithm. Section 3 introduces the proposed heuristics to improve the run times of the algorithm. Section 4 gathers some preliminary results that we obtained with comparable implementations of all proposed methods. Section 5 states some future actions that can be carried out and, finally, Section 6 summarizes the main contributions of the work.

2 Approximate point set match

Let us, for a moment, forget the specific task that we are searching for 3D motifs in proteins. Rather, we concentrate on the more abstract problem of searching for a small 3D point set within a large 3D point set. Hence, given a small (both in diameter and number of points) set of points as search pattern and a large set of points as search space, find all locations within the search space where the search pattern can be placed best according to a given distance metric and applying a given kind of geometric transformation.

More formally: Let $P = \{p_1, \ldots, p_k\} \in \mathbb{R}^3$, $|P| = k > 1$ be a finite set of 3D points, the search pattern. Let $S = \{s_1, \ldots, s_n\} \in \mathbb{R}^3$, $|S| = n > k$ be a finite set of 3D points, the search space. Find an injective matching function $\mu : P \longrightarrow S$ and a transformation $\tau : \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ such that $D(\tau(P), \mu(P))$ is minimum, where D is an appropriate distance measure between the point sets.

Usually, τ is taken as a rigid body transformation, i.e., a combination of a translation and a rotation. Further possibilities include reflection and scaling. Transformations with deformations (e.g., allowing for torsion or hinges) might be considered, too. Common distance metrics include, for instance, root mean square, maximum or average distance:

$$D(\tau(P), \mu(P)) = \sqrt{\frac{1}{k}\sum_{i=1}^{k} |\tau(p_i) - \mu(p_i)|^2}$$

$$D(\tau(P), \mu(P)) = \frac{1}{k}\sum_{i=1}^{k} |\tau(p_i) - \mu(p_i)|$$

$$D(\tau(P), \mu(P)) = \max_{i=1}^{k} |\tau(p_i) - \mu(p_i)|$$

Our algorithm for approximate point set match implemented in **psm** works as follows ([1, 4]): First, we build an undirected distance graph G_P over P, where the nodes of the graph are the points of P and the edges define a connected graph (for simplicity you might assume that G_P is the complete graph). Second, we build an undirected distance graph G_S over S considering only those edges, i.e., distances in S, being similar (see Section 3 below) to the edges in G_P. If the diameter of P is small compared to the diameter of S a regular grid over the search space is used to reduce the number of candidate edges in G_S.

Once the graphs are built, we apply a backtracking algorithm to find all subgraphs of G_S that match G_P both in graph structure and in edge distances. Let us explain the algorithm in some detail. Assume that the nodes in G_P are ordered in such a way that for all p_j, with $1 < j \leq k$, there exists an edge $\{p_i, p_j\}$ with $i < j$. Let G_P^i denote the subgraph induced by the nodes $\{p_1, \ldots, p_i\}$ of G_P; observe that $G_P^k = G_P$. All these k graphs are connected graphs and for each connected graph G_P such a sequence of graphs can be found (if G_P is the complete graph, any sequence of nodes defines such a sequence of connected graphs).

The backtracking algorithms starts with G_P^1 and tries to find iteratively matches for G_P^i in G_S with $1 \leq i \leq k$. Clearly, all nodes in G_S are candidate nodes to be matched to p_1. Now, assume we have already matched a subgraph G_P^i with $i < k$ and let $s_j = \mu(p_j)$, for $1 \leq j \leq i$, denote the nodes of G_S where p_j has been matched. We try to match the next node p_{i+1} in the sequence. All adjacent nodes of s_i that have not been matched so far and that exhibit similar distances to all corresponding nodes with smaller indices are candidates to be matched to p_{i+1}. If such a node of G_S cannot be found, i.e., G_P^{i+1} cannot be matched, G_P^i cannot be extended and hence backtracking takes place: we proceed with the next candidate for p_i. The backtracking algorithm eventually finds all possible matches of G_P in G_S.

For each match we compute the optimal transformation with the help of a direct method for RMS distance metric based on [9] or, for the other distance metrics, with the derivative free optimization algorithm taken from [5].

Let us analyze briefly the run time behavior of the backtracking algorithm. The subgraph matching problem itself is NP–complete. Assuming that the average degree of G_S is d, the worst case run time is $O(n(d \log d)^{k-1}(k-1)!)$, because in each extending iteration at most d neighbors have to be visited and $i - 1$ distance checks must be performed. The factor $\log d$ is due to the fact that we have to search for the corresponding edges in G_S. Storing the graph with an adjacency matrix (that would allow for constant time access) rather than using an adjacency list would make the memory requirements for practical cases prohibitive large. Note that G_P, for being assumed to be small, can be stored as adjacency matrix with constant access time for edge queries.

In practical cases, especially when applying the basic point set match algorithm to proteins, a much lower run time can be observed than stated in the worst case analysis. As described in the next section different heuristics can be used

– to reduce the remaining edges in G_S, hence the average degree d is reduced,
– to reorder the nodes in G_P, hence more confining edges are tested early, and
– to exploit problem specific restrictions, hence the number of candidate nodes and/or edges is reduced.

3 Search heuristics

Before going into details, let us first take a look at the similarity relation between two distances. We say a distance e is similar to a distance d whenever $e \in [d(1-\epsilon), d(1+\epsilon/(1-\epsilon)]$ with $0 \leq \epsilon < 1$. The upper bound can often be decreased to $d(1+\epsilon)$ in the case the symmetry of the search is not an issue.

The tolerance ϵ, with $0 \leq \epsilon < 1$, is a user provided value that determines to what extend the point set found in S can deviate from the pattern P. With $\epsilon = 0$ perfect matches are searched for. In [1, 4] ϵ was a fixed value for all edges. In the new implementation, we allow for a more flexible approach taking into account properties of a protein.

We present three heuristics to reduce the search time:

– One that works with the complete graph over the search pattern and reduces the number of remaining edges in G_S taking into account the distribution of distances being present both in P and S.
– One that uses different values for ϵ in the similarity relation depending whether the atoms defining the edge are located close or far on the protein chain.
– One that takes into account that certain chemical bonds within the protein are unlikely to be very flexible.

3.1 Rare distances first

Let the graph over the search pattern be complete. Once all distance intervals over the graph of the search pattern G_P are computed, we compute the distribution of those edges in G_S that fall into the intervals. The interval where the least number of edges of G_S are counted defines the, let's say, rarest distance. Clearly, we need to maintain in G_S only those edges that correspond to the rarest distance and all adjacent edges to their end nodes with the further restriction that the other end must have an edge with appropriate distance to the other end of the rare edge.

We compute the final graph G_S in four steps. First, we determine the distribution to find the rarest distance. Second, we mark all nodes in G_S that are adjacent to an edge similar to the rarest distance. Third, we mark all nodes in G_S that are simultaneously adjacent to the two end nodes of a rare edge. Note that the marks in the steps two and three are different. Fourth, we generate as graph G_S the graph induced by the edges between marked nodes.

With this possibly smaller graph, the substructure search with the backtracking algorithm is performed.

3.2 Flexible distance intervals

Proteins consist of one or various folded chains built as a sequence of amino acids. Usually, the farer two atoms are located on the chain, the larger the tolerance for their distance can be, because proteins are, to a certain degree, flexible structures. To take advantage of this fact in our algorithm, we use this knowledge to define different tolerances for different edges in the graphs.

We present a still very simple method to define such a flexible tolerance. Let Δ be the distance in amino acid count of two atoms in the search space, i.e., their sequence distance, often notated as $|i-j|$ for atoms i and j of the data set. Let M be the total number of amino acids in the search space. We define

$$f = \frac{\epsilon_f \cdot \Delta}{M-1} + 1$$

as factor to enlarge the tolerated distance for a match of an edge, i.e., we use instead of ϵ the value $f\epsilon$ in the similarity relation. The quantity $\epsilon_f \geq 0$ denotes a user specified value where care must be taken of that $f\epsilon$ does not exceed 1. To allow for a more user friendly way to introduce the additional tolerance ϵ_f with its influence on ϵ, we ask the user for a value $k \geq 1$ and compute

$$\epsilon_f = \left(1 - \frac{1}{k}\right) \cdot \left(\frac{1}{\epsilon} - 1\right)$$

or set $\epsilon_f = 0$ for $\epsilon = 0$. Hence, a value of $k = 1$ means no flexible tolerance, and the larger the value of k the larger the flexible tolerance will be. In other words, for $\epsilon_f = 0$ the search algorithm uses for all edges the same tolerance ϵ as stated above; the same is true for $\epsilon_f = 1$ and atoms belonging to the same amino acid; for other values of ϵ_f and/or atoms not belonging to the same amino acid, ϵ is enlarged with the factor f.

3.3 Exploitation of domain specific knowledge

According to [2] we take into account the mean distance for certain chemical bonds within the backbone of a protein. In more detail, the C–N distance in a peptide bond is typically 1.32 Å, the C–O distance in a carbon–oxygen bond is typically 1.24 Å, the C_α–C distance in a carboxyl group is typically 1.52 Å, and the C_α–N distance is typically 1.45 Å. Analyzing several proteins from the PDB suite (RCSB Protein Data Bank) [3], we confirmed the given mean values, and observed further a 2.232%, 2.3%, 2.332%, and 3.255% maximum increase/decrease in distance for these four chemical bonds.

We use a simple table look–up method to identify the corresponding value for the tolerance ϵ depending on the types of the atoms and their distance in amino acid count. The C_α–N bond is only present when both atoms belong to the same amino acid, the same is true for both the C–O and the C_α–C bonds. The C–N bond is present when the amino acid distance of the two participating atoms is one. To build the look–up table we assign to each atom an index i_a according to Table 1.

atom	any	C_α	N	C	O
i_a	0	2	4	8	14

Table 1. Atom indices according to their type.

If the amino acid distance Δ between the two atoms A_0 and A_1 defining an edge in the distance graph is 0 or 1, we compute a table index $i = \Delta + i_a(A_0) + i_a(A_1)$ which is used to address a table with 30 entries (see Table 2).

i	0–5	6	7–9	10	11–12	13	14–21	22	23–29
ϵ	0	0.03255	0	0.0232	0	0.2232	0	0.023	0

Table 2. Tabularized values for ϵ for pairs of atoms.

If the encountered entry in the table is 0, then we use the ϵ value given by the user (possibly taking into account the factor f as described in the previous section), if it is not 0, the table entry without modification is used as ϵ value for the specific edge.

4 Results

We present some preliminary results illustrating the improvements comparing the new version with the different improvements to the original **psm** software. The run times have been obtained in [seconds] on a GNU/Linux system with a Q9550 Intel processor at 2.83 GHz. **psm** is still implemented as a single threaded application.

As search space we used the protein 1IRU (hydrolase, mammalian proteasome) [15] with 47 589 atoms and an active site of the proteasome (see Figure 1). The active site contains 40 atoms distributed over at least four non–consecutive amino acids. The active site is a cutout of the proteasome which appears with certain modifications several times in the protein (see Figure 2). Whenever two matches in the search space share an atom, we consider the two matches belonging to the same cluster. **psm** reports to the user both the number of matches and the number of clusters encountered. A cluster shows where the motif is located in the protein, whereas the match itself shows in what constellation (and quality according to some pairwise distance function) the motif is present.

Figure 3 shows the preprocessing times (left plot) and the search times (right plot) for the **psm** version without any improvement (upper curves) and the same for the version where the domain specific knowledge has been taken into account, i.e., the distances between certain pairs of atoms as stated in Section 3.3 vary at most up to the tabularized percentages. The plots are drawn over the distance tolerance ϵ from 0 to 35%, i.e., the mismatch of the inter–atom distances

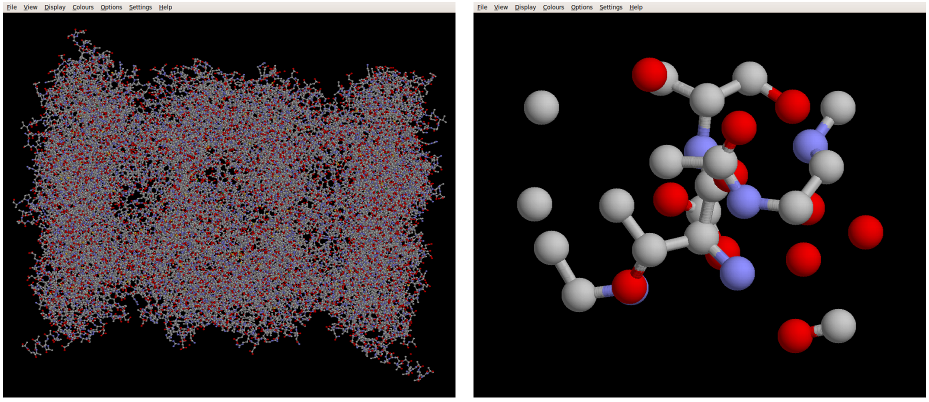


Fig. 1. Search space (proteasome) and search pattern for the test cases.

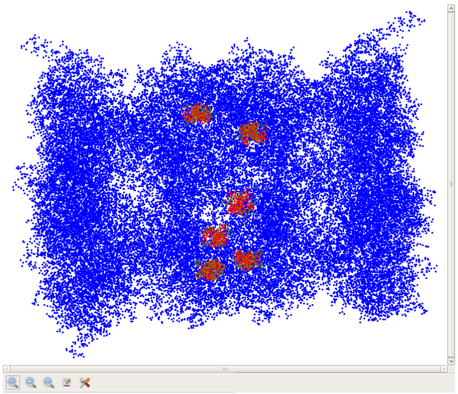


Fig. 2. Clusters found with **psm**.

comparing the search pattern to the encountered location in the search can be that large. As one can observe the preprocessing time is almost the same for both cases, however, the search time of the new version is half of the time of the old version.

If we look at the number of matches found by the two versions (see Figure 4), we see that including domain specific knowledge reduces the number of reported matches (note the logarithmic scale in the plot). However, it is important to mention that, at least in our current tests, the number of clusters and their location is for all values of ϵ exactly the same. We have observed the same behavior for all other tests described in the on–going.

Figure 5 shows the preprocessing times (left plot) and the search times (right plot) for versions of **psm** taking into account the distance distributions, i.e., searching for the rarest distance first. The upper curve does not use domain specific knowledge. The plots are drawn over the distance tolerance ϵ from 0 to

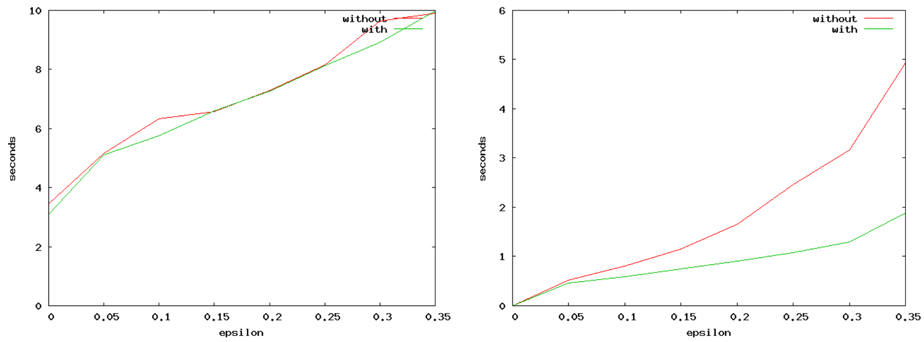


Fig. 3. Preprocessing times (left plot) and the search times (right plot) for the **psm** version without any improvement (upper curves) and the same for the version where the domain specific knowledge has been taken into account.

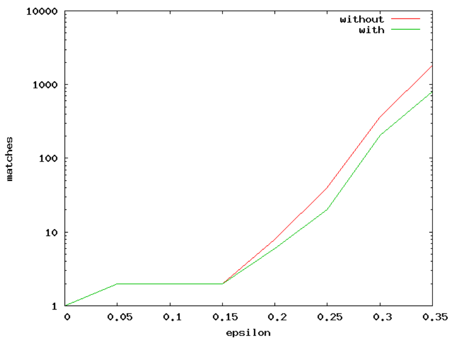


Fig. 4. Number of matches for the two versions without (upper curve) and with (lower curve) domain specific knowledge (note the logarithmic scale).

20%, i.e., the mismatch of the inter–atom distances comparing the search pattern to the encountered location in the search can be that large. As one can observe the preprocessing time is almost the same for both cases, but has increased about 20% compared to the versions without consideration of the distribution. The search time with use of domain specific knowledge again is, at least for larger values of ϵ, almost halved. Moreover, if one compares the search times shown in Figures 3 and 5, searching for the rarest distance first, reduces the search time slightly, especially for small tolerances. Hence, although the preprocessing time has been increased due to the processing of the distribution and the more complex graph construction, the faster search time may lead even to an overall faster algorithm, when partial matches (leaving several atoms out) have to be encountered.

Figure 6 shows the effect of introducing flexible distances according to the amino acid distance of the atoms (as introduced in Section 3.2). The preprocessing times (left plot) are almost the same as in the case of no flexible distance

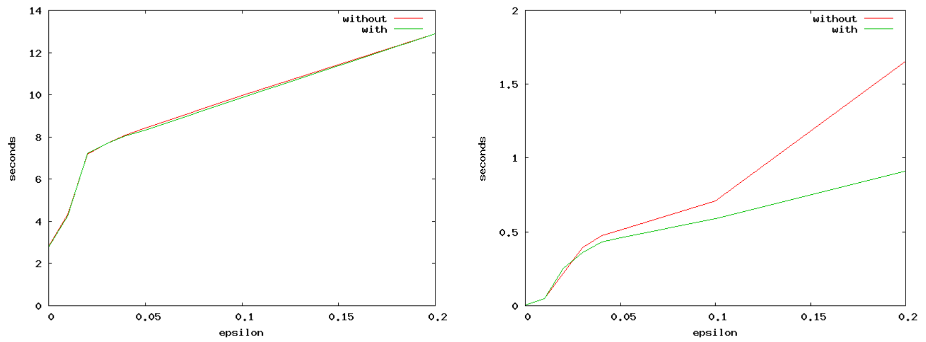


Fig. 5. Preprocessing times (left plot) and the search times (right plot) for versions of **psm** taking into account the distance distributions. The upper curve does not use domain specific knowledge, whereas the lower uses that knowledge.

(compare to left plot in Figure 3). The run times have increased considerably as expected, because we set $k = 1000$, a rather large value, i.e., the distance tolerances for widely separated atoms according to their amino acid distance became quite large. The important fact to notice is that the introduction of domain specific knowledge (lower curves) allows for much faster run times for larger tolerance ϵ.

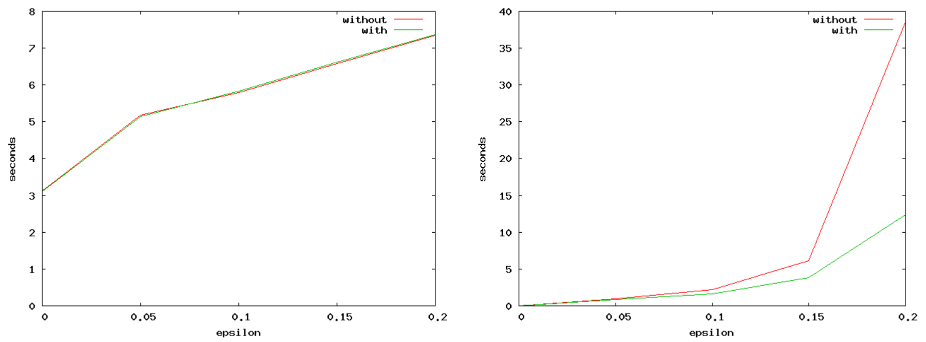


Fig. 6. Preprocessing times (left plot) and the search times (right plot) for versions of **psm** taking into account flexible distances. The upper curve does not use domain specific knowledge, whereas the lower uses that knowledge.

Figure 7 shows what happens if both flexible distances and analysis of the distance distribution is taken into account. Again preprocessing times (left plot) with and without domain specific knowledge is almost equal and more or less the same when not dealing with flexible distances (compare to left plot in Figure 5). The search times (right plot) show the same behavior as seen in Figure 6, i.e., with flexible distances it is not worthwhile to search with the rarest distance first, at least, for large values of the tolerance ϵ.

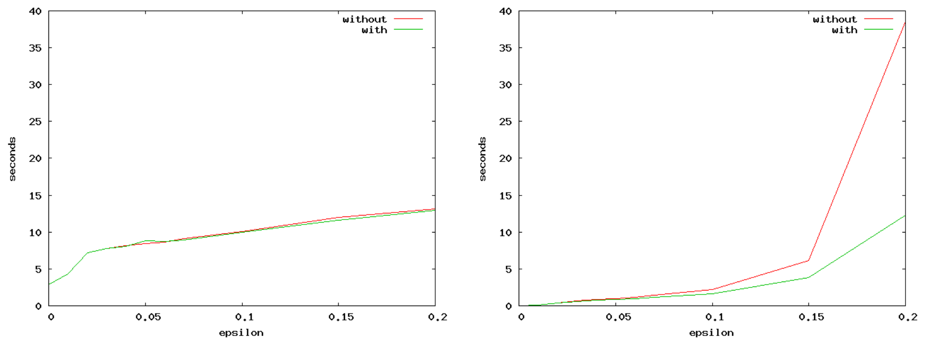


Fig. 7. Preprocessing times (left plot) and the search times (right plot) for versions of **psm** taking into account both flexible distances and analysis of the distance distribution. The upper curve does not use domain specific knowledge, whereas the lower uses that knowledge.

Preprocessing takes somewhat longer but searching is much faster. So it pays off, for instance, when searching for more than one pattern at a time, or when searching for partial matches leaving out one or several atoms.

5 Further work

The ideas and preliminary tests as presented in this article can be extended taking into account the following:

- Up to now we use only domain specific knowledge of the backbone of the protein. Possibly it would be useful to take into account further structures, such as rings, in residues which exhibit less flexible distances as well.
- We might improve the simple amino acid distance dependent tolerance to a more sophisticated one, especially considering the possible rotation of a triple of amino acids. Moreover, the dependence of the overall amino acid count of the protein should be eliminated, i.e., the value of M should be established to a fixed value gathering statistics calculated from PDB structures.
- During the backtracking we might consider the rotational restrictions for certain atom arrangements to early discard candidates.
- The user might introduce individual distance tolerances for certain or all distances within the search pattern.
- Clearly, more rigorous statistical evaluation of the proposed methods on a sufficiently large and diverse set of search spaces and motifs must be elaborated.

- Other type of domain specific knowledge might be introduced, for instance, removal of interior atoms whenever a surface pattern is searched for, matching of only chemically similar atoms, or incorporating the residue type restrictions taking into account properties like electrical charge or hydrophobicity.
- The search algorithm should be parallelized to run at least with a small number of threads taking advantage of modern processor architectures.

6 Conclusion

We presented three approaches and certain details of their implementation together with some preliminary results on how to improve the performance of a general purpose approximate point set match algorithm in the field of structural protein analysis. Introducing domain specific knowledge, such as less flexible chemical bonds, reduces the run time of a search to the half without increase in the preprocessing time. The modified algorithm finds the same locations in terms of clusters as the general algorithm. The run time of a search can be further reduced slightly taking into account the distance distribution in the search graphs. However, the preprocessing time is increased by roughly 20%. Hence, analyzing the distributions pays off whenever the preprocessing time can be amortized, e.g., if partial matches must be found. With the help of flexible distances according to the amino acid distance of the atoms, large tolerances can be employed with moderate increase of the overall search time.

References

1. R.A. Bauer, P.E. Bourne, A. Formella, C. Frömmel, C. Gille, A. Goede, A. Guerler, A. Hoppe, E.W. Knapp, T. Pöschel, B. Wittig, V. Ziegler, and Preissner R. Superimposé: a 3d structural superposition server. *Nucleic Acids Research*, 36 (Web Server Issue):W55–W59, 2008.
2. J.M. Berg, J.L. Tymoczko, and L. Stryer. *Biochemistry.* W.H. Freeman&Co Ltd., 2006.
3. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.
4. A. Formella. Approximate point set match for partial protein structure alignment. In F.M. Couto, J.S. Silva, and P. Fernandes, editors, *Proceedings of Bioinformatics: Knowledge Discovery in Biology (BKDB2005)*, pages 53–57. Facultade Ciencias Lisboa da Universidade de Lisboa, ISBN 972-9348-12-10, June 2005.
5. U. García-Palomares and J. Rodríguez. New sequential and parallel derivative–free algorithms for unconstraint optimization. *SIAM Journal of Optimization*, 13:79–96, 2002.
6. Anthony D. Hill and Peter J. Reilly. Comparing programs for rigid-body multiple structural superposition of proteins. *Proteins: Structure, Function, and Bioinformatics*, 64(1):219–226, 2006.
7. L. Holm, S. Kääriäninen, P. Rosenström, and A. Schenkel. Searching protein structure databases with dalilite v.3. *Bioinformatics*, 24:2780–2781, 2008.

8. A. Hoppe and C. Froemmel. Needle-haystack: A program for the rapid recognition of local structures in large sets of atomic coordinates. *Journal of Applied Crystallography*, 36:1090–1097, 2003.
9. W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallographica A*, 34:827–828, 1978.
10. R. Kolodny, P. Koehl, and M. Levitt. Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *Journal of Molecular Biology*, 346:1173–1188, 2005.
11. R. Maiti, G.H. Van Domselaar, H. Zhang, and D.S. Wishart. Superpose: a simple server for sophisticated structural superposition. *Nucleic Acids Research*, 32 (Web Server Issue):W590–W594, July 2004.
12. M. Novotny, D. Madsen, and G.J. Kleywegt. Evaluation of protein fold comparison servers. *Proteins*, 54:260–270, 2004.
13. A. Stark, S. Sunyaev, and R.B. Russell. A model for statistical significance of local similarities in structure. *Journal of Molecular Biology*, 326:1307–1316, 2003.
14. M. Thimm, A. Goede, S. Hougardy, and R. Preissner. Comparison of 2d similarity and 3d superposition. application to searching a conformational drug database. *Journal of Chemical Information Computing Science*, 44:1816–1822, 2004.
15. M. Unno, T. Mizushima, Y. Morimoto, Y. Tomisugi, K. Tanaka, N. Yasuoka, and T. Tsukihara. The structure of the mammalian 20S proteasome at 2.75 A resolution. *Structure*, 10(5):609–618, 2002.

Chapter 4

COLA - Computational Logic with Applications

Tabling for P-log Probabilistic Query Evaluation

Han The Anh, Carroline Kencana Ramli, and Carlos Viegas Damásio
$\{h.anh|c.kencana\}@fct.unl.pt$, $cd@di.fct.unl.pt$

CENTRIA, Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal.

Abstract. We propose a new approach for implementing P-log using XASP, the interface of XSB with Smodels. By using the tabling mechanism of XSB, our system is most of the times faster than P-log. In addition, our implementation has query features not supported by P-log, as well as new set operations for domain definition.

1 Introduction

Probabilistic reasoning became the major approach for reasoning under uncertainty. From the standpoint of logic and logic programming, the addition of probabilities allows us to represent and reason about finitely varying degrees of belief. From the standpoint of probability and Bayesian Networks, the addition of rule-based representations allows the creation and modification of probabilistic models more easily. The combination of these two lines of research has been attempted in the recent years, resulting in formalisms with both logical and probabilistic knowledge representation capabilities.

One of the approaches is Pearl's probabilistic causal models [8]. Many of the difficulties in probabilistic reasoning on Pearl's formalism lie not only in the use of probabilistic models, but in their formulation relying on propositional reasoning. Other formalisms have proposed the integration between logic and probability such as Pooles Probabilistic Horn Abduction (PHA) [3], the Independent Choice Logic (ICL) [4, 5], the LPAD formalism [9], and the recent language P-log [1].

P-log is a declarative language based on a logic formalism for probabilistic reasoning and action. P-log uses Answer Set Programming (ASP) as its logical and Causal Bayesian Networks (CBN) as its probabilistic foundations. A P-log program consists of a logical part and a probabilistic part. The logical part represents knowledge which determines the possible worlds of the program, including ASP rules and declarations of random attributes, while the probabilistic part contains probability declarations (pr-atoms) which determine the probabilities of those worlds [1, 2]. Although ASP has been proven to be a useful paradigm for solving a variety of combinatorial problems, its non-relevance property [11] makes the P-log system sometimes computationally redundant.

In this paper, we explore a new approach for implementing P-log using XASP, the interface of XSB with Smodels [11] – an answer set solver. With XASP, the relevance of the system is maintained [7]. Moreover, by using the tabling mechanism of XSB our system is most of the times faster than P-log. In addition, our implementation has new kind of queries not supported by P-log, as well as new set operations for domain definition.

The rest of this paper is organized as follows. Section 2 provides a description of the syntax and semantics of the extended P-log system, following closely the formalisms in [1, 2]. Section 3 exhibits some examples. Section 4 outlines the implementation, providing benchmarks comparing P-log(ASP) (original P-log implementation based on [1, 2]) and P-log(XSB) (the new implementation). The paper finishes with conclusions and directions for future work.

2 Extended P-log

In this section, the syntax and semantics of extended P-log programs are defined, being compatible with those ones of the original P-log [1]. The extended syntax has constructs for declaring new sorts by union or intersection of other sorts. This syntactic sugar enables a more declarative representation of many practical problems; for instance, the domain of students who attend both physics and math courses is the intersection of participants in each course, or the domain of cards in the poker game is the union of hearts, spades, diamonds and clubs. In addition, by using XASP, the logical part can use arbitrary XSB prolog code, thus, allowing for the representation of more complex problems that are more difficult or even impossible to express in the original P-log language. The semantics is given by an adapted program transformation into XASP, including the new set operations.

2.1 Syntax

In general, a P-log program Π consists of a sorted signature, declarations, a regular part, a set of random selection rules, a probabilistic information part, and a set of observations and actions.

Sorted signature and Declaration The sorted signature Σ of Π contains a set of constant symbols and term-building function symbols, which are used to form terms in the usual way. Additionally, the signature contains a collection of special function symbols called attributes. Attribute terms are expressions of the form $a(\bar{t})$, where a is an attribute and $\bar{t}$ is a vector of terms of the sorts required by a. A literal is an atomic statement, p, or its explicit negation, neg_p. In addition, p is considered to be the explicit negation of neg_p. The expressions p and $not\ p$ where not is the default negation of ASP are called extended literals.

The declaration part of a P-log program is defined as a collection of sorts and sort declarations of attributes. A sort c can be defined by listing all the elements $c = \{x_1, ..., x_n\}$, by specifying the range of values $c = \{L..U\}$ where L and U are the integer lower bound and upper bound, or even by specifying range of values of members $c = \{h(L..U)\}$ where $h/1$ is a unary predicate. We are also able to define a sort by arbitrarily mixing the previous constructions, e.g. $c = \{x_1, .., x_n, L..U, h(M..N)\}$. In addition, in the extended version, it is allowed to declare union and intersection of sorts: $c = union(c_1, ..., c_n)$ and $c = intersection(c_1, ..., c_n)$, respectively, where $c_i, 1 \leq i \leq n$, are declared sorts.

Declaration of an attribute a with domain $c_1 \times ... \times c_n$ and range c_0 is represented by: $a : c_1 \times ... \times c_n$ --> c_0. If a has no domain parameter, we simply write $a : c_0$. The range of a is denoted by $range(a)$.

Regular part This part of a P-log program consists of a collection of XSB Prolog rules, facts and integrity constraints (IC), formed by literals of Σ. An IC is encoded as a XSB rule with the `false` literal in the head.

Random Selection Rule This is a rule for attribute a having the form:

$$random(RandomName, a(\bar{t}), DynamicRange) \; \texttt{:-} \; Body$$

This means the attribute instance $a(\bar{t})$ is random if the preconditions in $Body$ are satisfied. The $DynamicRange$ allows to restrict the default range for random attributes. The $RandomName$ is a syntactic mechanism used to link random attributes to the corresponding probabilities. If there is no precondition, we simply put $true$ in the body. A constant $full$ can be used in $DynamicRange$ to signal that the dynamic domain equals to $range(a)$.

Probabilistic Information Information about probabilities of random attribute instances $a(\bar{t})$ taking particular value y is given by probability atoms (or pa-atoms for short) which have the following form:

$$pa(RandomName, a(\bar{t}, y), d(A, B)) \; \texttt{:-} \; Body.$$

It means if $Body$ were true, and the value of $a(\bar{t})$ were selected by a rule named $RandomName$, then $Body$ would cause $a(\bar{t}) = y$ with probability $\frac{A}{B}$.

Example 1 (Dice [1]). There are two dice, `d1` and `d2`, belonging to Mike and John, respectively. Each dice has scores from 1 through 6, and will be rolled once. The dice owned by Mike is biased to 6 with probability $1/4$. This scenario can be coded with the following P-log(XSB) program Π_{dice}

```
1. score = {1..6}.   dice = {d1,d2}.
2. owns(d1,mike). owns(d2,john).
3. roll : dice --> score.
4. random(r(D), roll(D), full) :- true.
5. pa(r(D), roll(D,6), d(1,4)) :- owns(D,mike).
```

Two sorts *score* and *roll* of the signature of Π_{dice} are declared in lines 1. The regular part contains two facts in line 2 saying that dice $d1$ belongs to Mike and $d2$ belongs to John. Line 3 is the declaration of attribute *roll* mapping each dice to a score. Line 4 states that the distribution of attribute *roll* is random. Line 5 belongs to probabilistic information part, saying that the dice owned by Mike is biased to 6 with probability $\frac{1}{4}$.

Note that the probability of an atom $a(\bar{t}, y)$ will be directly assigned if the corresponding `pa/3` atom is in the head of some *pa-rule* with a true body. To define probabilities of the remaining atoms we assume that by default, all values of a given attribute which are not assigned a probability are equally likely. For instance, probabilities of rolling Mike's dice to be $i, i \in \{1, 2, 3, 4, 5\}$, are the same and equal to 3/20.

Observations and Actions Observations and actions are, respectively, statements of the forms $obs(l)$ and $do(l)$, where l is a literal. Observations are used to record the outcomes of random events, i.e. random attributes and attributes dependent on them. The dice domain may, for instance, contain `obs(roll(d1,4))` to record the outcome of rolling dice $d1$. The statement $do(a(t, y))$ indicates that $a(t) = y$ is made true as the result of a deliberate (non-random) action. For instance, `do(roll(d1,4))` may indicate that $d1$ was simply put on the table in the described position.

2.2 Semantics

The semantics is defined in two stages. First it defines a mapping of the logical part of Π into its XASP counterpart $\tau(\Pi)$. The answer sets of $\tau(\Pi)$ plays the role of possible worlds of Π. Next the probabilistic part of $\tau(\Pi)$ is used to define a measure over the possible worlds as well as the probability of (complex) formulas. This part of the semantics exactly follows the one in [1].

The logical part of a P-log program Π is transformed into its XASP counterpart $\tau(\Pi)$ by the following five steps:

1. **Sort declaration:**
   - for every sort declaration $c = \{x_1, .., x_n\}$ of Π, $\tau(\Pi)$ contains $c(x_i)$ for each $1 \leq i \leq n$.
   - for every sort declaration $c = \{L..U\}$ of Π, $\tau(\Pi)$ contains $c(i)$ where $L \leq i \leq U$, with integers $L \leq U$.
   - for every sort declaration $c = \{h(L..U)\}$ of Π, $\tau(\Pi)$ contains $c(h(i))$ where $L \leq i \leq U$, with integers $L \leq U$.
   - for every sort declaration $c = union(c_1, ..., c_n)$, $\tau(\Pi)$ contains the rules $c(X) :- c_i(X)$ for each $1 \leq i \leq n$.
   - for every sort declaration $c = intersection(c_1, ..., c_n)$, $\tau(\Pi)$ contains the rule $c(X) :- c_1(X), \ldots, c_n(X)$.
2. **Regular part:** For each attribute term $a(\bar{t})$, $\tau(\Pi)$ contains the rules:
   - $false$:- $a(\bar{t}, Y1),\ a(\bar{t}, Y2),\ Y1 \backslash = Y2.$
     which is to guarantee that in each answer set $a(\bar{t})$ has at most one value.
   - $a(\bar{t}, y)$:- $do(a(\bar{t}, y)).$
     which is to guarantee that the atoms which are made true by a deliberate action are indeed true.
3. **Random selection:**
   - For attribute a, $\tau(\Pi)$ contains the rule: $intervene(a(\bar{t}))$:- $do(a(\bar{t}, Y)).$
   - Each random selection rule
     $random(RanName, a(\bar{t}), DynRange)$:- $Body.$
     is translated into:
     - $a(\bar{t}, Y)$:- $tnot(intervene(a(\bar{t}))),\ tnot(neg\_a(\bar{t}, Y)),\ Body.$
     - $neg\_a(\bar{t}, Y)$:- $tnot(intervene(a(\bar{t}))),\ tnot(a(\bar{t}, Y)),\ Body.$
     - $atLeastOne(RanName, \bar{t})$:- $a(\bar{t}, Y).$
     - $false$:- $tnot(atLeastOne(RanName, \bar{t})).$
     - if $DynRange$ is $full$, $\tau(\Pi)$ contains
       $pd(RanName,\ a(\bar{t}, Y))$:- $tnot(intervene(a(\bar{t}))),\ Body.$
     - if $DynRange$ is not $full$, $\tau(\Pi)$ contains two rules
       $false$:- $a(\bar{t}, Y),\ tnot(DynRange),\ Body,\ tnot(intervene(a(\bar{t}))).$
       $pd(RanName,\ a(\bar{t}, Y))$:- $tnot(intervene(a(\bar{t}))),\ DynRange,\ Body.$
4. **Observation and action:** $\tau(\Pi)$ contains actions and observations of Π.
5. For each literal l, $\tau(\Pi)$ contains the rule: $false$:- $obs(l),\ tnot(l).$

Similarly to ASP, in the body of each XASP rule additional domain predicates are necessary for grounding variables appeared in non-domain predicates (see example 2 for concrete details). In the transformation the XSB default table negation operator *tnot*/1 is used. In the transformation of random selection the auxiliary predicate `pd/2` is used to define default probabilities, recording for each world what are the possible values Y for random attribute term $a(\bar{t})$. The execution of a deliberate action on $a(\bar{t})$ makes the corresponding $intervene(a(\bar{t}))$ true, thereby blocking the generation of random alternatives for attribute $a(\bar{t})$. Also notice that our semantics is equivalent to the semantics defined in [1] for the original P-log syntax. In fact, we reformulated the transformation from the original paper to adapt it to the XASP syntax. For example, the cardinality expression of Smodels language used in the original paper is replaced with an even loop to generate stable models and rules for determining upper and lower bounds. The rationale for the transformation can be found in [1].

Notice that the probabilistic information part of a P-log program, consisting of pa-rules, is kept unchanged through the transformation

Example 2. For better understanding of the transformation, we provide here the resulting transformed program $\tau(\Pi_{dice})$ of Π_{dice} described in example 1:

```
1. score(1). score(2). score(3). score(4). score(5). score(6).
2. dice(d1). dice(d2). owns(d1,mike). owns(d2,john).
3. false:-score(X),score(Y),dice(D),roll(D,X),roll(D,Y),X \= Y.
4. roll(D,X) :- dice(D), score(X), do(roll(D,X)).
5. intervene(roll(D)):-dice(D),score(X),do(roll(D,X)).
6. roll(D,X) :- dice(D), score(X),
       tnot(intervene(roll(D))),tnot(neg_roll(D,X)).
7. neg_roll(D,X) :- dice(D), score(X),
       tnot(intervene(roll(D))),tnot(roll(D,X)).
8. atLeastOne(r(D),D) :- dice(D), score(X), roll(D,X).
9. false :- dice(D), tnot(atLeastOne(r(D),D)).
10.pd(r(D),roll(D,X)) :- dice(D), score(X),
       tnot(intervene(roll(D,X))).
11.pa(r(D),roll(D,6),d(1,4)) :- owns(D,mike).
```

Lines 1-2 are the transformation of sorts declaration. Lines 3-4 are the resulting code of the transformation for attribute `roll` (regular part). Lines 5-10 are the result of the transformation for the random selection part in line 5 of Π_{dice}. Line 11 is the probabilistic information part, being kept unchanged from the original program (line 6 of Π_{dice}). Notice that the domain predicates $score/1$ and $dice/1$ were added in some rules (lines 3-10) for variables grounding purpose.

3 Examples of Application

In this section we describe several examples that have been tested in P-log(ASP) and P-log(XSB). Several problems are easily solved using probabilistic knowledge, but here we focus on how can inferences be drawn logically.

3.1 Cards Problem

Suppose there is a deck of cards divided into spades, hearts, diamonds and clubs. Each suit contains a numbers and pictures. Numbers are from 1 to 10 and pictures are jack, queen and king. What is the probability of having single pair at hand [16]?

```
1. heart   = {h(1..10), h(jack), h(queen), h(king)}.
   spade   = {s(1..10), s(jack), s(queen), s(king)}.
   diamond = {d(1..10), d(jack), d(queen), d(king)}.
   club    = {c(1..10), c(jack), c(queen), c(king)}.
2. cards   = union(heart,spade,diamond,club).
3. number  = {1..5}.
4. draw : number --> cards.
5. random(r(N),draw(N),full):- true.
6. false :- same.
7. same  :- draw(N1,X), draw(N2,X), N1 \= N2.
8. pair(N1,N2):-draw(N1,X1),draw(N2,X2),N1\=N2,sameValue(X1,X2).
9. sameValue(X,Y) :- X =.. [_|V], Y =.. [_|V].
10.singlePair :-
     findall(pair(X,Y), pair(X,Y),[pair(C1,C2),pair(C2,C1)]).
```

Fig. 1: Cards Example

The corresponding P-log(XSB) program is shown in Figure 1. Lines 1-4 are declaration of attributes. Line 5 is a random selection rule, saying that the distribution of each attribute $draw_i$ is random. In this example, the same card cannot be drawn twice. This condition is modelled by an ICt in line 6. Lines 8-10 capture the existence of a single pair, namely, predicate $pair(N1, N2)$ in line 8 says that the two different drawings N1-th and N2-th provide a pair, i.e. give the same value (defined by predicate $sameValue/2$ in line 9), and line 10 says that there is only one pair, i.e. there are exactly two different but commutative pairs of drawings that hold (implemented using XSB built-in predicate $findall/3$).

Notice that in this example we use some built-in predicates of XSB that are not supported by ASP such as (=..)/2 and $findall/3$. This feature enables our system to be able to model more complicated problems. To model, for example, the rule for $sameValue/2$ in line 9, in P-log(ASP) we must use a number of rules for grounding that rule. Probability of drawing a single pair is obtained by the query `?- pr(singlePair,Vp).` The system is supposed to provide the answer [16]: Vp = 0.422569. However, our system only manages to give the answer for the problem with smaller size. Further discussion can be found in subsection 4.2.

3.2 Wetgrass Problem

Consider the Wetgrass example [10] in which all nodes are boolean random variables, where the two possible values are denoted by `t`(true) and `f`(false). The corresponding Bayesian Network can be found in Figure 2.

The corresponding P-log(XSB) program is shown in Figure 3. As we see in Figure 2, the event *grass is wet* (W=true) has two possible causes: either the water sprinkler is on (S=true) or it is raining (R=true). The domain declarations are described in lines 1-2. The attributes *sprinkler, rain, cloudy* and *wetgrass* are distributed randomly (line 3). The probabilistic relationships are captured by the Conditional Probability Tables

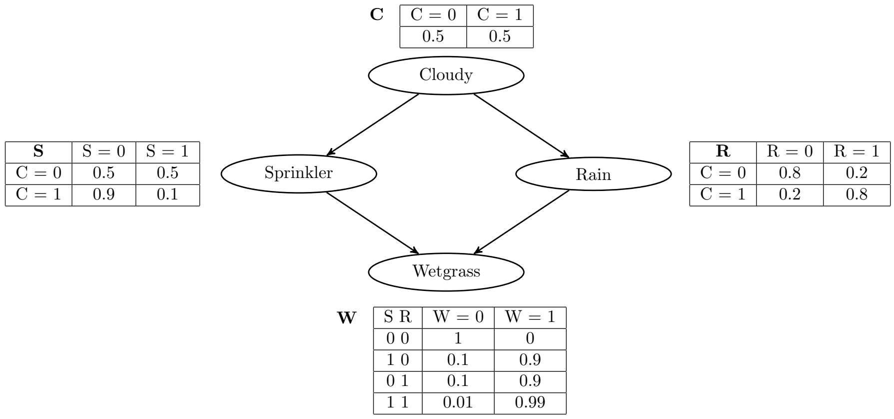


Fig. 2: Bayesian Network for Wetgrass Example

(CPTs) in the diagram. The probabilistic information pa-clauses in line 4 represent the conditional probabilities according to the CPTs table.

Suppose we observe that grass is wet. We want to know which one is the more likely cause of wetgrass, raining or sprinkler. The probability of raining being true given wetgrass is true can be easily obtained mathematically $Pr(R = true|W = true) = 0.7079$. Similarly, the probability of sprinkler being on given wetgrass is true: $Pr(S = true|W = true) = 0.4298$. These exact values can be found with the queries `?- pr(rain(t)'|'obs(wetgrass(t)),PR)` and `?- pr(sprinkler(t)'|'obs(wetgrass(t)),PS)` [1], respectively.

Probabilistic Reasoning with P-log(XSB) We show how P-log(XSB) can be used as a meta-level language for probabilistic reasoning. The example is extended with the scenario represented in lines 5-9. Imagine that some students have finished their lectures and they are planning to refresh themselves. They have to make decision on one of two choices: going to the beach or the cinema (line 5). If it is not raining, they choose to go to the beach (lines 6–7). Otherwise going to the cinema is the only option (line 8). The probability of raining to be true is higher if it was raining last night (line 9). Based on the observation, this morning they saw the grass was wet. So, what will they choose for their refreshing moment?

This can be done with the query `?-refreshing(X).` The system provides the answer: `X = goingToCinema` since it was raining (line 8), which is based on the fact that the probability of raining given that grass is wet is higher than the one of sprinkler being on under the same condition (line 9).

3.3 Random blocks problem [15, 14]

A problem in the random blocks domain consists of a collection of locations, knowledge of which locations left of and below each other, a set of blocks, and knowledge

[1] The probability of A given B is computed using the query `?- pr(A '|' B, V)`.

```
1. bool={t,f}.
2. cloudy:bool.  rain:bool.  sprinkler:bool. wetgrass:bool.
3. random(rc, cloudy, full).     random(rr, rain, full).
   random(rs, sprinkler, full).  random(rw, wetgrass, full).
4. pa(rc,cloudy(t),d(1,2)).  pa(rc,cloudy(f),d(1,2)).
   pa(rs,sprinkler(t),d(1,2)) :- cloudy(f).
   pa(rs,sprinkler(t),d(1,10)) :- cloudy(t).
   pa(rr,rain(t),d(2,10)) :- cloudy(f).
   pa(rr,rain(t),d(8,10)) :- cloudy(t).
   pa(rw,wetgrass(t),d(0,1)) :- sprinkler(f),rain(f).
   pa(rw,wetgrass(t),d(9,10)) :- sprinkler(t),rain(f).
   pa(rw,wetgrass(t),d(9,10)) :- sprinkler(f),rain(t).
   pa(rw,wetgrass(t),d(99,100)) :- sprinkler(t),rain(t).
5. refreshing(goingToBeach) :- goingToBeach.
   refreshing(goingToCinema):- goingToCinema.
6. goingToBeach :- sunny.
7. sunny :- not raining.
8. goingToCinema :- raining.
9. raining :- pr(rain(t) ’|’ obs(wetgrass(t)),PR),
              pr(sprinkler(t)’|’obs(wetgrass(t)),PS), PR > PS.
```

Fig. 3: Wetgrass example

of a location for each block. This problem was introduced in [15] to demonstrate the performance of ACE , a system for probabilistic inference based on relational BNs. In [14] it was used as a benchmark to compare latest P-log [12] to ACE (P-log is faster).

We use that problem to compare our system to the latest Plog. The representation of the problem in Plog(XSB) syntax is straightforwardly adapted from the one in P-log(ASP) [14]. Due to lack of space we will not show it here.

4 Implementation of the P-log(XSB) system

Our system consists of two main modules: transformation and probabilistic information processing. The first module transforms an original P-log(XSB) program into an appropriate form for further computation by the second module. Both modules were developed on top of XSB Prolog [20], an extensively used and state-of-the-art logic programming inference engine implementation, supporting the Well-Founded Semantics (WFS) for normal logic programs.

The tabling mechanism [19] used by XSB not only provides significant decrease in time complexity of logic program evaluation, but also allows for extending WFS to other non-monotonic semantics. An example of this is the XASP interface (standing for XSB Answer Set Programming) which extends WFS with Smodels to compute stable models [6]. In XASP, only the relevant part to the query of the program is sent to Smodels for evaluation [11]. This allows us to maintain the relevance property for queries over programs, something that ASP does not comply to [7]. ASP obtains all the complete stable models for the whole program, which might contain redundant information for the evaluation of a particular query. Our approach of using XASP interface sidesteps this issue, sending to Smodels only part of the program that is relevant to the query.

The transformation module transforms the original P-log (XSB) program into a XASP program using five transformation steps described in section 2. This program is then used as the input of the Probabilistic Processing module which will compute all stable models with necessary information for dealing with the query. Only predicates for random attributes and probabilistic information, which have been coded by predicates `pd/2` as the default probability and `pa/3` as the assigned probability are kept in each stable model. In the current version of XASP, those literals are collected by a (posterior) filter after all stable models were generated. This is improved in our system by (ourselves) equipping XASP with a prior filter which enables it to generate stable models with literals by need.

Having obtained stable models with necessary information, the system is ready to answer queries about probabilistic information coded inside the program. Besides queries in form of ASP formulas, our system was extended to be able to answer queries in the form of Prolog predicates which can be defined in a variety of ways. The code for defining the predicate can be included in the original P-log(XSB) program, in a separated XSB prolog program or even asserted into the system. This feature enables us to give very complicated queries which is difficult or even impossible for P-log(ASP) to tackle, e.g. *singlerPair* in the Cards problem. The implementation of this new feature can be done easily with XASP, using the query as a filter for ruling out unsatisfied stable models. In addition, arguing that the system's users usually need to query the program with a number of goals for different kinds of probabilistic information, the system is optimized for answering several queries. In this way the meta-queries illustrated in the previous Wetgrass example can be executed faster, enabling the construction of more sophisticated knowledge bases making use of probabilistic reasoning. Technically, this has been done by memorizing predicates used for processing queries. That means for most of predicates necessary for processing probabilistic queries, we only have to compute them once and reuse the results later without further computation. Moreover, since in the implementation of conditional probability, probabilities of two queries must be computed [1], it benefits even more from the tabling mechanism. The good effects of this choice are clear and discussed below.

4.1 Analysis of implementation

Based on the formalisms in section 2 we have successfully implemented P-log in XSB using XASP. In addition, we have compared some computation results of our system to the P-log(ASP) current version [12].

4.2 Evaluation

In this section we describe some benchmark problems used to compare the performance of our implementation in XASP with the one of P-log(ASP). We use the same examples described above, just with greater size. For clarity, we denote by $Dice\langle D, X\rangle$ the Dice example with D dice and each dice having X scores; $Cards\langle D, X\rangle$ – Card example with D drawn cards and X cards on the deck; $Block\langle D, X\rangle$ – Random block example with D blocks and X locations. Since the probability of a formula w.r.t. a program depends on stable models in which it is true and those models can be obtained by combining the ones of its atomic elements, we can use just the set of atomic formulas, for example $roll(d1, i)$ and $roll(d2, i)$ ($i = 1, \ldots, 20$) in $Dice\langle 2, 20\rangle$, for testing the performance of the systems, reflecting the performance of any formulas.

Table 1 shows the number of stable models, time for the first run, average time from second to tenth runs as well as average time of ten runs of atomic queries (queries for atomic formulas) w.r.t. P-log(XSB) and P-log(ASP) on a computer running Linux Ubuntu 8.10, 1.8 Ghz core 2 dual, and 2 GByte of RAM. The P-log(XSB) system was run on XSB 3.2. Both systems use Smodels 2.33, Lparse-1.1.1. Notice that since we use a set of atomic queries for comparison, only relevant part of the examples which does not contain P-log(ASP) unsupported constructs is necessary for the P-log(ASP) site.

We have run with a number of sets of ten randomly selected atomic queries in both systems and realized that the relative performance of the systems did not depend on the selected sets. This is because of the way the probability of formulas is computed: in any case, the unnormalized probability of every stable model of the program must be computed, and then reused for each stable model in the list of the ones that satisfy the formula. Also note that in our system it is possible to run all ten queries at once, making a conjunctive query with those ten queries in the body, and the performance is identical to the one obtained by running the queries separately. However, this is not the case with P-log(ASP) where the queries have to be run separately. We consider only the total time of computing stable models and deriving the answers. The time of transformation is considerably small on both systems, thus being ignored.

Table 1: Benchmark

		P-Log(XSB)				P-Log(ASP)			
Examples	Number of Stable Models	First Run	Second Run	Average (10 times)	Average (2nd - 10th)	First Run	Second Run	Average (10 times)	Average (2nd - 10th)
Dice⟨2, 20⟩	400	0.1160	0.0120	0.0216	0.0111	0.07	0.03	0.0456	0.0411
Dice⟨2, 50⟩	2,500	3.7880	0.1000	0.4872	0.1204	0.71	0.72	0.7189	0.72
Dice⟨2, 100⟩	10,000	88.2490	0.6600	9.4181	0.6591	7.41	7.41	Down in 6th run	
Cards⟨5, 8⟩	6,720	1.5520	0.3360	0.3764	0.2458	0.63	0.66	0.6367	0.6378
Cards⟨5, 10⟩	30,240	8.1320	1.7000	2.1221	1.4543	3.68	3.69	Down in 7th run	
Cards⟨5, 11⟩	55,440	15.9440	3.2970	4.7246	3.4780	7.64	7.52	Down in 4th run	
Cards⟨5, 12⟩	95,040	29.3250	6.0010	8.6161	6.3151	14.49	Down	-	-
Block⟨3, 22⟩	9,240	6.3440	0.5000	0.9724	0.3756	1.72	1.7	1.7033	1.7
Block⟨3, 30⟩	24,360	24.5290	1.6280	3.4518	1.1099	6.39	6.37	Down in 5th run	
Block⟨3, 35⟩	39,270	48.0030	2.9200	6.5396	1.9326	12.37	17.88	Down in 3th run	
Block⟨3, 40⟩	59,280	89.2610	4.9880	12.5167	3.9896	25.92	Down	-	-

According to the results, for the first run, our system is about 1.5 to 2 times slower than P-log(ASP), except for $Dice\langle 2, 100\rangle$ where ours is 11 times slower. This is due to the fact that the cardinality constraints of Smodels language have not been available for the current version of XASP (namely, $xnmr\_int$ interface). As discussed in Section 2 we need additional rules for determining upper and lower bounds of the constraints, resulting in that the larger cardinality interval is, the worse the performance of our system is. However, from the second time on, ours is faster, namely, from 3 to 9 times if the first one not being taken into account and from 2 to 5 times otherwise. Therefore, the more probabilistic information we need to extract from the knowledge base, the more useful our system is. Furthermore, P-log(XSB) is more stable (see the cases where P-log(ASP) crashes (denoted by $Down$)).

5 Conclusions and Future Work

We have described our approach for re-implementing P-log in XSB using XASP. With XASP, the relevance of the system is maintained, thus Smodels only needs to work with the relevant part and derives only necessary information for further processing. In addition, the tabling mechanism of XSB significantly decreases the computation time of the system by reusing the computations having been done. By comparing the two systems using some benchmarks, we have shown that although our system is slower for the first query, it is much faster for the subsequent queries.

We have extended P-log with new features, first of all, to query the system with more expressive queries not supported in original version. This feature enables users with a very powerful way to gain necessary information from the knowledge base. Furthermore, in many practical problems the domain of one attribute needs to be represented by, e.g. union, intersection, of other domains. Hence, some set operations for domain definition equipped in our system make the users easier in representing those problems.

In our implementation, we have made some important changes in XASP package, namely, $xnmr\_int$ interface, which result in a better performance of the system. We also plan to improve $xnmr\_int$ in order to accept the full lparse syntax, particularly the cardinality constraints. We expect that in this way our system will obtain a comparable performance for the first run.

In general, the approach to probabilistic reasoning by deriving all possible worlds has to deal with a very big list of stable models with a number of predicates. In any cases, we have to compute the unnormalized probability for each stable model of the list. Since the computation can be done in parallel, the performance of the system would very much benefit from multicore CPU computers by using multi-threading, which is very efficient in XSB, from version 3.0 [20]. This approach will be explored in our next version. We also envisage to explore properties of the transformed programs in order to control the exponential blow-up of stable models, w.r.t. some specific cases, e.g. programs resulting from the transformation of poly-tree Bayes Networks.

Last but not least, it is worth mentioning that our system has been successfully integrated in several XSB Prolog systems such as ACORDA [17], Evolution Prospection Agents system [18], for modelling uncertainty in decision making. Those systems employ several kinds of preferences which require probabilistic information. We may prefer one abducible to the other if the first one has greater probability, or one outcome to the other if the probability of the first one to occur is greater than the other one.

References

1. C. Baral, M. Gelfond, and N. Rushton. *Probabilistic reasoning with answer sets.* In Proceedings of LPNMR7, pages 21−33, 2004.
2. C. Baral, M. Gelfond and N. Rushton. *Probabilistic Reasoning with Answer Sets.* TPLP, Volume 9, Part 1, pages 57-144, January 2009.
3. D. Poole. *Probabilistic horn abduction and bayesian networks.* Artificial Intelligence, 64(1):81-129,1993.
4. D. Poole. *The independent choice logic for modelling multiple agents under uncertainty.* Artificial Intelligence, 94(1-2):5−56, 1997.
5. D. Poole. *Abducing through negation as failure: Stable models within the independent choice logic.* Journal of Logic Programming, 44:535, 2000.
6. I. Niemelã,P. Simons. *Smodels: An implementation of the stable model and well-founded semantics for normal logic programs.* Procs. of LPNMR, Berlin, 1997.
7. J. Dix. *A classification theory of semantics of normal logic programs: i. strong properties, ii. weak properties.* Fundamenta Informaticae, 22(3):227255,257288, 1995.
8. J. Pearl. *Causality.* Cambridge U. P. 2000.
9. J. Vennekens, S. Verbaeten, and M. Bruynooghe. *Logic programs with annotated disjunctions.* In ICLP, pages 431−445, 2004.
10. K. Murphy. *A Brief Introduction to Graphical Models and Bayesian Networks.* http://www.cs.ubc.ca/∼murphyk/Bayes/bayes.html, [Accessed 25 January 2008].
11. L. Castro, T. Swift, and D. S. Warren. *XASP: Answer Set Programming with XSB and Smodels.* http://xsb.sourceforge.net/packages/xasp.pdf
12. Latest P-log implementation (September 2008) downloadable at http://www.cs.ttu.edu/∼wezhu/
13. M. Gelfond, V. Lifschitz. *The stable model semantics for logic programming.* Procs of the 5th Intl. Conf. and Symp. on Logic Programming. 1070–1080, 1988.
14. M. Gelfold, N. Rushton and W. Zhu. *Combining Logical and Probabilistic Reasoning.* AAAI Spring Symposium, 2006.
15. M. Chavira, A. Darwiche, M. Jaeger, *Compiling relational Bayesian networks for exact inference.* Procs. European Workshop on Probabilistic Graphical Models, Netherlands, 2004.
16. M. Shackleford, A.S.A. *Problem 62 Solution - Probabilities in Poker.* http://mathproblems.info/prob62s.htm, [Last accessed 10 June 2008].
17. L. M. Pereira, C. K. Ramli, *Modelling Probabilistic Causation in Decision Making*, Procs. First KES Intl. Symposium on Intelligent Decision Technologies, Springer, Japan, 2009.
18. L. M. Pereira, H. T. Anh. *Evolution Prospection*, Procs. First KES Intl. Symposium on Intelligent Decision Technologies, Springer, Japan, 2009.
19. T. Swift. *Tabling for non-monotonic programming.* Annals of Mathematics and Artificial Intelligence, 25(3–4):201-240, 1999.
20. *The XSB System Version 3.0 Volume 2: Libraries, Interfaces and Packages.* July, 2006.

Learning and Reasoning about Uncertainty in the Semantic Web

Pedro Oliveira and Paulo Gomes

CISUC-DEI-FCTUC
University of Coimbra
Pólo II, 3030 Coimbra, Portugal
pcoliv@student.dei.uc.pt, pgomes@dei.uc.pt

Abstract. The main idea behind the Semantic Web is the representation of knowledge in an explicit and formal way. This is done using ontology representation languages as OWL, which is based on Description Logics and other logic formalisms. One of the main objectives with this kind of knowledge representation is that it can then be used for reasoning. But the way reasoning is done in the Semantic Web technology is very strict, defining only a right and wrong view of the world. The real world is uncertain and humans have learned how to deal with this crucial aspect. In this paper, we present an approach to reasoning with uncertainty information in the Semantic Web. We have applied Markov Logic, which is able to reason with uncertainty information, to several Semantic Web ontologies, showing that it can be used in several applications. We also describe the main challenges for reasoning with uncertainty in the Semantic Web.

Keywords: Semantic Web, Probabilistic Reasoning, Markov Logic.

1 Introduction

The idea of the Semantic Web [1] envisions a world where agents share and transfer structured knowledge in an open and semi-automatic way. In most of the cases, this knowledge is characterized by uncertainty. However, Semantic Web languages like OWL[1] do not provide any means of dealing with this uncertainty. They are mainly based on crisp logic, unable of dealing with partial and incomplete knowledge. Reasoning in the Semantic Web resigns to a deterministic process of verifying if statements are true or false.

In the last years, some efforts have been made in representing and reasoning with uncertainty in the Semantic Web (see [2] for a complete overview about the subject). These works are mainly focused on how to extend the logics behind Semantic Web languages to the probabilistic/possibilistic/fuzzy logics, or on how to combine these languages with probabilistic formalisms like Bayesian Networks. In all of these

[1] http://www.w3.org/TR/owl-features/

approaches, this is achieved by annotating the ontologies with some kind of uncertainty information about its axioms, using this information to perform uncertainty reasoning. Nevertheless, several questions arise: how are these uncertainties asserted? How can reasoning be done with this uncertainty information?

One promising approach to reasoning with uncertainty is Markov Logic [3]. In this type of logic there is no right and wrong world, there are multiple worlds with different degrees of probability. Markov Logic is based in first-order logic and probabilistic graphical models to deliver the probability of a given logic formula. This type of logic has been applied to several application domains [3] and has show to be robust and able to deal with uncertain knowledge.

In our work, we are studying how we can reason about uncertainty in OWL ontologies without any kind of uncertainty associated. In this paper, we describe an approach that uses Markov Logic to accomplish this task. First, the ontology is interpreted as first-order logic, and ontology individuals are used to learn the uncertainty of the resulting formulas. Next, we use Markov Logic inference capabilities to perform approximate probabilistic reasoning in the resulting model. We present several experiments of this approach with different OWL ontologies.

All the capabilities described in this paper are implemented in *Incerto*[2], an open source probabilistic reasoner for the Semantic Web.

The next sections introduce the concepts of Semantic Web and Markov Logic. Section 4 describes our approach to the transformation of OWL into Markov Logic. Section 5 presents the experimental work done and its main results. We finalize this paper by describing future work and conclusions of our work.

2 Semantic Web

In the current web, while it is easy to a human infer the meaning of objects in a web page, to a machine this task is not so easy, being only possible to interpret the keywords and links of those objects. The Semantic Web [1] tries to fill this knowledge gap between human and machines by adding background knowledge to the web, allowing machines to infer the real meaning of objects. This background knowledge is usually expressed by ontologies [4], i.e., sets of knowledge terms for some particular topic, including the vocabulary, semantic interconnections, and rules of logic/inference of those terms.

The most prominent markup language proposed by the *W3C* to model ontologies in the Semantic Web is the *Web Ontology Language*[3] (OWL). OWL provides an expressive shared vocabulary to represent knowledge in the Semantic Web. This vocabulary allows expressing axioms about classes, properties, and individuals of the domain. In this paper, we will focus on OWL2[4] [5], the new version of OWL proposed by the *W3C*, which subsumes the decidable subsets of the original OWL

---

[2] http://code.google.com/p/incerto/
[3] http://www.w3.org/2004/OWL/
[4] http://www.w3.org/TR/owl2-quick-reference/

(OWL-DL and OWL-Lite).

OWL2 is based on the Description logic $\mathcal{SROIQ}(\mathcal{D})$ [5]. Description logics [6] are a family of logical languages specially designed to model terminological domains. Formulas in Description Logics are composed by two symbols: *concepts* (i.e., sets of individuals) and *roles* (i.e., relationships between individuals). A relevant feature of Description Logics is their separation of knowledge bases in two distinct parts: the intensional knowledge in the form of a terminology, called *Terminological Box* (TBox), and the extensional knowledge, called *Assertional Box* (ABox). The TBox provides the vocabulary, in terms of concepts and rules, of the knowledge base. This is usually done by defining concepts using the logical equivalence constructor (e.g., $Woman \equiv Person \sqcap Female$). The Abox uses the TBox vocabulary to make assertions about individuals (e.g. $Woman(ANNA)$).

3 Markov Logic

Markov Logic [3] combines first-order logic and probabilistic graphical models (Markov networks [7]) in the same representation. The main idea behind Markov Logic is that, unlike first-order logic, a world that violates a formula is not invalid, but only less probable. This is done by attaching weights to first-order logic formulas: the higher the weight, the bigger is the difference between a world that satisfies the formula and one that does not, other things been equal. These sets of weighted formulas are called Markov Logic networks (MLNs). Given a set of constants (i.e., individuals) of the domain and an interpretation, the groundings of the formulas in an MLN can generate a Markov network by adding a variable for each ground atom, an edge if two ground atoms appear in the same formula, and a feature for each grounded formula. The probability distribution of the network is defined as

$$\mathrm{P}(X = x) = \frac{1}{Z} exp\left(\sum_{i=1}^{F} w_i n_i(x)\right), \tag{1}$$

where F is the number of formulas in the MLN, $n_i(x)$ is the (binary) number of true groundings of F_i in the world x, w_i is the weight of F_i, and Z is a normalizing constant.

There are two relevant tasks of Markov Logic for this work: weight learning and inference.

3.1 Weight Learning

Given an MLN without weights and a set of example data composed by individuals of the domain, weights can be learned generatively by maximizing the *pseudo-log-likelihood* [8] of that data. Basically, it is an iterative process where if the model predicts that a formula is true less often than it really is in the data, the weight is increased; otherwise, it is decreased. The pseudo-log-likelihood of world x given weight w is defined as

$$\log P_w^*(X = x) = \sum_i \log P_w(X_i = xi \mid N_x(X_i)), \quad (2)$$

where xi is the truth value of variable i, and $N_x(X_i)$ is the truth values of the neighbors of i.

3.2 Inference

The most interesting inference task in Markov Logic is to find the marginal and conditional probabilities of a formula given an MLN and possibly other formulas as evidence. Since exact inference can be too difficult in large domains, approximate inference algorithms, like those based on randomized sampling (e.g., *Markov Chain Monte Carlo* [7] (MCMC)), are usually used. However, MCMC is not efficient in domains where formulas with deterministic or near-deterministic dependencies exist (e.g., formulas with infinite weight) because these areas of the search space can be very difficult to traverse by simple flipping the value of the non-evidence variables. To solve this problem, we can use *MC-SAT* [9], a combination of MCMC and the *SampleSAT* satisfiability solver [10]. MC-SAT uses *slice sampling* to help capturing the dependencies between variables, allowing jumping from these difficult areas.

4 Markov Logic for the Semantic Web

As we previously seen, MLNs are formed by a set of weighted first-order logic formulas. If we want to use Markov Logic in the Semantic Web, we have to determine where these formulas and weights come from.

4.1 Formulas

The Semantic Web language used in this work (OWL2) is based on the Description Logic $\mathcal{SROIQ}(\mathcal{D})$. One characteristic of Description Logic languages is that they follow a *model-theoretic* semantics [6], and therefore can (in most of the cases) be interpreted as formulas in first-order logic. The main idea behind this interpretation is that concepts correspond to unary predicates, roles to binary predicates, and individuals correspond to constants. In our case, $\mathcal{SROIQ}(\mathcal{D})$ can be easily interpreted as first order formulas. Some examples of these translations are provided (Table 1).

Table 1. OWL2 examples of interpretation as first-order logic formulas. A complete description of the interpretation can be found on the *Incerto* website[5].

OWL2 Expression	First-order logic formula
$SubClassOf(CE_1, CE_2)$	$\forall x : CE_1(x) \Rightarrow CE_2(x)$
$TransitiveProperty(OPE)$	$\forall x, y, z : OPE(x,y) \wedge OPE(y,z) \Rightarrow OPE(x,z)$
$ClassAssertion(CE, a)$	$CE(a)$

4.2 Weights

The most obvious way to acquire the uncertainty from an ontology is to delegate this task to the ontology creators. This is the approach used by other works [2]. However, creating and maintaining large uncertainty-annotated ontologies can be a cumbersome and difficult task, invalidating all the gains that could arise from the annotation. This fact raises the need for developing mechanisms to learn this uncertainty automatically. This can be useful not only to help users when creating uncertain ontologies, but also to gain access to the vast number of non-annotated ontologies already available.

In this paper, we explore the use of the weight learning capabilities of Markov Logic to learn uncertainty information. As previously seen, in Markov Logic, formulas' weights can be learned generatively through example data. This example data comprises individuals of the domain and their relations. In the case of OWL2, this corresponds to the ABox of the ontology. Therefore, the ABox can be interpreted as ground atoms, and weights can be learned with that information.

5 Experimental Analysis

In this section, we present our experiences on using Markov Logic to learn and reason about uncertainty in OWL2 ontologies. The main objective of these experiments is to show the feasibility of our approach in real-world domains. All the experiences were made with *Incerto*, using Alchemy[6] [11] as the Markov Logic engine. All the ontologies and results of the experiences can be also found on the *Incerto* website[7].

5.1 The Financial Experiment

Evaluation Procedure and Data Set. Uncertainty reasoning is very important in discovering hidden knowledge in risk assessment domains. In this experiment, we will use a financial ontology, GoldDLP[8], to assess the risk of certain financial operations. In this ontology, there is information about a bank that offers services like

[5] http://code.google.com/p/incerto/wiki/OWL2FOL
[6] http://alchemy.cs.washington.edu/
[7] http://code.google.com/p/incerto/wiki/EPIA2009Experimentation
[8] http://www.cs.put.poznan.pl/alawrynowicz/semintec.htm

loans and credit cards to private persons. The ontology contains 116 class/property axioms and 297 individuals, mainly distributed between accounts, clients, credit cards, and loans. One of the most interesting tasks in this domain is to determine if a given loan is a problematic loan. There is an OWL class responsible for that information, named *ProblemLoan*, and some axioms about that class (e.g., *ProblemLoan* is the complement of *OkLoan*). The main task in this experiment is to determine each loan's probability of being a *ProblemLoan*.

Experimental Results. Using generative learning and MC-SAT, we found that nine loans have a probability >90% of satisfying the conditions necessary for being a *ProblemLoan*. If we compare the results with a non-probabilistic reasoner, like Pellet[9] [12], these are the same nine individuals identified deterministically by it. However, our approach returns some more interesting results that were not identified by Pellet. All the other loans have a probability between 35-39% of satisfying the conditions of *ProblemLoan*. This information is valuable because, roughly speaking, it demonstrates that any loan has an associated probability of being a problematic loan. This kind of results cannot be achieved using non-probabilistic reasoning, and therefore demonstrates the necessity of probabilistic reasoning to have a more profound understanding about the domain. However, if we use an existent Semantic Web probabilistic reasoner (e.g., Pronto[10] [13]), its results are the same of a non-probabilistic one, since the ontology does not contain any information about the uncertainty of its axioms.

5.2 The Social Network Experiment

One of the most used Semantic Web vocabularies is the *Friend of a Friend*[11] (FOAF) vocabulary. This vocabulary allows describing social network data (i.e., persons and their relations) in OWL, with special incentive in linking users from different social networks. There are several web-based social networks that provide information about their users in FOAF (see Mindswap[12] for a comprehensive list), and some projects are already exploiting that information (e.g., Google Social Graph API[13]).

The objective of this experiment is to use Markov Logic to explore the relational structure of FOAF networks. As data set, we choose Advogato[14], a social network of free software developers. Advogato provides three interesting FOAF properties to our analysis: *foaf:knows(x,y)*, meaning that user *x* knows user *y*; *foaf:currentProject(x,y)*, meaning that user *x* is currently working in project *y*; and *foaf:member(x,y)*, meaning that user *x* is member of the group *y*. After gathering and processing all the available FOAF profiles, we had a total of 6688 individuals, representing 4198 users, 2487 projects, and 3 groups. Based on the Link Mining literature [14][15], we identified

[9] http://pellet.owldl.com/
[10] http://pellet.owldl.com/pronto
[11] http://www.foaf-project.org/
[12] http://trust.mindswap.org
[13] http://code.google.com/apis/socialgraph/
[14] http://advogato.org/

three interesting tasks to our experiment: link prediction, link-based classification, and link-based cluster analysis.

5.2.1 Link Prediction

Link prediction [14] is the problem of predicting the existence of a link between two objects based on the relations of the object with other objects. In our domain, we are particularly interested in predicting the acquaintance between users, i.e., the *foaf:knows* property. For this purpose, based on our common sense about the domain, we defined three simple first-order logic rules to perform this task:

Table 2. Link prediction rules.

Weight	Formula
0.09	$knows(x,y) \wedge knows(y,z) \Rightarrow knows(x,z)$
2.70	$knows(x,y) \Leftrightarrow knows(y,x)$
1.11	$currentProject(x,z) \wedge currentProject(y,z) \Rightarrow knows(x,y)$

The first two rules define *knows* as a transitive and symmetric property, respectively, while the last rule states that if two persons work on the same project, they probably know each other. Weights were learned generatively with all the individuals available. To better describe the results of the link prediction, we developed a simple artificial example composed by 9 users and 3 projects. Next, using MC-SAT, we queried for the conditional probabilities of the *foaf:knows* property for all those users. A graphical representation of the example, accompanied by a table with the results, is provided.

Fig. 1. Graphical representation of the artificial example. Users are represented by circles (A-I) and projects by squares (P1-P3). Black directed edges represent the *foaf:knows* relation, while gray undirected edges represent the *foaf:currentProject* relation.

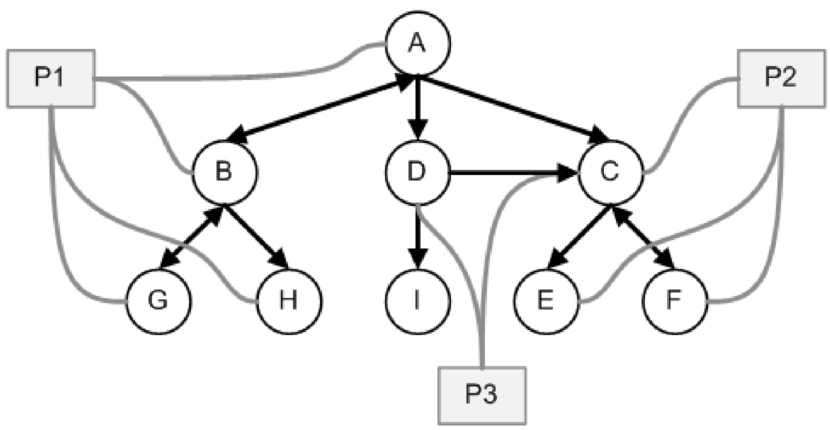


Table 3. *foaf:knows(x,y)* results of the previous example. Columns represent the *x*, lines the *y* (e.g., *P(foaf:knows(A,G))* = 0.90).

	A	B	C	D	E	F	G	H	I
A	0.83	0.97	0.95	0.93	0.48	0.50	0.91	0.82	0.47
B	1.00	0.97	0.48	0.50	0.44	0.43	1.00	0.98	0.48
C	1.00	0.49	0.93	1.00	0.97	1.00	0.44	0.46	0.47
D	1.00	0.50	0.98	0.84	0.47	0.48	0.50	0.50	0.92
E	0.46	0.44	1.00	0.48	0.86	0.86	0.45	0.43	0.41
F	0.48	0.43	1.00	0.48	0.85	0.86	0.43	0.43	0.41
G	0.90	1.00	0.46	0.49	0.45	0.43	0.84	0.89	0.42
H	0.83	1.00	0.49	0.51	0.43	0.45	0.89	0.83	0.45
I	0.49	0.45	0.46	1.00	0.40	0.42	0.42	0.44	0.59

Some interesting results can be seen in this example:

- *knows(A,G)* is greater than *knows(A,F)*, even if both users are at the same distance from *A*. The only difference between them is that *G* works in the same project than *A*, getting a bigger probability;
- *knows(D,A)*, *knows(C,A)*, and *knows(C,D)* have big probabilities, mostly because the symmetry of *knows*. However, the probability of *knows(C,D)* is the greatest, since both users also work in the same project, *P3*;
- Since *H* and *F* doesn't share any direct connection, the probability of *knows(H,F)* is low, but not null.

5.2.2 Link-based Classification

The main task in link-based classification [15] is to predict the category of an object based on the relations of that object with other objects. In our domain, there are three groups of users related to the experience of the user in the community: *Apprentice*, *Journeyer*, and *Master*. These groups are expressed through the *foaf:member* property. The objective of this experiment is to predict each user's group based on their connections to other users. For this purpose, we defined another simple rule that uses the relationship between users expressed on the three previous rules:

Table 4. Link-based classification rule.

Weight	Formula
0.19	$knows(x,y) \wedge member(x,z) \Rightarrow member(y,z)$

This rule states that the group of a user is influenced by the groups of the users that he knows. The weight of the rule was learned generatively in conjunction with the three rules of the previous experiment (their weights remained very similar). Next, we extracted a random sub-network composed by 172 users (11 Apprentices, 55 Journeyers, 93 Masters) and 54 projects and randomly removed the group information to 27% of the users (i.e., 47 users). With the rules of Table 2 and Table 4 and the sub-network individuals, we used MC-SAT to predict the membership of the missing group users. The results can be seen in the next table.

Table 5. Link-based classification results. Between brackets is the number of individuals of the group.

Group	Specificity	Precision	Recall	F-measure
Apprentice (4)	0.98	0	0	0
Journeyer (15)	0.97	0.83	0.33	0.48
Master (28)	0.37	0.7	1	0.82
Weighted Avg	0.61	0.68	0.70	0.64

Good results can be achieved on predicting user's groups taking only in account the relational structure of the network. The bad results on predicting the *Apprentice* group are probably derived from the small number of elements of that group in the test network. The results could be probably improved if other non-relational information about users was provided (e.g., nationality, age, sex).

5.2.3 Link-based cluster analysis

In the last experiment, we had seen how to classify users in a set of predefined groups. However, in some cases, the information about groups is not available and we still need to segment the users. The goal of link-based cluster analysis [15] is to cluster objects into groups that show similar relational characteristics. In our domain, it is interesting to cluster users given their acquaintances with other users. For this task, we can use the three rules presented in the link prediction task, since they can gave us a relational matrix of the *foaf:knows* property for all the users (i.e., the probability of all the users know each other). Using the same sub-network of the last task (172 users and 54 projects), we used MC-SAT with the previously referred rules to predict the *foaf:knows* property for all the 172 users. With those results, we applied two distinct clustering techniques: the general purpose *k-means* clustering algorithm [16], and the *Markov Cluster Algorithm*[15] (MCA) [17], an unsupervised graph clustering algorithm.

After some initial experimentation, we defined the number of desired clusters in the k-means algorithm to 3, and the *inflation* property of the MCA to 1.6 (which also produces 3 clusters). Since the initialization of cluster centroids in k-means is random, the algorithm was run 100 times and the best solution is the one presented. Table 6 provides the cluster sizes and the number of shared members between solutions.

Even if the underlying techniques are conceptually distinct, both solutions provide similar clusters, both in size and composition. The biggest clusters from both solutions (*C1* and *K1*) are very similar, as well the second biggest clusters (*C2* and *K2*).

[15] http://micans.org/mcl/

Table 6. Link-based clustering analysis results. The table represents the number of shared members between the clusters of the two algorithms (e.g., cluster C2 and K2 share 25 individuals). Between brackets is the size of each cluster.

		K-means		
		K1 (114)	K2 (47)	K3 (11)
MCA	C1 (135)	**102**	22	11
	C2 (30)	5	**25**	0
	C3 (7)	7	0	**0**

6 Future Work

During our experimentations, we had some problems in finding interesting ontologies with a sufficient number of individuals that allowed learning the weights with some confidence in the results. This is mainly due to the fact that a large number of Semantic Web ontologies currently available were made to model pure terminological domains, with the main objective of answer questions about concepts and not individuals. In these ontologies, we have to find other ways of gathering information to learn the uncertainty of the axioms. We identified four main approaches to tackle this problem:

- *Learn individuals.* This is the task studied in the field of ontology population [18]. By using previously trained classifiers or general syntactic rules, we can extract information about ontology individuals and their relations from textual corpus. Other way of populating ontologies is through the analysis of structured data, like relational databases or other ontologies. In this case, mappings [19] must be made between the structured data objects and the entities of the ontology.
- *Learn the uncertainties directly from textual corpus.* This is done by analyzing textual corpus for patterns like “70% of A is B” or “Most of the A’s are B’s”. This can be done again by using previously trained classifiers or general syntactic rules.
- *Use the structure of the ontology.* The structure of the ontology can provide interesting information about the uncertainty of its axioms. Some other works [20] [21] already explored similar approaches in ontologies, however with distinct objectives than ours. The field of network analysis [22] can provide us with some interesting concepts that can be potentially transferred to our specific case.
- *Collective learning of weights.* The idea is to learn the weights collectively from multiple ontologies about the same domain. This task can be achieved by exploring techniques from collective learning fields, like relational reinforcement learning [23].

7 Conclusions

In this paper, we have described an approach to the use of uncertainty reasoning in the Semantic Web using Markov Logic. We have shown how it can be used in practice to perform reasoning with OWL2 ontologies. Our approach enables the reasoning with uncertainty in the Semantic Web with a scalability factor that current tools do not provide. The presented work also addresses an important research question: how to derive uncertainty information from an ontology. The approach for producing this information is based on Markov Logic abilities to represent the world uncertainty. We think that our work constitutes a step forward in the creation of robust reasoning mechanisms for the Semantic Web.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, vol. 284, 2001, pp. 28-37.

[2] T. Lukasiewicz and U. Straccia, "Managing Uncertainty and Vagueness in Description Logics for the Semantic Web," *Web Semantics Sci Serv Agents World Wide Web*, 2008.

[3] P. Domingos, S. Kok, D. Lowd, H. Poon, M. Richardson, and P. Singla, "Markov Logic," *Probabilistic Inductive Logic Programming*, 2008, pp. 92-117.

[4] J. Hendler, "Agents and the Semantic Web," *IEEE Intelligent Systems*, vol. 16, 2001, pp. 30-37.

[5] B.C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler, "OWL 2: The next step for OWL," *Web Semantics: Science, Services and Agents on the World Wide Web*, 2008.

[6] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, 2007.

[7] D. Roller, N. Friedman, L. Getoor, and B. Taskar, "Graphical Models in a Nutshell," *Introduction to Statistical Relational Learning*, The MIT Press, 2007, pp. 13-55.

[8] J. Besag, "Statistical analysis of non-lattice data," *The Statistician*, vol. 24, 1975, pp. 179-195.

[9] H. Poon and P. Domingos, "Sound and Efficient Inference with Probabilistic and Deterministic Dependencies," *Proceedings of the National Conference on Artificial Intelligence*, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, pp. 458-463.

[10] W. Wei, J. Erenrich, and B. Selman, "Towards Efficient Sampling: Exploiting Random Walk Strategies," *Proceedings of the National Conference on Artificial Intelligence*, San Jose, CA: AAAI Press, 2004, pp. 670-676.

[11] S. Kok, P. Singla, M. Richardson, and P. Domingos, *The Alchemy system for statistical relational AI*, Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 2007.

[12] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical OWL-DL reasoner," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, 2007, pp. 51-53.

[13] P. Klinov, "Pronto: A Non-monotonic Probabilistic Description Logic Reasoner," *The Semantic Web: Research and Applications*, 2008, pp. 822-826.

[14] L. Getoor and C.P. Diehl, "Link mining: a survey," *SIGKDD Explor. Newsl.*, vol. 7, 2005, pp. 3-12.

[15] L. Getoor, “Link mining: a new data mining challenge,” *SIGKDD Explor. Newsl.*, vol. 5, 2003, pp. 84-89.
[16] J. Marques de Sá, *Pattern Recognition: Concepts, Methods and Applications*, Springer, 2001.
[17] S. Van Dongen, “Graph Clustering by Flow Simulation,” PhD Thesis, University of Utrecht, 2000.
[18] P. Cimiano, *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*, Springer, 2006.
[19] J. Euzenat and P. Shvaiko, *Ontology Matching*, Springer, 2007.
[20] P. Gomes, “Software Design Retrieval Using Bayesian Networks and WordNet,” *Advances in Case-Based Reasoning*, 2004, pp. 184-197.
[21] H. Stuckenschmidt and M. Klein, “Structure-Based Partitioning of Large Concept Hierarchies,” *The Semantic Web – ISWC 2004*, 2004, pp. 289-303.
[22] U. Brandes and T. Erlebach, *Network Analysis: Methodological Foundations*, Springer, 2005.
[23] P. Tadepalli, R. Givan, and K. Driessens, “Relational Reinforcement Learning: An Overview,” *Proceedings of the ICML'04 Workshop on Relational Reinforcement Learning*, 2004, pp. 1–9.

Using a Contextual Logic Programming Language to Acess Data in Warehousing Systems

Valéria Pequeno, Salvador Abreu, and João Carlos Moura Pires

CENTRIA, Departamento de Informática,
Faculdade de Ciências e Tecnologia, FCT, Universidade Nova de Lisboa
2829-516, Caparica, Portugal
http://www.fct.unl.pt

Abstract. Data Warehouses (DWs) are repositories containing the unified history of an enterprise used by decision makers for performance measurement and decision support. The data must be Extracted from heterogeneous information sources, Transformed and integrated to be Loaded (ETL) into the DW, using ETL tools, which are mainly procedural. This means that the knowledge of the procedures and adopted policies are hidden in several programs (written in different languages and paradigms). We propose a development of a framework to declaratively model the ETL process, including the semantic correspondence between schema's components, to provide a better understanding of the semantic associated with the ETL process. A prototype, based on contextual logic programming with persistence, is presented.

1 Introduction

A Data Warehouse (DW) is an integrated data repository that represents the unified history of an enterprise at a suitable level of detail to be useful for analysis [1]. The data must be Extracted from different information sources, Transformed and integrated to be Loaded (ETL) into the DW. DW data is then delivered to Data Marts (DM), probably with some more changes. DMs are subsets of DW data designed to serve specific demands of a particular group of users. Moreover, a DW or a DM should be created and accessed through metadata that provides detailed documentation for data in the DW system, such as applied transformations and origin of data.

The design and population of DW through ETL processes is a difficult and time-consuming task involving considerable cost in human and financial resources. Data models complexity expands both in sources and in DW, which gives rise to the difficulty of managing and understanding these models [2, 3]; and data volumes growing at a significant pace. Although there are specialised tools with graphical interface to do the mapping between the source information and the DW system, they are mostly procedural. This means that the knowledge of procedures and policies are hidden in diverse codes, which are totally dependent on experts and technicians. These tools focus strongly on data movement, as the models are only used as a means to this aim.

An ETL process, for building a DW system, is not concerned just with mapping between schemata, but also with an effective data integration that expresses a unified view of the enterprise. In this context, it is crucial to have a conceptual reference model [4–6]. A Reference Model (RM) is an abstract framework that provides a common semantic that can be used to guide the development of other models and help with data consistency [5].

It is proposed in [7] to take a declarative approach, which is based on making clear the relationship between data sources and DW using correspondence assertions and taking into account the Reference Model, independently of the ETL process involved. Furthermore, the ETL process itself can use this information.

A proof-of-concept prototype, which is the basis of this paper, has been implemented using a standard programming language Prolog, as well as a logic programming framework called Information Systems COnstruction language (ISCO). ISCO is based on a Contextual and Constraint Logic Programing [8]. It allows the construction of information systems, which can transparently access data from various heterogeneous sources in a uniform way, like a mediator system [9]. This paper describes as ISCO can be used to access data in a reference model-based warehousing environment.

The remainder of this paper is structured as follows. Section 2 shows the workings of conceptual design of the ETL processes as well as the motivation to this research. Section 3 describes our reference model-based data warehouse architecture. Contextual Logic Programming and ISCO tool are briefly approached in Section 4. Section 5 deals with the issue of modelling and representation in the data warehouse and ISCO framework. Section 6 illustrates some details of implementation. The paper ends with Section 7, which points out the new features of the approach presented here and planned future work on this topic.

2 Research motivation

Nowadays, there is a plethora of commercial ETL tools in the market place, but very few of them are from academic work. Most of the tools suggest reduced support at the conceptual level.

In the academic area, ETL research for DW environment is focused mainly on the process modelling concepts, data extraction and transformation, and cleaning frameworks [10–12]. So far, the authors of this paper are not aware of any research that precisely deals with both mappings (structural and instance) between the sources and the DW, and with the problem of semantic heterogeneity in a whole conceptual level. There are few prototypes, which usually are implemented to perform technical demonstration and validation of the developed research work [13, 11]. As example of works that developed some implementation, we quote [4, 14, 15]. Reference in [4] presents a methodology that was applied in the TELECOM ITALIA framework. Similar to our work, their proposal include a reference model (cited as "enterprise model") designed using an Enriched Entity-Relationship (EER) model. Their prototype focused on logical schemata and on data movement, any transformation (e.g. restructuring of schema and values) or mapping of instances were deferred for the logical level. In our approach everything stays at the conceptual level. References [14, 15] use ontologies as a common data model to deal with the data integration problem. Skoutas and Simitsis in [14] use a graph-based representation to define the schemata (source and DW) and an ontology, described in OWL-DL. Based on this ontology and the annotated graphs, automated reasoning techniques are used to infer correspondences and conflicts between schemata. Salguero et. al. in [15] extended OWL with temporal and spatial elements, and used the annotation properties of OWL to store metadata about the temporal features of information sources. References [14, 15] mentioned nothing about the mapping of instances, neither they use a reference model in their architecture.

In a data integration scenario, other than DW, there are several works available, mainly to establish the structural mapping between the sources and the global schema (see [16–18] for a survey.); and some works involving the problem to map instances that represent the same entity in the real-world, the instance matching problem (see [18, 19] for a survey.). Approaches for structural matching of schemata focus on schema matching, i.e., on (semi-) automatically identifying semantic correspondences between schema components. In order to do this, the proposed techniques ([20–22]) exploit several kinds of information, including schema characteristics, background knowledge from dictionaries and thesauri, and characteristics of data instances. Approaches for dealing with the instance matching problem cover several kinds of data, such as object, tuples, web data, etc., and many different strategies, including look-up tables, heuristics, etc.. These topics were not considered in our research, as we focused on making explicit the relationship between schemas (in terms of both structure and instance).

In the ETL market, some approaches focus on code generation from specifications of mapping and data movement, which are designed by Information Technology (IT) specialists using graphical interfaces [23]. It is the case, e.g., of Pentaho Kettle, an open-source ETL tool, which has an easy-to-use graphical interface and a rich transformation library, but the designer only works with pieces of structures. Others ETL approaches focus on representation of ETL processes [23, 24]. Orchid [24], for instance, is a system part of IBM Information Server that facilitates the conversion from schema mappings to ETL processes and vice-versa. Some Database Management Systems (DBMS) vendors have embedded ETL capabilities in their products, using the database as "engine" and Structured Query Language (SQL) as supporting language.

Also in market ETL tools can be found that do not depend on any particular database technology, allowing easy integration with Business Intelligence (BI) projects deployment (e.g. Oracle Data Integration). Further, there are ETL tools metadata-driven, which are becoming the current trend with ETL data processing. This approach addresses complexity, meets performance needs, and also enables re-use. Informatica PowerCenter was the pioneer. Many of these tools have integrated metadata repositories that can synchronise metadata from (source) systems, databases and other BI tools. The metadata is represented by proprietary scripting languages, which run within a centralised ETL server [25].

Essentially, the ETL tools are procedural. This means that the knowledge of the procedures and adopted policies are hidden in several programs (written in different languages and paradigms), as well as it is strongly dependent on experts and technicians. An other feature in the ETL process is that the data and business rules evolve requiring the ETL code to be modified and maintained properly. An additional difficulty occurs when ETL tools are used in a data integration context, since each one manages metadata differently. Furthermore, there is not a standard to draw models or to describe data.

This work intends to address the problems stated above. The research focuses on a declarative approach, since a logic-based formalism allow us to deal with the complexity of managing data warehouses and the associated ETL processes in a concise and very perceptible way. Moreover, the semantic integration between the different data sources is accomplished using correspondence assertions to relate concepts from various sources.

3 Data Warehouse Architecture

Our proposal for DW organisation, presented in [7], offers a way to express the existing data models (source, DW, DM, RM) and the relationship between them. The approach is based on *Schema language* (L_S) and *Perspective schema language* (L_{PS}).

Schema language (L_S) is used to describe the actual data models (source, DW, DM, RM). The formal framework focuses on an object-relational paradigm, which includes definitions adopted by the main concepts of object and relational models as they are widely accepted in literature – cf. [26, 27].

Perspective schema language (L_{PS}) is used to describe *perspective schemata*. A perspective schema is a special kind of schema that describes a data model (part or whole) (*target schema*) in terms of other data models (*base schemata*). In Fig. 1, $\mathbf{P}_{S|DW}$ is a perspective schema whose base schema is the source schema **S** and the target schema is the data warehouse schema **DW**.

A simple sales scenario is used as a running example through out this paper. The example consists of two schemata: **S** and $\mathbf{P}_{S|DW}$, shown in Fig. 1. **S** and $\mathbf{P}_{S|DW}$ include information about sales of products, being that $\mathbf{P}_{S|DW}$ contains summarised information regarding sales.

L_{PS} mainly extends L_S with two components: Correspondence Assertions (CAs) and Matching Functions (MFs). Correspondence Assertions formally specify the relationship between schema components in a declarative fashion. CAs are classified in four groups: Property Correspondence Assertion (PCA), Extension Correspondence Assertion (ECA), Summation Correspondence Assertion (SCA), and Aggregation Correspondence Assertion (ACA). Property CAs relate properties of a target schema to the properties of base schemata. The Extension CAs are used to describe which objects/tuples of a base schema should have a corresponding semantically equivalent object/tuple in the target schema. The

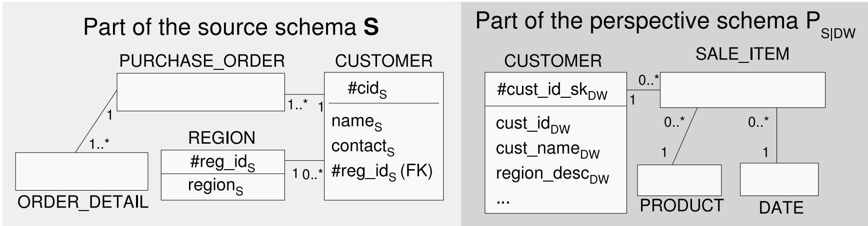


Fig. 1. Simple sales example - partial representation.

Summation CAs are used to describe the summary of a class/relation whose instances are related to the instances of another class/relation by breaking them into logical groups that belong together. They are used to indicate that the relationship between the classes/relations involve some type of aggregate functions or a normalisation process. The Aggregation CAs link properties of the target schema to the properties of the base schema when a SCA is used. Examples of CAs are shown in Fig. 2.

Property Correspondence Assertions (PCAs)
ψ_1: $\mathbf{P_{S\|DW}}$ [CUSTOMER] • **cust_id**$_{DW}$ → **S** [CUSTOMER] • **cid**$_S$
ψ_2: $\mathbf{P_{S\|DW}}$ [CUSTOMER] • **cust_name**$_{DW}$ → **S** [CUSTOMER] • **name**$_S$
ψ_3: $\mathbf{P_{S\|DW}}$ [CUSTOMER] • **region_desc**$_{DW}$ → **S** [CUSTOMER] • FK_2 • region$_S$
Extension Correspondence Assertion (ECA)
ψ_4: $\mathbf{P_{S\|DW}}$ [CUSTOMER] → **S** [CUSTOMER]

Fig. 2. Examples of correspondence assertion.

In Fig 2, the CA ψ_4 defines that the relation CUSTOMER of perspective schema $\mathbf{P_{S|DW}}$ is semantically equivalent to relation CUSTOMER of schema **S**. It means that for each instance **o** in **S**.CUSTOMER, there is a correspondent instance **o′** in $\mathbf{P_{S|DW}}$.CUSTOMER such that **o** and **o′** represent the same entity in the real world. The CAs ψ_1, ψ_2, and ψ_3 define the relationship between the properties of relations $\mathbf{P_{S|DW}}$.CUSTOMER and **S**.CUSTOMER. The property **region_desc**$_{DW}$ is not directly related to a property of $\mathbf{P_{S|DW}}$.CUSTOMER, but to the property **region**$_S$ of relation $\mathbf{P_{S|DW}}$.REGION through the path expression FK_2 • region$_S$ (FK_2 is the name of the foreign key in $\mathbf{P_{S|DW}}$.CUSTOMER that refers to $\mathbf{P_{S|DW}}$.REGION).

Matching functions indicate when two data entities represent the same instance of the real world. These functions, as occur in [28], define a 1:1 correspondence between the objects/tuples in families of corresponding classes/relations. In particular, the work in [7] based on matching function signatures, being that their implementation shall be externally provided, since their code is very close to the application domain.

Fig. 3 illustrates the basic components of the proposed architecture and their relationships. The schemata **RM**, **DW**, **DM**, $\mathbf{S}_1$,...,$\mathbf{S}_n$ are defined using the lan-

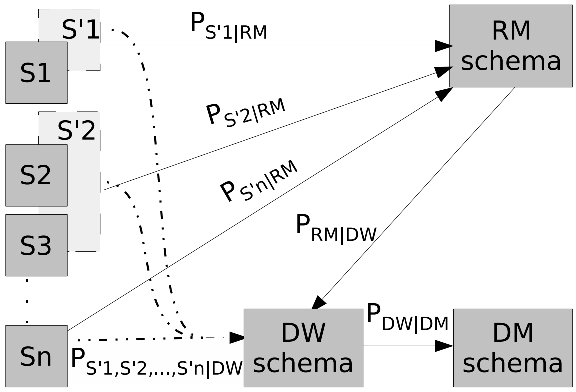


Fig. 3. Proposed architecture.

guage L_S and represent, respectively, the reference model, the data warehouse, a data mart, the source schemata $\mathbf{S}_1$,...,$\mathbf{S}_n$. The schemata $\mathbf{S'}_1$ and $\mathbf{S'}_2$ are defined using the language L_{PS}. They are special kinds of perspective schemata (called *view schema*), since the target schema is described in the scope of a perspective schema, instead of just referring an existing schema. $\mathbf{S'}_1$ and $\mathbf{S'}_2$ represent, respectively, the view schemata $\mathbf{S'}_1$ (a viewpoint of schema $\mathbf{S}_1$), and $\mathbf{S'}_2$ (an integrated viewpoint of schemata $\mathbf{S}_2$ and $\mathbf{S}_3$). The relationships between the target schema and the base schemata are shown through the perspective schemata $\mathbf{P}_{s'1|RM}$,..., $\mathbf{P}_{s'n|RM}$,$\mathbf{P}_{RM|DW}$, and $\mathbf{P}_{s'1,s'2,...,s'n|DW}$ (denoted by arrows).

In [7] is proposed an inference mechanism that, given a set of both schemata and perspective schemata as base, and a perspective schema as target, can deduce a new perspective schema. In context of the Fig. 3, the perspective schema $\mathbf{P}_{s'1,s'2,...,s'n|DW}$ can be automatically deduced by the inference mechanism, having as base the schemata $\mathbf{S}_1$,...,$\mathbf{S}_n$ and the perspective schemata $\mathbf{P}_{s'1|RM}$,..., $\mathbf{P}_{s'n|RM}$, and as target the perspective schema $\mathbf{P}_{RM|DW}$. For a more detailed and formal description of L_S and L_{PS} languages, the reader is referred to [29, 30, 7].

4 Contextual Logic Programming and ISCO

Logic Programming languages are akin to relational databases but provide a significantly higher expressive power, due to their two fundamental mechanisms of nondeterminism and unification, both of which form the basis of the Prolog language. However, it can be argued that standard Prolog is lacking in several areas, which include program structuring facilities and data persistence management. The ISCO programming system addresses both of these issues.

Contexts: the purpose of Contextual Logic Programming (CxLP) was initially to deal with Prolog's traditionally flat predicate namespace, which seriously hindered its usability in larger scale projects. A more recent proposal [31] rehabilitates the ideas of CxLP by viewing contexts not only as shorthands for a modular theory but also as the means of providing dynamic attributes which affect that theory: we are referring to unit arguments, as described in Abreu and Diaz's work [32]. It is particularly relevant for our purposes to stress the *context-as-an-implicit-computation* aspect of CxLP, which views a context as a first-class Prolog entity – a term, behaving similarly to an object in what it carry state (the unit argument terms) and respond to messages (goals evaluated in context).

Persistence: having persistence in a Logic Programming language is a required feature if one is to use it to construct actual information systems; this could conceivably be provided by Prolog's internal database but is best accounted for by software designed to handle large quantities of factual information efficiently, for instance relational database management systems. The semantic proximity between relational database query languages and logic programming makes the former a privileged candidate to provide Prolog with persistence.

ISCO [8] is a proposal for Prolog persistence which includes support for multiple heterogeneous databases and which extends access to technologies other than relational databases, such as LDAP directory services or, more significantly, the semantic web in the form of SPARQL queries over OWL ontologies [33]. ISCO has been successfully used in a variety of real-world situations, ranging from the development of a university information system to text retrieval or business intelligence analysis tools [8].

ISCO's approach for interfacing to DBMSs involves providing Prolog declarations for the database relations, which are equivalent to defining a corresponding predicate, which is then used as if it were originally defined as a set of Prolog facts. While this approach is convenient, its main weakness resides in its present inability to relate distinct database goals, effectively performing joins at the Prolog level. While this may be perceived as a performance-impairing feature, in practice it's not the show-stopper it would seem to be because the instantiations made by the early database goals turn out as restrictions on subsequent goals, thereby avoiding the filter-over-cartesian-product syndrome.

5 Prototype architecture

The present proposal has implemented a proof-of-concept prototype using a Prolog language. The prototype comprises six cooperating modules, namely the *schema manager*, the *inference mechanism*, the *schema repository*, the *ISCO translator*, the *ISCO-generated applications*, and the *ISCO repository*. The architecture of the prototype is depicted in Fig. 4.

The *schema manager* module was written using native-prolog. It is used by the designer to manage the schemata (in language L_S) as well as the perspective schemata (in language L_{PS}). The designer, using the *schema manager*, should define all source schemata, the data warehouse schema, the reference model schema, and perspective schemata that he/she needs.

The *inference mechanism* has been written using native-prolog. It is a rule-based rewriting system that automatically generates new perspective schemata

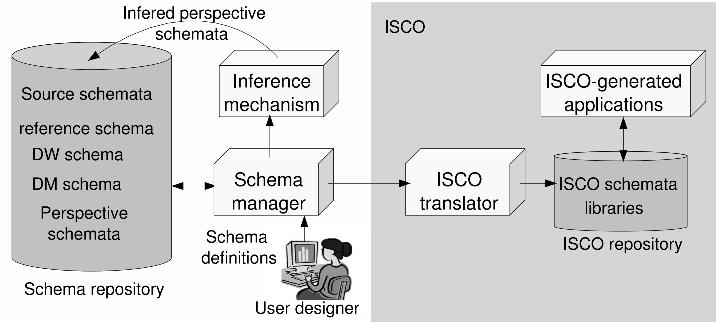


Fig. 4. Prototype architecture.

based on previous ones. It is formed by rules for rewriting CAs, rules for rewriting match function signatures, and rules for rewriting components that are presents in CAs or in matching function signatures, being a total of 39 rules.

The *ISCO translator* performs the mapping between schemata written in L_S or L_{PS} languages to ISCO schemata. Classes or relations of schemata written in language L_S are directly mapped classes in ISCO, which are also called " classes" and which compile to regular Prolog access predicates. Classes or relations of perspective schemata (written in language L_{PS}) are also mapped to ISCO classes based on CAs and match function signatures. In the current implementation, when perspective schemata are translated to ISCO schemata, it is assumed that their base schemata were already mapped to ISCO. *ISCO translator* has been written in native prolog. All functions, inclusive of the match functions, declared in the original perspective schemata are defined in a library called *v_utils*, which is common to all ISCO schemata created (more details in the next section).

The *ISCO-generated applications* includes all files that are necessary to access data from information sources. So, data in any perspective schema mapped to ISCO, specifically any inferred perspective schema between the DW and its sources, can be queried in a transparent way, just as in a mediator approach (more details in the next section).

The *schema repository* stores both the schemata (in language L_S) and perspective schemata (in language L_{PS}), including any inferred perspective schema created by the inference mechanism, while the *ISCO repository* is used to store ISCO schemata and ISCO files (libraries, units, etc.).

The next Section present implementation details using the running example to describes the process of ISCO-generation applications in warehousing environments using the implemented prototype.

6 Implementation issues

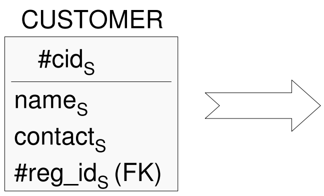
Each class or relation in the (perspective) schemata are mapped to ISCO classes using the *ISCO translator*. The ISCO classes may reflect inheritance, keys, indexes, foreign keys and sequences to name a few. The process involved differ enough, depending on whether the original schema was written in L_S language or is a perspective schema.

In the case of schemata defined in L_S language, a declaration is added to ISCO schema in order to provide ISCO with the necessary information to access an external data source, such as an ODBC-accessed database. In the context of the running example (see Fig. 1), the ISCO schema that is mapped from schema **S** shall contain the clausule:

external(**S**, postgres(S)).

This clausule means that **S** is an outside database hosted in PostgreSQL.

All classes or relations in the original schema are simply mapped to ISCO classes, which should be declared as *external* and *mutable*. External means that the class has been created in an independent database and mutable means that its instances can change. For instance, the relation **S**.CUSTOMER is mapped to ISCO as illustrated in Fig 5.

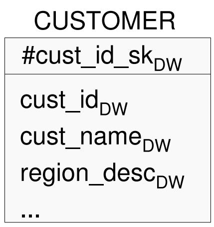



```
01. external(S,customer),
02. mutable class customer.
03.     cids: int. key.
04.     names: text.
05.     contacts: text.
06.     reg_ids: region.reg_ids.
```

Fig. 5. Example of the mapping of a relation (in a schema) to a ISCO class.

In Fig. 5, lines 1 and 2 mean that the instances of ISCO class CUSTOMER are in database **S**, class CUSTOMER, line 3 means that the property **cid**$_S$ is an integer number and a primary key, and line 6 means that the property **reg_id**$_S$ is a foreign key that refers to class REGION through property **reg_id**$_S$. Note that, in this example, the ISCO class and the class in the database have the same name, but could be different ones. The ISCO class CUSTOMER defines the predicate **customer/4**, which behaves as a database predicate but relies on an external system (in this case the PostgreSQL ORDBMS) to provide the actual facts. In the current implementation, it is assumed that all instances of ISCO classes have an object identity (OID), which is a integer number automatically generated by the system.

In the case of perspective schemata, the classes or relations are usually mapped to *computed* classes. It means that the class instances will be generated each time that a query is made to the class, similar to the concept of view in SQL. Computed classes are expected to contain one or more rules. These rules define how the computed class instances are obtained and always come after the computed class definition. In the body of each rule, the variables, which represent the computed class arguments, must have the same name of the respective argument that they represent and they must be all in uppercase. This is necessary in order for Prolog to link each variable with its respective computed class argument correctly. Classes or relations in the perspective schema are mapped to computed classes in ISCO when they are related to classes or relations in a base schema through some extension correspondence assertion (ECA). A example is illustrated in Fig. 6.




```
01. computed class customer.
02.     cust_id_skdw: int. key.
03.     cust_iddw: int.
04.     cust_namedw: text.
05.     region_descdw: text.

06. rule:- s:> customer@(oid=Osource,cids=CUST_IDDW,
        names=CUST_NAMEDW,reg_ids=Var1),
07.     s:> region@(reg_ids=Var1,regions=Var2),
08.     REGION_DESCDW=Var2,
09.     CUST_ID_SKDW=⟨⟨createSKey⟩⟩,
10.     OID=⟨⟨createNewOID⟩⟩.
```

Fig. 6. Example of the mapping of a relation (in a perspective schema) to a ISCO class.

In Fig 6, line 6 defines a query to the ISCO class CUSTOMER in source schema **S**. This query is obtained using the CAs $\psi_1 - \psi_4$ (see Fig. 2), being that the values of **cust_iddw** and **cust_namedw** are acquired directly from this query while the value of **region_descdw** requires an additional query to ISCO class REGION in schema **S** (lines 7 and 8). Lines 9 and 10 define the necessary steps to generate, respectively, surrogate keys and oids for the computed class. The surrogate key is an integer number automatically generated based on the object identity in variable "Osource". The oid for the computed class (OID) is a compound object identity with the following structure:

$$\text{oid(schema,class,oids)},$$

being that *class* is the name of the computed class which OID belongs, *schema* is the name of the schema which *class* belongs, and *oids* is a list of compound object identities in the form of (S′,C′,**o′**), with C′ being the name of a class or relation in a schema S′ containing the object identity **o'** from what the OID is derived. The object identity **o′**, in turn, can be a compound object identity or a simple object identity (an integer number). For example, OIDs for computed classes may look like:

$$\text{oid}(\mathbf{P}_{S|DW},\text{CUSTOMER},\text{oid}(\mathbf{S},\text{CUSTOMER},667789))$$

which means the object in the computed class CUSTOMER is derived from object in **S**.CUSTOMER whose OID is 667789.

Once having the ISCO schemata, the following phase is to generate a GNU Prolog/CX executable containing the native-code executable version of all ISCO predicates. GNU Prolog/CX compiles Prolog (and ISCO) programs to native executables. Each schema and perspective schema described in ISCO, as well the library *v_utils*, will correspond to units whose terms can be instantiated and collected into a list to form a context. A set of operations and operators are available in GNU Prolog/CX to construct contexts, being the o more usual in our application the *context extension* operation given by the operator `:>`. The goal **U** `:>` **G** extends the current context with the unit **U** and resolves **G** in the new context, as if it were regular Prolog. For instance, to make a interrogation to the computed class CUSTOMER, we can use the following syntax:

$$\texttt{v\_utils :> P}_{S|DW}\texttt{ :> customer(A,B,C,D).}$$

In this goal, we start by extending the initially empty context with unit v_utils. After, this new context is again extended with the unit $P_{S|DW}$, and it in the latter context that goal `customer(A,B,C,D)` is derived.

7 Conclusions and Future Work

In this article we have discussed an implementation that permits the generation of applications which transparently access source information in a reference model-based warehousing system based on a logic-based formalism. We had access to a system which already provides a Logic-based programming layer while being able to transparently access facts stored in existing RDBMs. Having a logic programming appearance lay down a setting where programs are first-class objects on which meta-reasoning may be performed, including proofs of correctness or explanations for results.

The prototype has been developed using a contextual logic programming with persistence, called ISCO [8]. ISCO provides: a) high levels of performance, by virtue of being derived from GNU-Prolog; b) expressiveness, due to the use of constraint logic programming; c) simplicity; d) persistence; and e) structured code, as a result of using Contextual Logic Programming constructs. Constraints (over finite domains) are used to generate more efficient SQL codes for database predicates, besides being a form of search-space pruning which is complementary to backtracking in that propagation works as an a-priori filter on the search.

ISCO allows access to heterogeneous data sources and to perform arbitrary computations. User-queries can be done in ISCO, in a transparent way to access the information sources, even the data of the DW schema. This feature can be useful in some situations, although a mediator strategy in a DW context, in general, may not be appropriate, due to particular nature of the DW. Specifically, this work can be used in applications that hide from their users the complexity involved in accessing multiple data sources, since these sources are in databases that can be queried via remote access in real-time.

Some improvements of our prototype will be done, namely: to use foreign keys in ISCO classes with aggregate functions, or derived by a normalisation process; improvement the generation of surrogate keys. The final prototype will be applied to various situations, some synthetic and others real, as a means of providing experimental validation of the usefulness of the chosen approaches.

For future work, we are presently working on how the perspective schemata can be used to automate the materialisation of the ETL process. Various other on-going research is related to this work, such as: a) management of schema versioning using contexts in ISCO; and b) incorporating aspects from temporal contextual logic programming [34] to deal with evolving data and schemata, and so to further reduce the complexity of the system, by allowing the temporal component to be abstracted. Another important direction for future work is to develop a graphical user-friendly interface to declare the schemata in our language, and thus, hide some syntax details.

References

1. Inmon, W.H.: Building the data warehouse. 4th edn. Wiley Publishing (2005)
2. Pérez, J.M., Berlanga, R., Aramburu, M.J., Pedersen, T.B.: A relevance-extended multi-dimensional model for a data warehouse contextualized with documents. In: DOLAP'05: Proc. of the 8th ACM Intl. Workshop on Data Warehousing and OLAP, USA, ACM (2005) 19–28
3. Matias, R., Moura-Pires, J.: Revisiting the olap interaction to cope with spatial data and spatial data analysis. In: ICEIS'07 - Proc. of the 9th Intl. Conf. on Enterprise Information Systems. Volume DISI. (2007) 157–163
4. Calvanese, D., Dragone, L., Nardi, D., Rosati, R., Trisolini, S.M.: Enterprise modeling and data warehousing in TELECOM ITALIA. Inf. Syst. **31**(1) (2006) 1–32
5. Imhoff, C., Galemmo, N., Geiger, J.G.: Mastering Data Warehouse Design - Relational and Dimensional Techniques. Wiley Publishing (2003)
6. Knackstedt, R., Klose, K.: Configurative reference model-based development of data warehouse systems. Idea group publishing (2005) 32–39
7. Pequeno, V.M., Pires, J.C.G.M.: Using perspective schemata to model the ETL process. In: Intl. Conf. on Management Information Systems, France (June 2009)
8. Abreu, S., Nogueira, V.: Using a logic programming language with persistence and contexts. In: INAP'05: 16th Intl. Conf. on applications of declarative programming and knowledge management. Volume 4369 of Lecture Notes in Computer Science., Springer (2006) 38–47 (Revised Selected Papers).
9. Wiederhold, G.: Mediators in the architecture of future information systems. In: IEEE Computer. Volume 25(3). (1992) 38–49
10. Lujn-mora, S., Vassiliadis, P., Trujillo, J.: Data mapping diagrams for data warehouse design with UML. In: ER'04: 23rd Intl. Conf. on Conceptual Modeling, Springer (2004) 191–204
11. Raman, V., Hellerstein, J.: Potter's wheel: An interactive data cleaning system. In: 27th VLDB Conference, Italy (2001)
12. Albrecht, A., Naumann, F.: Managing etl processes. In: NTII'08: Intl. Workshop on New Trends in Information Integration, New Zealand (2008) 12–15
13. Vassiliadis, P., Vagena, Z., Skiadopoulos, S., Karayannidis, N., Sellis, T.K.: ARKTOS: towards the modeling, design, control and execution. Information Systems **26(8)** (2001) 537–561

14. Skoutas, D., Simitsis, A.: Ontology-based conceptual design of ETL processes for both structured and semi-structured data. Int. J. Semantic Web Inf. Syst. **3**(4) (2007) 1–24
15. Salguero, A., Araque, F., Delgado, C.: Ontology based framework for data integration. WSEAS Trans. Info. Sci. and App. **5**(6) (2008) 953–962
16. Hai, Do, H.: Schema Matching and Mapping-based Data Integration: Architecture, Approaches and Evaluation. VDM Verlag, Germany (2007)
17. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal **10**(4) (2001) 334–350
18. Doan, A., Halevy, A.Y.: Semantic integration research in the database community: A brief survey. AI Magazine **26**(1) (2005) 83–94
19. Bleiholder, J., Naumann, F.: Data fusion. ACM Comput. Surv. **41**(1) (2008) 1–41
20. Nottelmann, H., Straccia, U.: Information retrieval and machine learning for probabilistic schema matching. Inf. Process. Manage. **43**(3) (2007) 552–576
21. Chiticariu, L., Hernández, M.A., Kolaitis, P.G., Popa, L.: Semi-automatic schema integration in clio. In: VLDB '07: Proceedings of the 33rd international conference on Very large data bases, VLDB Endowment (2007) 1326–1329
22. Engmann, D., Maßmann, S.: Instance matching with COMA++. In: BTW Workshops. (2007) 28–37
23. Kimball, R., Caserta, J.: The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. John Wiley & Sons (2004)
24. Dessloch, S., Hernandez, M., Wisnesky, R., Radwan, A., Zhou, J.: Orchid: Integrating schema mapping and ETL. ICDE'08: IEEE 24th International Conference on Data Engineering (April 2008) 1307–1316
25. Raminhos, R.F.: Extraction and transformation of data from semi-structured text files using a declarative approach. Master's thesis, Faculdade de Ciências e Tecnologia of the Universidade Nova de Lisboa (2007)
26. Codd, E.F.: A relational model of data for large shared data banks. In: Communications of the ACM. (1970) 377–387
27. Cattell, R.G., Barry, D., eds.: The Object Database Standard ODMG 3.0. Morgan Kaufmann Publishers (2000)
28. Zhou, G., Hull, R., King, R.: Generating data integration mediators that use materialization. J. Intell. Inf. Syst. **6(2/3)** (May 1996) 199–221
29. Pequeno, V.M., Pires, J.C.G.M.: A formal object-relational data warehouse model. Technical report, Universidade Nova de Lisboa (November 2007)
30. Pequeno, V.M., Pires, J.C.G.M.: Using perspective schemata to model the ETL process. Technical report, Universidade Nova de Lisboa (2009)
31. Abreu, S., Diaz, D.: Objective: in Minimum Context. In Palamidessi, C., ed.: ICLP'03: 19th Intl. Conf. in Logic Programming. Volume 2916 of Lecture Notes in Computer Science., India, Springer-Verlag (December 2003) 128–147
32. Abreu, S., Diaz, D.: Objective: in minimum context. In Palamidessi, C., ed.: In Proc. of 19th Intl. Conf. in Logic Programming (ICLP 2003), Lecture Notes in Computer Science 2916, Springer-Verlag (2003) 128–147
33. Lopes, N., Fernandes, C., Abreu, S.: Representing and Querying Multiple Ontologies with Contextual Logic Programming. ComSIS: Computer Science and Information Systems **05**(02) (December 2008)
34. Nogueira, V., Abreu, S.: Temporal contextual logic programming. Electron. Notes Theor. Comput. Sci. **177** (2007) 219–233

Chapter 5

EAC - Emotional and Affective Computing

Personality, Emotion and Mood Simulation in Decision Making

Ricardo Santos[1,2], Goreti Marreiros[1,3], Carlos Ramos[1,3], José Neves[4], José Bulas-Cruz[5]

[1] GECAD – Knowledge Engineering and Decision Support Group
Porto, Portugal
{goreti,csr}@dei.isep.ipp.pt
[2] School of Management and Technology of Felgueiras – Polytechnic of Porto
Felgueiras, Portugal
rjs@estgf.ipp.pt
[3] Institute of Engineering – Polytechnic of Porto
Porto, Portugal
{goreti,csr}@dei.isep.ipp.pt
[4] University of Minho
Braga, Portugal
jneves@di.uminho.pt
[5] University of Trás-os-Montes e Alto Douro
Vila Real, Portugal
jcruz@utad.pt

Abstract. In this paper is proposed the integration of personality, emotion and mood aspects for a group of participants in a decision-making negotiation process. The aim is to simulate the participant behavior in that scenario. The personality is modeled through the OCEAN five-factor model of personality (Openness, Conscientiousness, Extraversion, Agreeableness and Negative emotionality). The emotion model applied to the participants is the OCC (Ortony, Clore and Collins) that defines several criteria representing the human emotional structure. In order to integrate personality and emotion is used the pleasure-arousal-dominance (PAD) model of mood.

1 Introduction

Nowadays groups are used to make decisions about some subject of interest for the organization or community in which they are involved. The scope of such decisions can be diverse. It can be related to economic or political affairs like, for instance, the acquisition of new military equipment. But it can also be a trivial decision making as the choice about a holiday destination by a group of friends. Therefore, it may be claimed that Group Decision Support Systems (GDSS) have emerged as the factor that makes the difference one assess the behavior and performance of different computational systems in different applications domains, with a particular focus on socialization. Groups of individuals have access to more information and more resources what will (probably) allow reaching "better" and quicker decisions. However working in group has also some difficulties associated, e.g. time consuming; high costs; improper use of group dynamics and incomplete tasks analysis.

Many of this will take a new dimension if we consider that they will be resolved by a group of individuals, each one with a different type of personality. Our society is characterized by the use of groups to make decisions about some subject of interest for the organization in which they are involved. If we predict the personality of our adversaries we could find the best arguments to be used in the negotiation process in order to reach a consensus or a better decision in the shortest possible time. Emotions have proven effects on cognitive processes such as action selection, learning, memory, motivation and planning. Our emotions both motivate our decisions and have impact on our actions.

The use of multi-agent systems is very suitable to simulate the behaviour of groups of people working together and, in particular, to group decision making modelling, once it caters for individual modelling, flexibility and data distribution [1][2]. Various interaction and decision mechanisms for automated negotiation have been proposed and studied. Approaches to automated negotiation can be classified in three categories [3], namely game theoretic, heuristic and argumentation based. We think that an argumentation-based approach is the most adequate for group decision-making, since agents can justify possible choices and convince other elements of the group about the best or worst alternatives.

Agent Based simulation is considered an important tool in a broad range of areas e.g. individual decision making (what if scenarios), e-commerce (to simulate the buyers and sellers behaviour), crisis situations (e.g. simulate fire combat), traffic simulation, military training, entertainment (e.g. movies).

According to the architecture that we are proposing we intend to give support to decision makers in both of the aspects identified by Zachary and Ryder [4], namely supporting them in a specific decision situation and giving them training facilities in order to acquire skills and knowledge to be used in a real decision group meeting. We claim that agent based simulation can be used with success in both tasks.

In our multi-agent architecture model [5] we have two different types of agents: the Facilitator agent and the Participant agent. The Facilitator agent is responsible for the meeting in its organization (e.g. decision problem and alternatives definition). During the meeting, the Facilitator agent will coordinate all the processes and, at the end, will report the results of the meeting to the participants involved. The Participant Agent will be described in detail in the next section.

In this work is presented a new argumentation process with the inclusion of personality using the Five-Factor Model of Personality (FFM) [6] and emotion using the OCC model [7]. The mood of the participants will also be represented by the use of the PAD mood space [8].

2 Participant Agent

The participant agent has a very important role in the group decision support system assisting the participant of the meeting. This agent represents the user in the virtual world and is intended to have the same personality and to make the same decision as if it were the real participant user. For that reason we will present the architecture and a detailed view of all the component parts. The architecture is divided in three layers: the knowledge layer, the interaction layer and the reasoning layer (Figure 1).

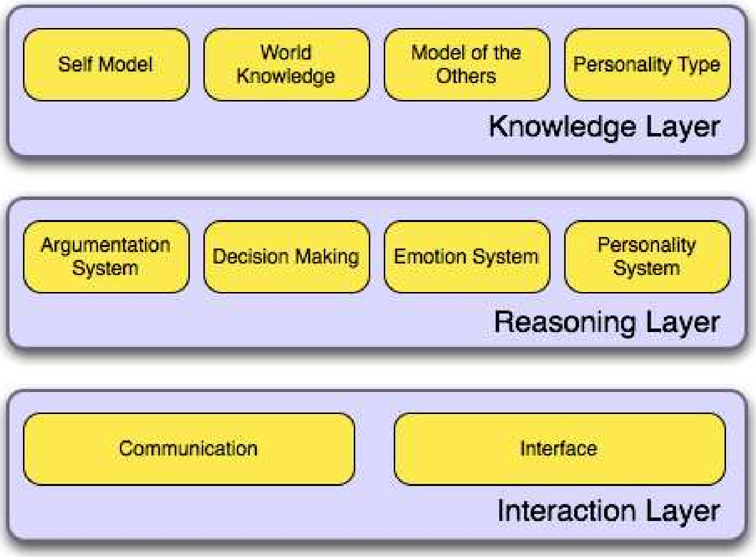


Figure 1 - Participant Agent Architecture

2.1 Knowledge Layer

In the knowledge layer the agent has information about the environment where he is situated, about the profile of the other participant's agents that compose the meeting group, and regarding its own preferences and goals (its own profile).

The personality is defined in this layer through the Big Five Inventory (BFI) [9] and available publicly to be used by the other opponent participants.

The information in the knowledge layer has some kind of uncertainty [11] and will be made more accurate along the time through interactions done by the agent. The credibility of the participants and the perception that one user has about the others will be refined along he time in the "Model of the Others" component.

A database of profiles and history with the group's model is maintained and this model is built incrementally during the different interactions with the system. The community should be persistent because it is necessary to have information about previous group decision making processes, focusing credibility, reputation and past behaviours of other participants.

2.2 Reasoning Layer

The agent must be able to reason based on complete or incomplete information. In this layer the reasoning mechanism is based on the information available in the knowledge layer and on the messages received from other agents through the interaction layer. The reasoning mechanism will determine the behaviour of the agent and allow the acquisition of new knowledge, essentially based on previous experiences.
The reasoning layer contains four major modules:

- The argumentation system – that is responsible for the arguments generation. This component will generate persuasive arguments, which are more related with the internal emotional state of the agent, and about what he thinks from others' profiles (including the emotional state).
- The decision-making module – will support agents in the choice of the preferred alternative. The preferred alternatives are in the self-model of the participant agent, being filtered and sorted by this component.
- The emotional system [1]– will generate emotions and moods, affecting the choice of the arguments to send to the other participants, the evaluation of the received arguments and the final decision. The emotions that will be simulated in our system are those identified in the reviewed version of the OCC (Ortony, Clore and Collins) model: joy, hope, relief, pride, gratitude, love, distress, fear, disappointment remorse, anger and hate.
- The personality system – will identify the personality of the other participants in order to find the best strategy for the argumentation on the negotiation process based on the FFM of personality [12].

2.3 Interaction Layer

The interaction layer is responsible for the communication with other agents and by the interface with the user of the group decision-making system. All the messages received will be sorted, decoded and sent to the right layer based on their internal data. The knowledge that the participant user has about his actions and of the others are obtained through this layer.

3 Personality Type Identification

In order to make agents more human-like and to increase their flexibility to argument and to reach agreements in the negotiation process, we updated the previous agent participant model [13] and included the personality system component.
Personality plays an important role on the behaviors of the participants in a decision meeting. "Behaviors are influenced by personalities so that personality refers to sets of predictive behaviors by which people are recognized and indentified [14]".
The personality is divided in 30 attributes, each one called a personality facet. The personality facets are clustered in five groups, called personality factors or traits. The

five-factor model of personality is best known as OCEAN: Openess, Conscientiousness, Extraversion, Agreeableness, and Negative Emotionality; and is the most widely accepted model of personality [15]. The identification of the personality of each participant its classified using the BFI [9] and fit in one of the themes based on the FFM. A theme is a characteristic personality pattern that reflects the combined effect of two or more factors or facets [16].
There are several types of themes based on the FFM for each set of personality types. For the decision making area the themes that could be applied are: the conflict styles and the decision style. We select the conflict style because we will be using the personality in the negotiation process where many disagreements and conflicts arise. The conflict styles theme uses only four of the five factors of the model that are: the agreeableness, the conscientiousness, the extraversion and the negative emotionality.

4 Emotional System

Our participant agent is composed by an emotional system, which, beside other tasks, will generate emotions. Those emotions are the identified in the revised version of the OCC model [7]: joy, hope, relief, pride and gratitude, love, distress, fear, disappointment, remorse, anger and hate.

Table 1: Revised OCC Model

Positive reaction	**Appraised events**	**Categories**
Joy	Because something good happened	Undifferentiated
Hope	About the possibility of something good happening	Goal-based
Relief	Because a feared bad thing didn't happen	
Pride	About a self-initiated praiseworthy act	Standards-based
Gratitude	About an other-initiated praiseworthy act	
Love	Because a person finds someone or something appealing	Taste-based
Negative reactions	**Appraised events**	**Categories**
Distress	Because something bad happened	Undifferentiated
Fear	About the possibility of something bad happening	Goal-based
Disappointment	Because a hoped-for good thing didn't happen	
Remorse	About a self-initiated blameworthy act	Standards-based
Anger	About an other-initiated blameworthy act	
Hate	Because a person finds someone or something unappealing	Taste-based

The agent emotional state (i.e. mood) is also calculated in this module based on the emotions generated. To model mood we use Albert Mehrabian's pleasure (P), arousal (A) and dominance (D) trait which form the PAD space. These traits are independent of each other and form a 3D space. The pleasure level relates to the emotional state's positivity or negativity, arousal shows the level of physical activity and mental alertness, and dominance indicates the feeling of control. These trait's values lie between the positive (+1) and negative (-1) ends of each dimension. Mehrabian defined eight mood types based on the combinations of negative (-) and positive (+) values for each dimension: pleasant (+P), unpleasant (-P); aroused (A+), unaroused (A-); and dominant (D+), submissive (D-). Table 2 shows all the mood types defined by Mehrabian.

Table 2: Mehrabian Mood Types

Trait combination	Mood type
+P+A+D	Exuberant
-P-A-D	Bored
+P+A-D	Dependent
-P-A+D	Disdainful
+P-A+D	Relaxed
-P+A-D	Anxious
+P-A-D	Docile
-P+A+D	Hostile

Because Mehrabian also defines the relationship between the OCEAN personality traits and the PAD space, we can translate the 5D personality vector (P) into a corresponding PAD space mood point [8]. Considering an OCEAN personality (O, C, E, A, N) the initial mood is calculated in the following way:

InitialMood = (P, A, D)
$P = 0.59 * A + 0.19 * N + 0.21 * E$
$A = 0.57 * N + 0.30 * A + 0.15 * O$
$D = 0.60 * E + 0.32 * A + 0.25 * O + 0.17 * C$

When the system updates the emotional state, the mood point shifts in the 3D PAD space. The change is based on which emotion is activated. Mehrabian defined more than 240 emotions although as we use the revised version of the OCC model that is composed only by 12 we made a correlation between both models [10]. Table 3 shows our correlation between OCC emotions and PAD space [8].

Table 3: Correlation between OCC emotions and the PAD space

Emotion	Pleasure	Arousal	Dominance	Mood type
Joy	0.40	0.20	0.10	+P+A+D Exuberant
Hope	0.20	0.20	-0.10	+P+A-D Dependent
Relief	0.20	-0.30	0.40	+P-A+D Relaxed
Pride	0.40	0.30	0.30	+P+A+D Exuberant
Gratitude	0.40	0.20	-0.30	+P+A-D Dependent

Love	0.30	0.10	0.20	+P+A+D Exuberant
Distress	-0.40	-0.20	-0.50	-P-A-D Bored
Fear	-0.64	0.60	-0.43	-P+A-D Anxious
Disappointment	-0.30	0.10	-0.40	-P+A-D Anxious
Remorse	-0.30	0.10	0.60	-P+A-D Anxious
Anger	-0.51	0.59	0.25	-P+A+D Hostile
Hate	-0.60	0.60	0.30	-P+A+D Hostile

Each participant agent has a model of the other agents, in particular the information about the other agent's mood. This model deals with incomplete information and the existence of explicit negation. Some of the properties that characterize the agent model are: gratitude debts, benevolence, and credibility.
Although the emotional component is based on the OCC model, with the inclusion of mood, it overcomes one of the major critics that usually is pointed out to this model: OCC model does not handle the treatment of past interactions and past emotions.

5 Negotiation Process

For the negotiation the process is divided in three stages: Pre-Negotiation, where the participants should gather and analyse information and set objectives; In-Negotiation: where the participants should analyse, argue, persuade others and achieve and agreement if possible; Post-Negotiation: where the participants should confirm the agreement and review the negotiation. For each one of these stages different emotions are generated. Next we will explain the emotions that are generated for the stages.

5.1 Pre-Negotiation

In the pre-negotiation the agent first does an analysis on the adversaries and next establishes the objectives for the negotiation.

The adversaries' analysis generates taste-based emotions:

- Love – Because a person finds someone or something appealing;
- Hate – Because a person finds someone or something unappealing.

Establishing the objectives to the negotiation generates goal-based emotions:

- Hope – About the possibility of something good happening;
- Fear – About the possibility of something bad happening.

5.2 In-Negotiation

The aim of the meeting is to achieve an outcome which both sides can accept. The in-negotiation is the most important process because it is where the proposals are exchanged. During the in-negotiation process, participant agents may exchange the following locutions: request, refuse, accept, request with argument.

- Request (AgPi, AgPj, α, arg) - in this case agent AgPi is asking agent AgPj to perform action α, the parameter arg may be void and in that case it is a request without argument or may have one of the arguments specified at the end of this section.
- Accept (AgPj, AgPi, α) - in this case agent AgPj is telling agent AgPi that it accepts its request to perform α.
- Refuse (AgPj, AgPi, α) - in this case agent AgPj is telling agent AgPi that it cannot accept its request to perform α.

The purpose of the participant agent is to assist the user. For example, in Figure 2, it is possible to see the argumentation protocol for two agents. This is the simplest scenario, because in real world situations, group decision making involves more than two agents and, at the same time AgP1 is trying to persuade AgP2, that agent may be involved in other persuasion dialogues with other group members.

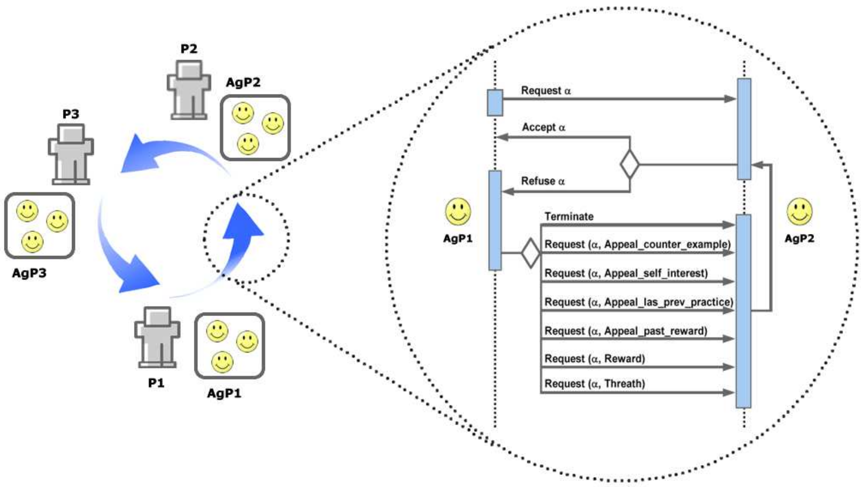


Figure 2 - Argumentation Protocol

Argument nature and type can vary, however six types of arguments are assumed to have persuasive force in human based negotiations [17][7]: threats; promise of a future reward; appeal to past reward; appeal to counter-example; appeal to prevailing practice; and appeal to self interest [18]. These are the arguments that agents will use to persuade each other. This selection of arguments is compatible with the power relations identified in the political model: reward, coercive, referent, and legitimate [19][20].

In the past has been made a study of the impact personality had in the argumentation process [16]. To summarise the previous study of the impacts that personality FFM factors have on each argument intended to be used in the negotiation process, a table

(Table 4) was created to show all the permitted arguments. This table and the study is only applied to the personalities of the conflict styles theme.

Table 4: Possible arguments to each personality on the conflicts style

	Appeal to Self Interest	Appeal to prevailing practice	Appeal to counter example	Appeal to past reward	Reward	Threat
Negotiator	Yes	Yes	Yes	Yes	Yes	Yes
Aggressor	Yes	No	No	Yes	Yes	Yes
Submissive	No	No	Yes	No	No	No
Avoider	No	Yes	Yes	No	No	Yes

This work is also intended to consider emotion in the argumentation process. The arguments defined by Sarit Kraus [18] have a natural order of argument power to be sent: Appeal to self-interest, Appeal to prevailing practice, Appeal to counter example, Appeal to past reward, Reward, Threat. To generate the emotions we divided the arguments in appeals, rewards and threats.

A. Appeals

Appeals generate undifferentiated emotions:

- Joy – Because something good happened;
- Distress – Because something bad happened.

B. Rewards

Rewards generate standards-based and undifferentiated emotions. When the actions are related to whom is making the reward it generates standard-based emotions, when it is an response to an action it generates undifferentiated emotions.

Rewards sent:

- Pride - About a self-initiated praiseworthy act (emotion generated when the reward is sent to a counterpart);
- Joy – Because something good happened (emotion generated in the response to the reward sent);
- Distress – Because something bad happened (emotion generated in the response to the reward sent).

Rewards received

- Remorse - About an other-initiated praiseworthy act.

C. Threats

Threats generate standard-based and undifferentiated emotions. When the actions are related to whom is making the threat it generates standard-based emotions when it is an response to an action it generates undifferentiated emotions.

Threats sent

- Remorse - About an other-initiated praiseworthy act;
- Joy – Because something good happened (emotion generated in the response to the reward sent);
- Distress – Because something bad happened (emotion generated in the response to the reward sent).

Threats received

- Hate - About an other-initiated blameworthy act.

In order to exemplify this process for our multi-agent model [5] a diagram is presented (Figure 3) with two participant agents (AgP1 and AgP2) for a general meeting. To explain the diagram we are going to describe the numbered circles (1, 2, 3, 4, 5 and 6).

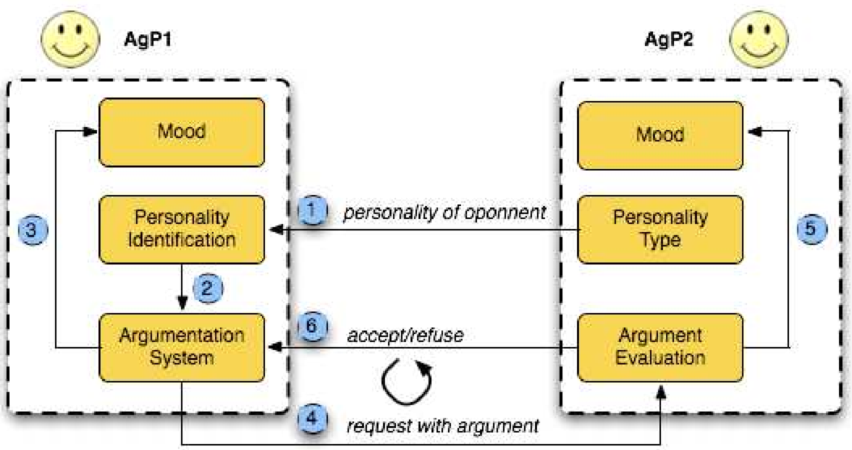


Figure 3 – Argumentation dialog using personality, emotion and mood

In the diagram is possible to see two agents (AgP1 and AgP2) where AgP1 would like to make a request to AgP2. The more important steps that occur in our model are the following:

1. AgP1 receives the personality of AgP2 and proceeds to the personality identification component. In this component the received information verified to see if is compatible with previous negotiations with this participant;
2. The personality type is sent to the argumentation system;
3. The argumentation system updates the mood component based on the emotions generated;
4. The argumentation system component selects the possible set of arguments and starts making a request with the weaker argument;

5. AgP2 receives the request evaluates it, the emotions are generated and updates his mood;
6. AgP2 accepts or refuses the request.

Several iterations can occur in steps 3 to 6, depending on the set of possible arguments to be sent to AgP2.

5.3 Post-Negotiation

The negotiation should be reviewed by all the members to analyze the conclusion and verify if the objectives established at the beginning (Pre-Negotiation) were achieved.

The analysis of the negotiation generates goal-based emotions:

- Relief – Because a feared bad thing didn't happen;
- Disappointment – Because a hoped-for good thing did not happen.

6 Conclusion

This work proposes the inclusion of the personality and emotion in the negotiation process of an argument-based decision-making. In spite of using two different components to model personality and emotion we start using the PAD mood space, which is able to support OCEAN and OCC models. Is proposed as well a mapping of the OCC emotions to the PAD mood space. Each person is unique and has different reactions to the exchanged arguments. Many times a disagreement arises because of the way we began arguing and not because of the content. Our main goal on a decision meeting is to reach consensus where everyone can be satisfied about the result. The principal determinant of a member's degree of satisfaction with his or her group's decision is the extent to which the member agrees with the decision [21].

Each participant agent represents a group decision member. This representation facilitates the simulation of persons with different personalities. The discussion process between group members (agents) is made through the exchange of persuasive arguments, built around the same premises stated before.

As future work we intend to make more use of the PAD mood space instead of the personality themes. Personality themes are of a great use in the beginning of the argumentation process but the mood space in the long term is better because it can add more information.

References

1. Marreiros, G., Ramos, C., Neves, J.: Dealing with Emotional Factors in Agent Based Ubiquitous Group Decision. Lecture Notes in Computer Science, 3823 pp. 41-50 (2005).
2. Davidsson, P.: Multi agent based simulation: beyond social simulation. Proc. of the Second international Workshop on Multi-Agent Based Simulation. S. Moss and P. Davidsson, Eds. Springer-Verlag NJ, pp. 97-107,(2001).

3. Jennings, N., Faratin, P., Lomuscio, A., Parson, S., Sierra, C. and Wooldridge, M.: Automated negotiation: Prospects, methods, and challenges. Journal of Group Decision and Negotiation, 2(10), pp. 199-215 (2001).
4. Zachary W., Ryder J.: Decision Support Systems: Integrating Decision Aiding and Decision Training. Handbook Of Human-Computer Interaction. Elsevier Science, chapter VII pp 1235–1258 (1997).
5. Marreiros, G., Santos, R., Ramos, C., Neves, J., Novais, P., Machado, J., Bulas-Cruz, J.: Ambient intelligence in emotion based ubiquitous decision making. In: Artificial Intelligence Techniques for Ambient Intelligence, IJCAI 07, Hyderabad, India (2007).
6. McRae, R. R., & Costa, P. T.: Validation of the five-factor model of personality across instruments and observers. Journal of Personality and Social Psychology, 52, 81-90 (1987).
7. Ortony, A.: On making believable emotional agents believable. In Trappl R. et al. (Ed.), Emotions in Humans and Artifacts (pp. 189--212). Cambridge, Massachusetts London, England: MIT Press (2003).
8. Mehrabian, A.: Analysis of the Big-Five Personality Factors in Terms of the PAD Temperament Model. Australian Journal of Psychology, Vol 48, no 2, pp 86-92 (1996).
9. John, O. P., & Srivastava, S.: The Big Five Trait Taxonomy: History, Measurement, and Theoretical Perspectives. In L. A. Pervin & O. P. John (Eds.), Handbook of personality: Theory and research (Vol. 2, pp. 102–138). New York: Guilford Press (1999).
10. Mehrabian, A.: Pleasure-Arousal-Dominance: A General Framework for Describing and Measuring Individual Differences in Temperament, *Current Psychology*, Winter, V14, N4, pp. 261-292 (1996).
11. Neves, J.: A Logic Interpreter to Handle Time and Negation in Logic Data Bases. In Proceedings of ACM'84, The Fifth Generation Challenge, pp. 50-54, (1984).
12. Ghasem-Aghaee, N., & Oren, T. I.: Effects of cognitive complexity in agent simulation: Basics. SCS'04, pp. 15-19 (2004).
13. Santos, R., Marreiros, G., Ramos, C. and Neves, J., Bulas- Cruz, J. Multi-agent Approach for Ubiquitous Group Decision Support Involving Emotions. LNCS, 4159, pp. 1174 – 1185 (2006).
14. Salancik, G., Pfeffer, J.: Who Gets Power – And how they hold on to it- A Strategic Contingency Model of Power. Organizational Dynamics, Vol. 5, pp. 3-21 (1977).
15. Marreiros, G., Santos, R., Ramos, C., Neves, J. and Bulas-Cruz, J.: ABS4GD: A Multi-agent System that Simulates Group Decision Processes Considering Emotional and Argumentative Aspects. AAAI Spring Symposium on Emotion, Personality and Social Behaviour, Stanford, pp 88-95 (2008).
16. Santos, R., Marreiros, G., Ramos, C. and Neves, J., Bulas- Cruz, J. Using Personality Types to Support Argumentation. Argumentation in Multi-Agent Systems (ArgMAS 2009).
17. Davidsson, P.: Multi agent based simulation: beyond social simulation. Proc. of the Second international Workshop on Multi-Agent Based Simulation. S. Moss and P. Davidsson, Eds. Springer-Verlag NJ, pp. 97-107 (2001).
18. Kraus, S., Sycara, K. and Evenchick, A.: Reaching agreements through argumentation: a logical model and implementation. Artificial Intelligence, 104(1-2), pp 1-69 (1998).
19. O'Keefe, D.: Persuasion: Theory and Research. SAGE Publications (1990).
20. Pruitt, D.: Negotiation Behavior. Academic Press, New York, N.Y. (1981).
21. Miller, C. E.: The social psychology effects of group decision rules. In P. B. Paulus (Ed.), Psychology of group influence (2nd ed., pp. 327-355). Hillsdale, NJ: Lawrence Erlbaum (1989).

An Emotion-Driven Interactive System

Francisco Ventura, António Pedro Oliveira and Amílcar Cardoso

Centre for Informatics and Systems, University of Coimbra, Portugal
{fventura, apsimoes}@student.dei.uc.pt, amilcar@dei.uc.pt

Abstract. Emotion-based interactive music systems have great application potential, namely in entertainment and healthcare. This paper describes an installation that explores the interactive capabilities of EDME, an emotion-based music engine we are developing. At the core of the installation there is an affective computer system that selects appropriate music and images to express its emotional state. The installation allows people to experience and influence the emotional behavior of this system. We made two experiments where people were able to ascribe emotions to the system in a natural way, relating arousal with the amount of movement and valence with the number of presences, as expected. The system transmitted expected emotions and gains from using both media (music and images) to express its emotional state.

Keywords: Interactive music system, automatic music production, emotions and music.

1 Introduction

Building interactive systems with the capability to produce music with an intentional emotional content involves multidisciplinary competences in Affective Computing [20], Human Computer Interaction, Music Psychology and a mixture of these domains [19]. The interaction may involve the perception of emotions from users [3, 8, 26]. This can be done with sensors to recognize psychophysiological signals, facial expressions, gestures and others [1, 7, 17]. The system must be able to adapt its emotional state to the perceived information and express it in some human recognizable way. The expression of its emotional state may be performed by suitably adapting media like music and images.

We intend to explore the capability of an affective system in the expression of a desired emotion to an environment [3, 8, 18, 25, 26], in such a way that people may recognize system emotional state in a natural way. We are developing a solution with a high degree of adaptability due to the flexibility of 2 auxiliary structures (music base and knowledge base) being used in EDME, an emotion-based music engine that intends to control the emotional expression of produced music, that makes the system a good choice for almost every context. This paper starts with a review of systems related to our work (section 2). Then, we describe our emotion-based interactive installation (section 4) that intends to interact with people by using music and images. This installation intends to explore the interactive capabilities of EDME (section 3).

The installation allows people to experience and influence the emotional behavior of the computer system. The description of the two experiments made to test this system is presented in section 5. Section 6 analysis results from the second experiment and, finally, section 7 draws some conclusions.

2 Related work

The emotional interaction with people by using different media is a promising field of research. The importance of developing systems with such a capability is evident to the society. They can be used in contexts where there is a need to create environments capable of inducing certain emotional experiences: intelligent spaces [26], virtual environments [3, 15, 21], healthcare spaces [16], dance spaces [25], wearable devices [5, 8, 9, 17, 18, 22, 24, 27], etc. They can also be applied in the production of soundtracks for computer games [6, 10, 11]. The objectives underlying these contexts can be various: reflect the mood of game characters in the output of the system [6]; build recreational and therapeutic devices for people with neurological and physical disabilities [16, 25]; improve exercise performance [5, 17, 18, 22, 27]; adapt system output to a desired emotional state [1, 4, 7, 9, 10, 11, 12, 14, 15, 17, 21]; or, in a broad sense, dynamically express a desired emotion of/to an environment [3, 8, 18, 25, 26].

These systems are usually composed by a perception module that perceives the emotion from the environment by using data coming from different sources: psychophysiological sensors [1, 4, 5, 8, 9, 12, 15, 16, 17, 18, 27], cameras [3, 7, 13, 14, 17, 25], microphones [26], pressure sensors [26], temperature sensors [8], motion sensor [8, 13, 22, 25, 27], pollution sensors [8], physical data [4, 24] and self-report data [4, 9, 27]; and by an action module that reacts to the environment possibly by using different media (e.g., music and images). When using music to react to the environment, these systems usually make use of a composition/selection algorithm to generate music driven by emotional controls that intend to influence the emotional state of the user(s) of the environment. The algorithms used to generate music [3, 10, 11, 12, 21, 25, 26] deal with mapping between music and emotional data obtained with the referred sources. Musical features of the generated music reflect the intended emotional variations of the user(s).

Ada is a system especially relevant in this area [26]. It is an intelligent space that communicates moods, emotions and behaviors in real-time by using sound and light. Moods are defined by arousal and valence variables. Like others, Ada communicates by composing music to express the moods and emotions. Arousal variable changes tempo, volume and octave register. Valence variable changes consonance and pitch material. It is interesting the possibility to extend some of these systems to be used in a network of integrated music controllers [13]. These controllers create a collaborative interface between emotions and music generation.

3 EDME - Emotion-Based Music Engine

EDME is a system that produces music expressing a desired emotion. This objective is accomplished in 3 main stages: segmentation, selection and transformation; and 3 secondary stages: features extraction, sequencing and synthesis. We are using 2 auxiliary structures: a music base and a knowledge base. The music base has pre-composed MIDI music tagged with music features. It is prepared offline for MIDIs of any musical style. The knowledge base is implemented as 2 regression models that consist of relations between each emotional dimension (valence and arousal) and music features. It can be adapted to the emotional feedback given by listeners (via questionnaires or psychophysiological signals). The characteristics of these auxiliary structures make the system easily adaptable to every context.

Aided by Figure 1 we will describe with more detail each of these stages. Pre-composed music of the music base is input to a segmentation module that produces fragments. These fragments must as much as possible be musically self-contained and express a single emotion. Segmentation discovers fragments by looking for note onsets with the higher weights. These weights are attributed according to the importance and degree of variation of five features: pitch, rhythm, silence, loudness and instrumentation. The features extraction module then obtains music features that are used to emotionally label the fragments with the help of the knowledge base, which are then stored in the music base.

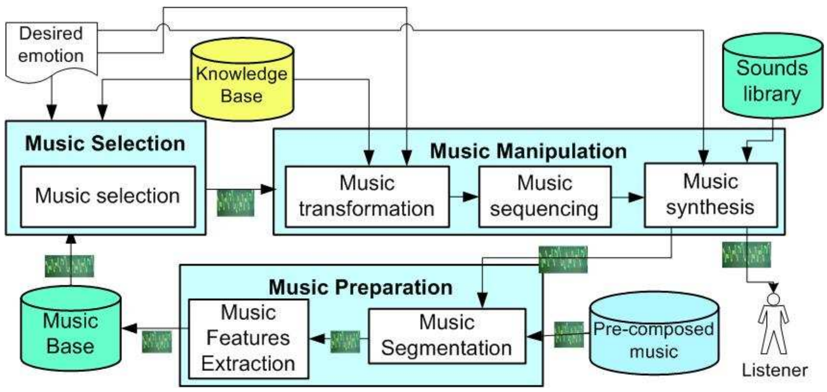


Fig. 1. Diagram portraying the 3 main stages (segmentation, selection and transformation) and 3 secondary stages (features extraction, sequencing and synthesis) of EDME.

The selection module obtains musical pieces with an emotional content similar to the desired emotion. These pieces are obtained from the music base, according to similarity metrics between desired emotion and music emotional content. Selected pieces can then be transformed to come even closer to the desired emotion. Transformation, also supported by the knowledge base, is applied in different characteristics of 5 groups of features: rhythm, melody, harmony, instrumentation and dynamics. The knowledge base has weights that control the degree of transformation for each characteristic. Sequencing module puts pieces produced by the transformation module in a smooth sequence by changing appropriate musical features. This sequence is given to a synthesis module that uses information about the General MIDI (GM) instruments [23] and timbral features to guide the selection of sounds from a library of sounds. This library is composed by various samples available from sample libraries of GM instruments (e.g., [28, 29]).

4 Emotion-Based Interactive Installation

We developed an installation to provide an experimental context for assessing the interaction capabilities of EDME. We integrated the music engine in a multi-agent system built over JADE [2] that controls the overall behavior of the machinery. A computer connected to a camera, a projector, and speakers compose the setup (Figure 2).

The camera is placed on the ceiling of room. The interaction area representing by the grey circle is constrained by the field of view of the camera. One of the walls is a wide translucent screen where images are displayed under the computer's command.

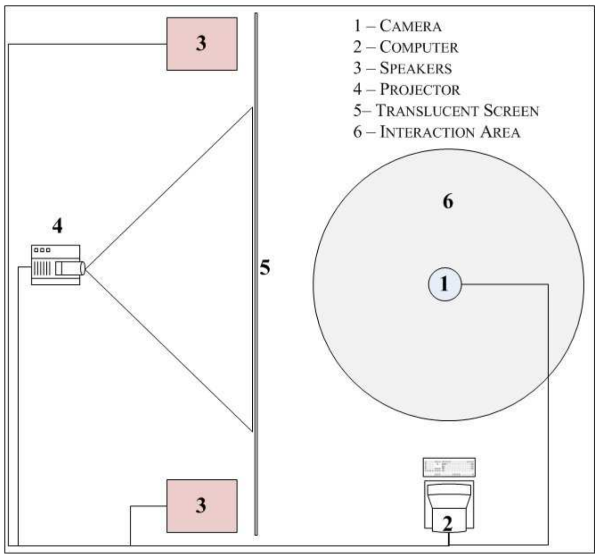


Fig. 2. Plan of the experimental setup.

The installation adapts the expressed emotions to the presence of people in the interaction area: the system expresses positive valence emotions on the presence of people and negative valence emotions when left alone. Moreover, people movement induces an increase in the arousal, whilst lack of activity induces a decrease. The system collects data from the environment through the camera and expresses emotions by means of the music it plays and the images it projects.

At the beginning of each trial session, the camera takes a picture of the environment. This first picture is considered to be the model. This image will represent the scenario of the experiment. As each session lasts for short time intervals, where the daylight and other environmental changes are not important, we assume that the scenario stays immutable during each session. Of course, if we needed to take a longer experience the light and environment changes could be prevented by a re-adaptation of the initial environment. Although the system prevents this, it is disabled for our experiences, since these experiences were made in closed rooms with artificial and constant light.

We are considering two environment variables: level of presence and movement. The system takes a new picture every second to update these variables. The level of presence is determined by computing the percentage of image change between the new picture and the scenario. This means that two persons induce a higher presence level than one. Moreover, the level is not dependent of people's movement, as long as they stay in the field of view of the camera. Movement is calculated by comparing two adjacent pictures. Using this method, only instant movement is calculated. As long as there is movement in the radius, the arousal increases, and when movement stops, arousal starts to decrease.

The environment information is then converted in a *stimulus* that is used to define the emotion that the system will express. Increase in the level of presence (e.g., one person enters the interaction area) produces a positive stimulus for valence; the reverse applies for decrease in the level of presence. As for movement, as long as there is activity in the interaction area, the stimulus for arousal increases. When movement stops, arousal starts to decrease. Arousal increase and decrease are linear with time.

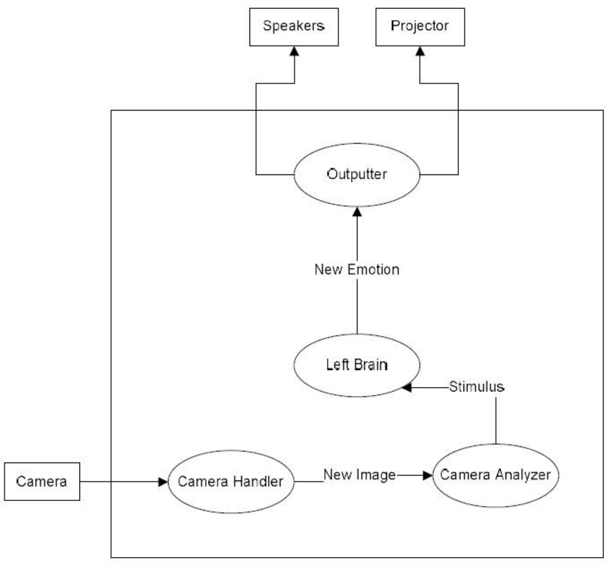


Fig. 3. Multi-agent system architecture.

The architecture of our multi-agent system is represented in Figure 3. An agent *Camera Handler* is in charge of gathering environment data. At the beginning of each trial session, the camera takes a picture of the environment. This image will represent the *scenario* of the experiment. As each session lasts for short time intervals, we assume invariance of daylight and other environmental conditions and, consequently, that the scenario stays immutable during the experiment.

An agent *Camera Analyser* converts the environment information in a *stimulus* that is used by an agent *Left Brain* to synthesise the new emotional state of the system according to the current emotional state and the received stimulus. Increase in the level of presence (e.g., one person enters the interaction area) produces a positive stimulus for valence; the reverse applies for decrease in the level of presence. As for movement, the stimulus for arousal increases as long as there is activity in the interaction area. When movement stops, arousal starts to decrease. Arousal changes are linear with time. Emotional states are synthesised with continuous change.

The new synthesised emotion is sent to the *Outputter* agent, who wraps EDME and is in charge of choosing the music to play and the image to project. The aim of the images is to reinforce the interaction experience provided by the installation. Images are chosen based in three levels of valence, sad, neutral and happy, and don't show arousal levels.

5 Experiments

We made two experiments with our system. The first one was a preliminary one that had the objective of observing the interaction between the system and participants (30 people); the second one was focused in obtaining participants feedback (via a questionnaire) about various components of the behavior of system. Figure 4 shows the two steps that composed both experiments. The first step consists in the perception of data from the environment: number of participants and quantity of movement. This data is used to define the emotional state of our system: valence (degree of happiness) and arousal (degree of activation). For instance, if it has the values 0 for arousal and 0 for valence, the system is very sad, if it has the values 10 for arousal and 10 for valence, the system is very happy and excited. The number of participants influence valence; the quantity of movement influence the arousal. The system's emotional state influences the way the system expresses to the environment by selecting different music and images to portray to the audience.

Participants interact with the system through sensors that detect presence and movement. At the core of the experiment, there is a computer that produces ambience music and projects images in a screen. Based on affective computing techniques, the computational system reflects in the musical features the variation of the number of presences and quantity of movement: for instance, produced music and images are happier with the presence of more people on the room; or can be more active when more movements are detected.

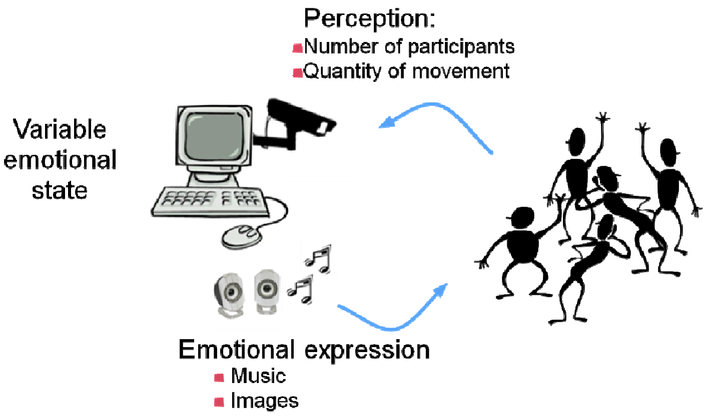


Fig. 4. Diagram illustrating the 2 steps of the two experiments: emotional perception and expression.

The first experiment was made in a session called "Discovering the Faculty on Saturdays". This session was attended by 30 students from secondary schools with ages from 10 to 18 years old. Figure 5 shows the environment with some participants.

Fig. 5. Picture taken during the first experiment.

In the second experiment we had groups of, approximately, 4 people interacting with the system in sessions of approximately 10 minutes. In the end of each session participants were asked to answer to a questionnaire with 7 questions (Table 1) with the objective to use obtained data/answers to evaluate the efficacy of the system. Each question was answered with values from the integer interval between 0 and 10. Sessions were attended by 23 people from the university (students and professors) with ages from 18 to 60 years old. Each session was conducted in three phases: first, the emotional state of the system was expressed by using only music; then, only images; and finally both media. This allowed us to evaluate the effect of each medium in emotional communication.

Table 1. 7 questions of the questionnaire given to the participants.

1)	The system expressed happiness with many presences and sadness with few presences
2)	The system expressed activation with much movement and relaxation with the lack of movement
3)	What is the importance of music in the emotional expression of the system
4)	What is the importance of images in the emotional expression of the system
5)	Music expressed expected emotion
6)	Images expressed expected emotion
7)	Efficacy of the system in the expression of the expected emotions

Both experiments were preceded by a series of preparatory steps: manual emotional classification of images and offline synthesis of music (Figure 6). Manual emotional classification consisted in attributing three levels of valence (sad, neutral, happy) based on personal background on this topic. Offline synthesis of music consisted in: segmentation of pre-composed music; automatically extraction of features from obtained segments; and automatic selection of music expressing different emotions. This music was then given to a synthesis module that converted selected MIDI music into mp3 audio music.

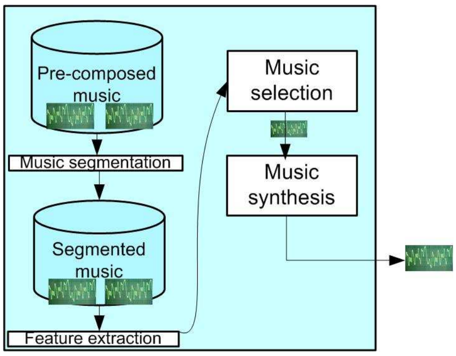


Fig. 6. Diagram illustrating the processes involved in the offline synthesis of music.

6 Results

The verbal feedback given by the participants of the first experiment was positive: the system interacted in an expected emotional way. This section is devoted to the analysis of the answers obtained with the questionnaire (Table 1) given to the 23 participants of the second experiment. Figure 7 presents the mean and standard deviation for the 7 questions of the questionnaire. We obtained high standard deviations when analyzing the importance of the image in the system and in the analysis of the expression of the expected emotion with the image. After the analysis of all the answers we came to the conclusion that, generally, the system correctly related arousal with the amount of movement and valence with the number of presences. Despite the fact that music seems a medium more important than images to express emotions, it was less successful than images in expressing the desired emotion. In general, the system was efficient in the transmission of the expected emotions. These conclusions give a first clue about the behavior of the system; however their significance is limited by the low number of participants (23), as well as, by the presence of some questions with multiple components (e.g., first and second questions).

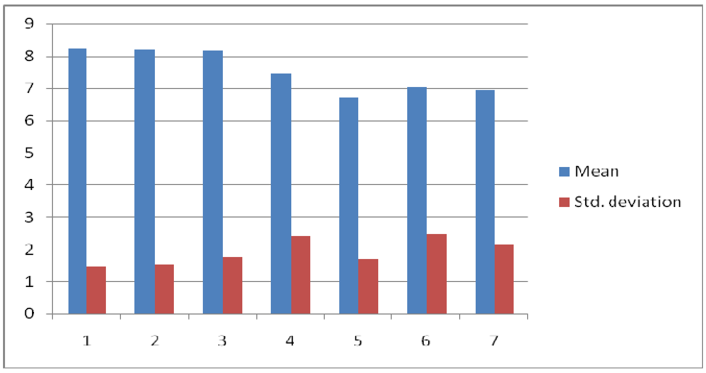


Fig. 7. Mean and standard deviations for the answers of the questionnaire.

Table 2. Correlation coefficients between the answers of the questionnaire (Table 1).

Answers	1)	2)	3)	4)	5)	6)	7)
1)	-	46	39	42	54	31	50
2)	46	-	29	45	56	44	**71**
3)	39	29	-	**82**	61	**67**	59
4)	42	45	82	-	55	**68**	60
5)	54	56	61	55	-	**71**	**76**
6)	31	44	67	68	71	-	**78**
7)	50	71	59	60	76	78	-
Mean	44	49	56	59	62	60	66

We also analyzed the correlation coefficients between the answers (Table 2) and concluded that the expression of the arousal of the system, the expression of expected emotions with music and images (answers 2, 5 and 6) has a high degree of correlation with the efficacy demonstrated by the system (answer 7). This leads us to have a special attention devoted in the analysis of the emotional effect of music and images, namely in the arousal dimension. The importance of music and images in the emotional expression and the expression of expected emotion in music (answers 3, 4 and 5) are in some way correlated with expression of expected emotion in images (answer 6). The importance of the emotional expression with music (answer 3) is highly correlated with the importance of the emotional expression with images (answer 4), which seems to reinforce the importance of using both media to express the emotional state of the system.

7 Conclusions

We prepared an installation that allows participants to experience and influence the emotional behavior of an interactive computer system. The experiences conducted show that people ascribe emotions to the system in a natural way. Results of the second experiment show the importance of both media (music and images) in the expression of the emotional state of the system; and the importance of analysing the emotional effect of music and images, namely in the arousal dimension. Our system can be, hopefully, more efficient by improving the EDME engine; by making an engine similar to EDME for images; and by complementing our work with studies of psychology.

We also obtained several suggestions from the participants to improve the system: synchronize music and images; use more emotional states for images; generate music and images in real-time; improve selection of music and images; decrease the delay of adaption of music to the environment; improve transitions among music; and have more sensors for movement and presences.

References

1. Arslan, B., Brouse, A., Simon, C., Lehembre, R., Castet, J., Filatriau, J. and Noirhomme, Q.: A real time music synthesis environment driven with biological signals. International Conference on Acoustics, Speech, and Signal Processing (2006).
2. Bellifemine, F., Poggi, A. and Rimassa, G.: JADE – A FIPA-compliant agent framework. Proceedings of PAAM, vol. 99, pp. 97-108 (1999).
3. Casella, P. and Paiva, A.: Magenta: An architecture for real time automatic composition of background music. International Workshop on Intelligent Virtual Agents. Springer, pp. 224–232 (2001).
4. Chung, J. and Vercoe, G.: The affective remixer: Personalized music arranging. Conference on Human Factors in Computing Systems. ACM Press New York, pp. 393–398 (2006).
5. Dornbush, S., Fisher, K., McKay, K., Prikhodko, A. and Segall, Z.: Xpod: A human activity and emotion aware mobile music player. International Conference on Mobile Technology, Applications and Systems (2005).
6. Eladhari, M., Nieuwdorp, R. and Fridenfalk, M.: The soundtrack of your mind: Mind music-adaptive audio for game characters. International conference on advances in computer entertainment technology. ACM Press NY (2006).
7. Funk, M., Kuwabara, K. and Lyons, M.: Sonification of facial actions for musical expression. New Interfaces for Musical Expression (NIME). National University of Singapore, pp. 127–131 (2005).

8. Gaye, L., Mazé, R. and Holmquist, L.: Sonic city: The urban environment as a musical interface. New Interfaces For Musical Expression (NIME). National University of Singapore, pp. 109–115 (2003).
9. Healey, J., Picard, R. and Dabek, F.: A new affect-perceiving interface and its application to personalized music selection. Workshop Perceptual User Interfaces (1998).
10. Hoeberechts, M., Demopoulos, R.J. and Katchabaw, M.: A Flexible Music Composition Engine. Audio Mostly 2007: The Second Conference on Interaction with Sound (2007).
11. Ka-Hing, J., Sze-Tsan, C., Kwok-Fung, C. and Chi-Fai, H.: Emotion-driven automatic music arrangement. International Conference on Computer Graphics and Interactive Techniques. ACM Press NY (2006).
12. Kim, S. and André, E.: Composing affective music with a generate and sense approach. Proceedings of Flairs 2004 - Special Track on AI and Music. AAAI Press (2004).
13. Knapp, R. and Cook, P.: Creating a network of integral music controllers. New Interfaces For Musical Expression, pp. 124–128 (2006).
14. Lyons, M. and Tetsutani, N.: Facing the music: a facial action controlled musical interface. Conference on Human Factors in Computing Systems. ACM Press NY, pp. 309–310 (2001).
15. McCaig, G. and Fels, S.: Playing on heart-strings: experiences with the 2hearts system. New interfaces for musical expression, pp. 1–6 (2002).
16. Miranda, E. and Boskamp, B.: Steering generative rules with the eeg: An approach to brain-computer music interfacing. Sound and Music Computing (2005).
17. Nakra, T.: Inside the Conductors Jacket: Analysis, Interpretation and Musical Synthesis of Expressive Gesture. PhD thesis, Massachusetts Institute of Technology (1999).
18. Oliver, N. and Flores-Mangas, F.: Mptrain: a mobile, music and physiology-based personal trainer. Conference On Human-Computer Interaction With Mobile Devices And Services. ACM Press NY, vol. 8, pp. 21–28 (2006).
19. Pantic, M., Sebe, N., Cohn, J. and Huang, T.: Affective multimodal human-computer interaction. ACM International Conference on Multimedia. ACM NY, pp. 669-676 (2005)
20. Picard, R.: Affective Computing. MIT Press Cambridge (1997).
21. Robertson, J., De Quincey, A., Stapleford, T. and Wiggins, G.: Real-time music generation for a virtual environment. Workshop on AI/Alife and Entertainment (1998).
22. Sakata, N., Kurata, T., Kourogi, M. and Kuzuoka, H.: Situated music: An application to interactive jogging. International Symposium on Wearable Computers (2006).
23. Selfridge-Field, E.: Beyond MIDI: the handbook of musical codes. MIT Press (1997).
24. Schiemer, G. and Havryliv, M.: Wearable firmware: The singing jacket. Proceedings of ACMA, pp. 66–71 (2004).
25. Wallis, I., Ingalls, T. and Campana, E.: Computer-Generating Emotional Music: The Design of An Affective Music Algorithm. International Conference on Digital Audio Effects (2008).
26. Wassermann, K., Eng, K., Verschure, P and Manzolli, J.: Live soundscape composition based on synthetic emotions. IEEE Multimedia, vol. 10, pp. 82-90 (2003).
27. Wijnalda, G., Pauws, S., Vignoli, F. and Stuckenschmidt, H.: A personalized music system for motivation in sport performance. IEEE Pervasive Computing, vol. 4, pp. 26–32 (2005).

28. Project SAM Symphobia: (accessed on July 14th 2009) http://www.projectsam.com/Products/Symphobia/
29. Garritan Personal Orchestra: (accessed on July 14th 2009) http://www.garritan.com/GPO-features.html

Chapter 6

GAI - General Artificial Intelligence

An Intelligent Interface Agent for an Airline Company Web Portal

Luís Barbosa[1], António J.M. Castro[1], Luís Paulo Reis[1]

[1] LIACC-NIAD&R, FEUP, Department of Informatics Engineering, University of Porto, 4200-465 Porto, Portugal
{lbarbosa, ajmc, lpreis}@fe.up.pt

Abstract. In this paper we introduce an implementation of an Intelligent Interface Agent to support the use of a web portal in an airline company. The interface agent architecture and data model is presented. We formalized concepts such as relevance and proximity regarding the data structure. The concepts of personal opinion and general opinion are also introduced and formalized. A statistical analysis was performed to obtain the best value when processing the general opinion. Some results of that analysis are presented and we conclude discussing our work and presenting future improvements.

Keywords: Intelligent Interface Agent, User Interface Agent, User Modeling.

1 Introduction

TAP Portugal - Portal DOV [1] is currently a hub of information and services to all crewmembers. It centralizes information and content drawn from diverse sources enabling users to have access to information and services previously accessed at the TAP installations or by telephone. For instance, now one crewmember can view his schedule of work for the day or next week by simply logging to the Portal at his home or in any place with an Internet connection. Alternatively, if a service or department requires a new Training Manual to be made available for the pilots, it can resort to Portal DOV (Web Portal) to publish it.

The Web Portal grants the user a greater mobility and independence, but it has some limitations. As information and documentation are regularly changed, so the user must be up to date about the significant changes. For instance, a crewmember must take an active attitude to check up his roster for relevant changes, even though they may not occur. The process of returning to the Web Portal only to check up the current state of information can turn out a tedious or monotonous task. This sort of tasks are the competence of interface agents, whose purpose is to facilitate the users work by doing boring or repetitive tasks, freeing the user to use the freed time more efficiently. The Web Portal interface agent allows the user to define tasks about the surveillance of information sources by different levels of priority. An example would be a flight attendant that defines a high-level alert task to monitor changes concerning the days off in her roster and very-low-level alert task associated with the publication of new internal communications. With the help of the agent she could be more promptly acquainted about changes in the current state of information seen as important to her. The Web Portal relying on interface agents to represent crewmembers could result in two major gains: (i) Even more mobility and independence to the users, as the need to access Web Portal to verify the current information diminishes and (ii) increase in the awareness about crucial information for each professional category by making suggestions to the crewmember. Our Interface Agent has two main goals: (1) allow the user to define properly tasks set to monitor changes in relevant content and ensure the user is properly informed and, (2) make suggestions perceptive to the user about new tasks.

Although there will be an interface agent representing each crewmember and appointed to the vigilance of information deemed important by the owner, it does not require direct interaction with the existent agents. The agent will operate blind to the existence of other agents. The suggestions are based on the general opinion of the users with the same profile in regard to what is important to them. An example would be the case where a flight attendant on vacation gets a suggestion to create a new very-high-level alert task on the Web Portal communications. This occurred because in her absence happened major changes in the access to Web Portal and the next announcement would be extremely important to all **crewmembers** access it properly. This example illustrates one way in how the alert task and suggestions can help the crewmembers in keeping up with what is important to the professional category of crewmembers at a time.

The rest of the paper is organized as follows. In section 2 we present some work of other authors regarding intelligent interface agents. Section 3 describes the interface agent architecture including the data model and the relevance and proximity concepts. The interface agent was designed following the methodologies and guidelines presented in [2]. Section 4 shows how we process the relevance, including the definition of the concepts of Personal and General Opinion. In section 5 we present the General Opinion Processing and the statistical analysis to obtain the best value when processing the general opinion. Section 6 shows the results of the statistical analysis. Finally, we conclude our work in section 7.

2 Related Work

An Intelligent Interface Agent (IIA) cooperates with the user in performing its tasks, working as a *personal user assistant*. The agent is pro-active taking the initiative and not passive. Usually, an IIA is not the interface between the user and the application. Instead, it observes the interaction between the user and the program learns with it and interacts both with the program and the user. *UCEgo* [7] is a Unix consultant that, using natural language and modeling user goals and plans, takes the initiative in offering the user information about certain Unix concepts or commands, correcting, at the same time, any misconceptions. *MOKSAF* [5][6][8] is an example of a multi-agent system for user interfaces. The MOKSAF environment is a system for military mission planning and designed to explore teamwork within heterogeneous human/agent teams. In [4] we can find an example of an IIA applied to an eye disorder diagnosis system. This IIA is able to reason on user actions and provide assistance on the target application. The personal news agent presented in [12], adapts to the user's preferences and interests based on voice feedback from the user. An interesting use of an IIA is presented in [13]. The *QueryGuesser* agent personalizes users' interaction with databases assisting users of computer applications in which retrieving information from a database is a key task.

The IIA uses different machine learning techniques to build user profiles. For example, in [9] the *PersonalSearcher* agent uses hierarchical clustering to obtain different web topics that a user might be interested. The *NewsAgent* presented in [10] uses case-based reasoning and a topic hierarchy to discover which newspaper news a user prefers. Decision trees are applied in CAP [11] to learn users' scheduling preferences.

3 The Interface Agent Architecture and Data Model

Our intelligent interface agent architecture is composed of just one agent as depicted in Figure 1. The operations inside the box represent the roles performed by the agent. The *AlertPreferencesSet* role allows the crewmember to define the behavior of the agent and the *UserAlert* role allows the interface agent to show the information to the crewmember. The *DataSourceMonitor* and *InformationDifferenceView* roles are performed automatically by the interface agent, taking into consideration the preferences of the user and the information that exists on the web portal. On the environment we have information resources that the interface agent can use. The *User Preferences* and *Information Reference* are resources that the interface agent is able to read and write on it. The *Roster* and *Documents* are resources of the web portal and the interface agent is able to read them remotely through a web service. All these resources are part of the interface agent data model, described on the next sections.

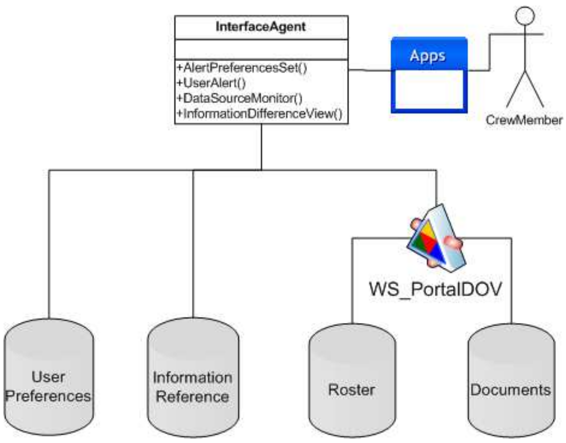


Fig. 1. Interface Agent Architecture

The Interface Agent Data Model is composed of four distinct concepts. First we have the *Data Source* concept, which characterizes the information the agent deals with when searching for new events. The second, *Agent Task*, characterizes the tasks submitted by the user. The third, *CrewMember*, provides the grounds for a practical and proper crewmember depiction. The fourth, *Data Relation*, is oriented to the description of the relations between the previous concepts that compose the model and how they can enable the learning of user behavior. To design the model the technique adopted was the creation of a relational database with the appliance of normalization.

3.1. Data source concept: The data sources are distinct pools of data or information, with a specific purpose and behavior, used by the Web Portal to provide information and the services required by the crewmembers to execute their responsibilities. Every data source has its own entities, or abstract units that help to organize or sort out the information. The entities are themselves composed of a collection of elements. In our work we examined two data sources, the *Roster* and *Documentation*. The *Roster* comprises a set of activities a crewmember will have to execute during a period of time. The crewmember roster is usually planned with a 30-day advance. As activities of crewmember we have for example *Pairing*, *Ground Duty* or *Day Off* to name a few. The *Documentation* encompasses all the available documentation in the Web Portal. A pair of elements defines documentation: *documentation type* and *publisher*. *Documentation type* informs the category to which the document belongs to in the internal organization. It can be in the form of *Internal Communication* and *Schedules* for instance. The *publisher* represents the entity or service responsible by the release of different documents found in the database. In relation to time, the documentation is periodically updated with new documents of different types submitted by the various publishers. The update can take one form of three actions: addition, change and removal of a document.

3.2. Agent task concept: The agent task is the central structure. It represents the user's fundamental requirement, or the need to be updated about events in the data sources. This need is expressed by conjugation of **four** concepts: *who, what and where, when and how* and *importance*.

- *Who:* is directly related to crewmember, the owner of the agent, responsible for the creation of tasks to be performed by the agent. Crewmember has its own data structures designed to support the appropriated representation.
- *What and where:* represents the data sources and the selection of inherent elements chosen by the user to be watched over for new events. The data sources are *Roster*, composed of a schedule of activities, and *Documentation*, the collection of types of documents submitted by various publishers.
- *When and how:* represents the task active period. The period of activity is directly related with how or recurrence type. The recurrence type defines the *modus operandi* of the agent in relation to time. There are four types of recurrence, namely: *Never*, *Number of times*, *Until* and *Forever*.
- *Importance:* show how relevant is the task to the crew member.

3.3. Crewmember concept: Represents the application user. It is organized in three views:

- *Application User*: Information required to user access and interaction.
- *Basic Personal Information*: Exhibits personal information about the user.

- *Professional Category*: Represents information about the user in terms of position (or rank) in a flight. It includes information about the crewmember such as: pool (type of aircraft the crewmember flies, for example, narrow body, wide body, A320 fleet, etc.), crew group (flight crew or cabin crew) and rank (the place in the chain of command within a crew group, for example, captain, cabin supervisor, etc.). This view is a fundamental aspect of the extraction of information about crewmember attitude toward the data sources. It is the prime aspect in the crewmember characterization to the judgment of the information present in the web portal, promoting the adequate suggestions.

3.4. Data relation: Data relation is the fundamental path to attain knowledge about the considerations of the crewmember and the constituent professional classes in regard to the data sources when creating the tasks. The *Data Relation* provides the interface agent with the means to extract suggestions to the crewmember about new tasks. The knowledge is captured by the conjugation of three concepts: relevance, proximity and view. These three concepts serve as a function to transmit the weight of a relationship between elements of a data source and a view derived from the professional category. The relationship weight function has the generic form:

$$Weight = F(Elem(Type(k, ds), v)) \qquad (1)$$

F – is the function applied to extract the weight of the data source element (either the relevance or the proximity function, both defined below).
Type(k, ds) – function to return an entity *k* part of the data source *ds* structure.
Elem(Type(k, ds)) – is the function that returns an element from the data source *d*s. The variable *ds* can be one of two values: *Roster* or *Documentation*.
v – the view represents the professional category chosen for interpretation of the element data source. At the foundation of the professional category is the concept rank.

3.4.1. Relevance: Relevance, *or how significantly the data source element is employed*, is the function revealing the importance one element present in a data source has to a professional category. The relevance function (mentioned in Equation 1), is:

$$Relevance = R(Elem(Type(k, ds), i), v) \qquad (2)$$

Type(k, ds) – the same as in equation 1.
Elem(Type(k, ds), i, v) – is the function that returns the element *i* from the entity present *k* in data source *ds*. The variable *ds* can be one of two values: *Roster* or *Documentation*.
v – the same as in equation 1.

3.4.2. Proximity: Proximity, *or how strongly each element from the data source is associated to another when defining tasks*, informs how frequently each element present in a data source relates to another by the professional category standpoint. To calculate the weight from the relationship the function *Proximity* requires two different elements from the same data source. The proximity function (mentioned in Equation 1) is:

$$Proximity = P(Elem(Type(y,ds)\ i,v), Elem(Type(z,ds),j),v), if(y=z) \longrightarrow i \neq j \quad (3)$$

Elem(Type(y, ds), i, v) **–** is the function that returns the element *i* from the entity *y* present in data source *ds*. The variable *ds* can be one of two values: *Roster* or *Documentation*. When addressing the same entity (*y*=*z*), variables *i* and *j* must be different;

3.4.3. Data relation structure: To capture the relationships between the data source elements and the professional categories, a structure capable of supporting the functions of *relevance* and *proximity* in each data source was defined. The central component in the data relations' structure would be the professional category viewpoint used to evaluate each data source element, namely *pool*, *crew group* or *rank*. The professional category chosen to define the viewpoint between the three was *rank*. The reason to the choice made was expediency, since through the rank we can also establish the relevance and proximity based on the crew group viewpoint with reduced effort. The number of tables needed to express relevance and proximity in each data source by a viewpoint is proportional to the number of elements types present in the same data source. For *relevance* the number of tables is given by *n*, where *n* is the number of entities present in the data source. For *proximity* the tables must express the closeness in each entity and between pairs of entities. So the total number of tables follows the progression:

$$T_{tables} = n + \frac{n!}{2(n-2)!} \quad (4)$$

In the worst-case scenario the total number of tables necessary to form Data Relations structure is:

$$nViews * \sum_{i=1}^{nDatasources} 2 * nElemTypes(i) + \frac{nElemTypes(i)!}{2(nElemTypes(i)-2)!} \quad (5)$$

nViews – number of viewpoints to judge the entities.

nDatasources – number of data sources.

nElemTypes(i) – number of element types the exist in data source *i*.

For example in our case, with one viewpoint and two data sources, by application of the formula the total number of tables would be 1*(2+2*2+1) = 7.

4 Relevance Processing

Overview: Relevance Processing consists in revising the relevance of a data source element when a new task is submitted by a crewmember. The revision is directly aligned to the view derived from the crewmember professional category. Whenever a new task is created, a relevance value is given directly by the crewmember. The relevance has the goal of establishing how significant the task is to the user and to define the agent behavior to act accordingly. **Hence this relevance defines the importance of a set of elements of a data source at the moment to the user.**

Before submission, the task is processed to accommodate the user behavior. The user behavior during the interaction may convey implicit value pointing the agent to define more accurately the final relevance. Finally, when the task is accepted, the *Data Relations* structure, which expresses how the elements relate to the users and between themselves, must be updated to include the new relevance and so evolve as the crewmembers attitude about the same elements changes. How the elements relate to each other by a professional category view is the support to elaborate suggestions to the crewmember about new tasks. For the sake of understanding lets adopt an alternative naming convention, i.e. the idea of "*personal opinion*" and "*general opinion*".

Personal opinion definition: The relevance deriving from the user interaction as the crewmember *"personal opinion"* at the moment in regard to a set of elements is:

$$Op = R(Elem(Type(k, ds), i), ProfView(cm), Crewmember), i \longrightarrow 1 \ldots total \quad \textbf{(6)}$$

ProfView(cm) – function which extracts the view representing the professional category from the crewmember *cm*.
Elem(Type(k, ds), i, ProfView(cm)) – function that returns the element *i* from the entity present *k* in data source *ds*. The variable *ds* can be one of two values: *Roster* or *Documentation*.

General opinion definition: The value of the relevance found in *Data Relations*, for the same set of elements viewed by a professional category to which the crewmember belongs, is expressed as:

$$Og = R(Elem(Type(k, ds), i), ProfView(cm), Data\ Relations), i \longrightarrow 1 \ldots total \quad (7)$$

5 General Opinion Processing and Statistical Analysis

Overview: Because it was opted to keep <u>only one general opinion</u> at a time, the method defined to deliver a new general opinion would be inspired by the exponential moving average. The key issue in the process of generating the new general opinion would be the smoothing factor α applied in the exponential moving average equation (see equation 9). The statistical analyses procedure adopted follows the five steps method mentioned in *Estatística* [3], that is, (i) goal setting and conditions, (ii) definition of data gathering process, (iii) data gathering, (iv) data analysis and (v) ascertainment of conclusions about the population acquired in the previous step.

Goal settings and conditions: The main purpose of this study was to select the most acceptable method by comparing their behavior in a simulation with different smoothing factors to the average personal opinion. Another interesting condition to observe was how the very short memory term would affect the extraction of general opinions. All the methods were subordinated to the structure imposed by the *Data Relations* and so only one record of the general opinion would be stored at a time. This factor, although limiting, could be dealt without difficulty given the following conditions were met:

– Small classification range of the opinions, to limit possible irreconcilable deviations. In this case the range would be in a fairly manageable five qualities scale: V*ery low*, *Low*, *Normal*, *High*, *Very High*;
– General opinions included all the previous past general opinions and personal opinions feedback. This condition is mandatory and is inherent to all the methods devised;
– Personal opinions would not deviate too much when addressing opinions to the same elements. Given the relative small scale, only dealing with extreme opinions would be of concern.

The ability to react to greater and punctual deviations is dependent in two interconnected factors:

– The structure of the data used to capture the general opinions;
– The smoothing value chose for calculation of new general opinions with new personal opinions.

While the personal opinions are based on five level strata according to their importance, all the intermediate calculations about the general opinion can work with different precisions. When needed a suggestion the general opinion obtained would be converted by rounding to a discrete value in the same five level scale, as indicated in equation 8.

$$Sug_i = round(Og_i) \quad \textbf{(8)}$$

The ability to translate later the data to a proper discrete group enables the future general opinion to be more sensitive and consistent with the evolution of the opinions given by the crewmembers. By converting the general opinion to a suggestion via rounding allows the five groupings to have the same dimension and so ensure that bias toward a discrete value is avoided. The exponential moving average is presented in equation 9.

$$EMA_t = EMA_{t-1} + \alpha(Price - EMA_{t-1}) \quad \textbf{(9)}$$

EMA_t – new exponential moving average value;
EMA_{t-1}– previous exponential moving average value;
Price – current value given;
α – percent-based smoothing factor, $0 \leq \alpha \leq 1$

Equation 10 shows the ensuing formula, which drove the calculation of new general opinions. The smoothing α determines how the general opinion should shift toward the new opinions.

$$Og_i = Og_{i-1} + \alpha(Op_i - Og_{i-1}), \alpha \exists]0,1] \quad \textbf{(10)}$$

Og_i – new general opinion value;
Og_{i-1} – previous general value;
Op_{i-1} – submitted personal opinion;

The main purpose was to directly observe the behavior of the method with different values $\alpha=0,5$, $\alpha=0,33$ and $\alpha=0,66$ to select the most close to the average personal opinion in a sequence of personal opinions submission. Because only one record is stored, the general opinion must be recalculated every iteration for the new personal opinion.

Data gathering process: The data gathering process would have as basis the simulation of *personal opinion* submission for the same entity. Every simulation was established to evaluate the behavior of the methods in regard to two outcomes: how the *general opinion* and the *suggestion* made would respond after a *personal opinion* was submitted. The data gathering process consisted in observing the outcomes *Og* (equation 10) and *Sug* (equation 8) through the application of different factor α to the submission of *n personal opinions*. The number of personal opinions defined to the simulation was 50. The recorded values were compared by the perspective of the mean average *(X)* and mode *(Mod)*. The random sampling was utilized to avoid bias introduction [3] as the typical pattern of submitting personal opinions was not known. Each *personal opinion* had the value scale of submission translated from a quality based to a discrete scale from 1 to 5 (very low to very high). The same discrete scale was used to convert *general opinion* to *suggestion*.

$\bar{x}\,(\lvert\bar{x}(Op) - (Og_i,\alpha)\rvert), i\exists[1..50]$	$\bar{x}\,(\lvert\bar{x}(Op) - (Sug_i,\alpha)\rvert), i\exists[1..50]$
$\begin{cases} \bar{x}\,(\lvert\bar{x}(Op) - (Og_{i,\alpha=0,5})\rvert) \\ \bar{x}\,(\lvert\bar{x}(Op) - (Og_{i,\alpha=0,33})\rvert) \\ \bar{x}\,(\lvert\bar{x}(Op) - (Og_{i,\alpha=0,66})\rvert) \end{cases}$	$\begin{cases} \bar{x}\,(\lvert\bar{x}(Op) - (Sug_{i,\alpha=0,5})\rvert) \\ \bar{x}\,(\lvert\bar{x}(Op) - (Sug_{i,\alpha=0,33})\rvert) \\ \bar{x}\,(\lvert\bar{x}(Op) - (Sug_{i,\alpha=0,66})\rvert) \end{cases}$

Fig. 2: Group of calculations used in study 1

The entity would have a neutral *general opinion* value in the start. In the base simulation two descriptive statistics measurements were inspected:

– *X* – the average mean provides a rough estimate about the average opinion of the users and the behavior of the methods applied to extract general opinions and suggestions.
– *Mod* – the mode indicates the value or range of values where the data concentration is bigger. In the specific case tells about the crewmembers most submitted value and general opinion and suggestion value made.

$\lvert\bar{x}\,(Op) - \bar{x}(Og_{i,\alpha})\rvert, i\exists[1..50]$	$\lvert\bar{x}\,(Op) - \bar{x}(Sug_{i,\alpha})\rvert, i\exists[1..50]$
$\begin{cases} \lvert\bar{x}(Op) - \bar{x}(Og_{i,\alpha=0,5})\rvert \\ \lvert\bar{x}(Op) - \bar{x}(Og_{i,\alpha=0,33})\rvert \\ \lvert\bar{x}(Op) - \bar{x}(Og_{i,\alpha=0,66})\rvert \end{cases}$	$\begin{cases} \lvert\bar{x}(Op) - \bar{x}(Sug_{i,\alpha=0,5})\rvert \\ \lvert\bar{x}(Op) - \bar{x}(Sug_{i,\alpha=0,33})\rvert \\ \lvert\bar{x}(Op) - \bar{x}(Sug_{i,\alpha=0,66})\rvert \end{cases}$

Fig. 3: Group of calculations used in study 2

The stored table data would be processed in three groups of study involving pairs *(Opi, Ogi)* and *(Opi, Sugi)* from each α and a process of study. The first study determines the global average difference between the average personal opinion and the general opinion and suggestion of the simulation. Figure 2 shows the group of calculations used in this study.

Having as reference the mean average personal opinion, the second study finds how the opinion and general suggestion mean average deviates in the simulations. Figure 3 shows the group of calculations used.

6 Results

The first study was conceived to see the average difference in every record between the most regular general opinion and suggestion to the average personal opinion. The difference gives a sense of how close the general opinion and suggestion generated by α is to the average personal opinion. Table 1 shows the result obtained in study 1.

Table 1: Results of Study 1

	α = 0,5		*α = 0,33*		*α = 0,66*	
	Og	*Sug*	*Og*	*Sug*	*Og*	*Sug*
Op	0,59	0,62	0,40	0,57	0,77	0,80

The second study shows the direct difference between the personal opinion average mean to general opinion and suggestion. Greater the value of deviation, greater chance of the general opinion and suggestion not be synchronized with the most prevalent personal opinion. Table 2 shows the result obtained.

Table 2: Results of Study 2

	α = 0,5		*α = 0,33*		*α = 0,66*	
	Og	*Sug*	*Og*	*Sug*	*Og*	*Sug*
Op	0,01	0,00	0,02	0,04	0,00	0,06

The first study compares the performance of the α by every simulation record while the last evaluates the performance globally by processing the stored results of the 50 simulations in regard to the average value.

The α = *0,5* updates the current general opinion by making it lean toward the new opinions with a neutral pace. This value neither highlights the general opinion nor accentuates the contribution of new personal opinions.

The α = *0,33* emphasizes the value of the current general opinion, making the progression to new opinion slower. A factor α = *0,33* can reduce the impact of random extreme opinions which could distort the estimate of a realistic general opinion.

The α = *0,66* facilitates the adoption of new personal opinions. This method is ideal when sudden changes in the personal opinions occur periodically spaced, making the general option harmonize faster. If the behavior of the personal opinions shifts constantly around extreme values makes the general opinion too sensible to the changes. In Table 1 we can see that the α = *0,33* has the better performance. In table

2, α = *0,5* had the lowest deviation score (*Og = 0,01; Sug = 0,00*) making the values returned by his application usually more close to the personal opinion tendency.

To a lower smoothing factor the general opinion and the suggestion tends to be more close to the average personal opinion, while they seem to suffer from worse effectiveness in keeping pace with the most recent personal opinion. By the objective of the analyses both α = *0,5* and α = *0,33* are generally better than the alternative, having the lower deviation and the smaller difference value from the average personal opinion. This performance was already expected at the beginning of the simulation. The combination of these two factors means that lower α gets values more close to the most common personal opinion at a very satisfactory rate. The smoothing value should be updated in the future to increasingly adapt to the behavior of the population:

- If the population tends to have a very regular set of opinions punctuated with sporadic extreme opinions α should decrease.
- If the population shifts to new set of opinions periodically *α* should increase;

In our work we have set the value of α to **0,45.**

7 Conclusions and Future Work

By implementing this Web Portal Interface Agent we hope to improve the dissemination of crucial information to each professional category. All the crewmembers help to define what is important to them at a time and indirectly share that knowledge or opinion between them in the form of general opinion. Because the gathering and processing of the tasks is continuous this makes current suggestions always sensitive to the needs of the users. Two immediate tests can provide more accurate results about the actual performance of the agent, being the first one a more extended use of the interface agent by the crewmembers and evaluation of the crewmember's feedback about the agent behavior and utility. With time, a more precise perception about the general behavior of the crewmembers can become more evident and so help to locate a smoothing factor more efficient. The smoothing factor would be gradually adjusted according to the variations.

Alternatively, a different approach to the calculation of suggestion can be adopted. The current data structure supports the implementation of different methods to find a general opinion by different ways because it stores the data concerning all the past tasks. The past data is an enormous source of information for the application of different statistical methods. An interesting test would be to evaluate the difference in performance between the exponential moving average and suitable method of comparison like mean average or preferably the weighted arithmetic mean. Since the data structure allows with little effort the implementation of different methods for the calculation of relevance we have greater flexibility and a greater chance of adaptability of the agent to current and future patterns of crewmember behavior.

One interesting path would be to expand the number of present category views applied to extract suggestions and so get increasingly more detailed users profiles. While the number of viewpoints to evaluate opinions may grant more insight about the nature of each user and consequently more adequate answers, it comes at a cost.

The cost would derive directly of the rate of growth of the data structure presented in section 3. One future course of action could be to broaden the number of data sources monitored. Currently the Interface Agent is expressly conceived for crewmember in regard to the interaction with the two data sources, *documentation* and *roster*. Yet there is more information sources present in **Web Portal** with the possibility of exploration. In our opinion the interface agent is a benefit working in both ways. By the perspective of crewmember, promotes the timely acquisition of relevant information granting the user with more independence and autonomy. To the **Web Portal**, the major improvement comes with clarification of the notion of crucial content throughout the professional categories. This allows a more efficient diffusion of information without the direct interference of the airline company personal.

Acknowledgments. The second author is supported by FCT (Fundação para a Ciência e Tecnologia) under research grant SFRH/BD/44109/2008. The authors are grateful to TAP Portugal for allowing the use of real data from the company.

References

1. TAP Portugal – Portal DOV [online] http://portaldov.tap.pt, consulted on April 2009.
2. Castro, A. and Oliveira, E.: The rationale behind the development of an airline operations control centre using Gaia-based methodology. IJAOSE, Vol. 2, No. 3, pp.350–377 (2008).
3. Guimarães, Rui Campos and Cabral, José Sarsfield.: ESTATÍSTICA, McGraw-Hill de Portugal. (1997)
4. Jing, Yanguo, Brown, Keith and Taylor, Nick.: Intelligent Interface Agents for a System to Diagnose Eye Disorders. Proceedings 1st IJCAAM, Bologna, Italy, pp. 794-795 (2002).
5. Lenox, T., Hahn, S., Lewis, M., Payne, T., and Sycara, K.: Task Characteristics and Intelligent Aiding, Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics, Nashville, TN, 1123-1127, (2000).
6. Lenox, T., Payne, T., Hahn, S., Lewis, M., and Sycara, K.: Agent-based Aiding for Individual and Team Planning Tasks, Proceedings of IEA 2000/HFES, (2000).
7. Chin, D.: Intelligent interfaces as agents, in *Intelligent user interfaces*. Sullivan, J. and Tyler S. (editors), ACM Press, New York, USA (1991).
8. MOKSAF web site [online]. http://www.cs.cmu.edu/~softagents/moksaf/index.html. Consulted in April 2009.
9. Godoy, D., and Amandi, A.: PersonalSearcher: An Intelligent Agent for Searching Web Pages. Advances in Artificial Intelligence-LNAI 1952, pp. 43-52, (2000).
10. Cordero, D., Roldán, P., Schiaffino, S., and Amandi, A.: Intelligent Agents Generating Personal Newspapers. In Proceedings ICEIS 99, pp. 195–202 (1999).
11. Mitchell, T., Caruana, R., Dermott, J.M., and Zabowski, D.: Experience with a Learning Personal Assistant. Communications of the ACM 37(7), 80–91, (1994).
12. Billsus, D. and Pazzani, M.J.: A personal news agent that talks, learns and explains. In O. Etzioni, J.P.Muller, and J.M. Bradshaw (Eds.), Proceedings of the 3rd Agents'99. ACM Press: Seattle, WA, USA (pp. 268–275), (1999).
13. Schiaffino, S., and Amandi, A.: An Interface Agent Approach to Personalize Users' Interaction with Databases. Journal of Intelligent Information Systems, Vol. 25, No. 3, pp. 251-273 (2005).

Restoring CSP Satisfiability with MaxSAT*

Inês Lynce[1] and Joao Marques-Silva[2]

[1] IST/INESC-ID, Technical University of Lisbon, Portugal
ines@sat.inesc-id.pt
[2] CSI/CASL, University College Dublin, Ireland
jpms@ucd.ie

Abstract. The extraction of a Minimal Unsatisfiable Core (MUC) in a Constraint Satisfaction Problem (CSP) aims to identify a subset of constraints that make a CSP instance unsatisfiable. Recent work has addressed the identification of a Minimal Set of Unsatisfiable Tuples (MUST) in order to restore the CSP satisfiability with respect to that MUC. A two-step algorithm has been proposed: first, a MUC is identified, and second, a MUST in the MUC is identified. This paper proposes an integrated algorithm for restoring satisfiability in a CSP, making use of an unsatisfiability-based MaxSAT solver. The proposed approach encodes the CSP instance as a partial MaxSAT instance, in such a way that solving the MaxSAT instance corresponds to identifying the smallest set of tuples to be removed from the CSP instance to restore satisfiability. Experimental results illustrate the feasibility of the approach.

Key words: constraint satisfaction problems, minimal unsatisfiable cores, minimal set of unsatisfiable tuples, maximum satisfiability

1 Introduction

The identification of unsatisfiable problem instances poses a few questions, including knowing why the instance is unsatisfiable and how it can be repaired to become satisfiable. For example, configuring a product may not always result in a feasible configuration, but in that case the customer would be pleased to receive explanations to understand what made the configuration unfeasible or alternatively to receive some hints about how to make it feasible. Such feedback is in general expected to be as precise as possible, i.e. to identify a minimal reason of unsatisfiability and to minimize the impact of restoring satisfiability.

If one considers Boolean satisfiability (SAT), then these questions are answered identifying Minimal Unsatisfiable Subformulas (MUSes) and obtaining Maximum Satisfiability (MaxSAT) solutions. Similarly, in the context of the Constraint Satisfaction Problem (CSP) these questions are answered by identifying Minimal Unsatisfiable Cores (MUCs) and obtaining Maximum CSP (MaxCSP) solutions. However, as recently pointed out [7], in CSPs one may consider unsatisfiable sets of tuples instead of unsatisfiable sets of constraints. The first is

* Research partially funded by FCT project SHIPs (PTDC/EIA/64164/2006).

considered to be a finer-grained explanation as unsatisfiable sets of tuples may eventually not contain as many tuples as unsatisfiable sets of constraints.

Let us get back again to the configuration of a product. Suppose you have two configuration requirements for your new car: (1) it has to be a sports car and (2) you want it to have the most inexpensive type of seats. It turns out that the car model that you have selected does not allow such configuration. So you may be simply told that those two requirements are not compatible or, much better, that a sports car must have leather seats which is incompatible with having cloth seats. The latter explanation is certainly more precise. Also, it will be easier for you to repair the initial configuration either keeping all but one of the characteristics of a sports car (leather seats) or forgetting about inexpensive seats.

This paper addresses the problem of repairing an unsatisfiable CSP instance by removing the smallest number of tuples. This has in general less impact than removing constraints, and in the worst case the same impact. A MaxSAT encoding is used to solve the problem of repairing a CSP for which constraints are defined as conflicts, with no need of first extracting a MUC. Using an unsatisfiability-based MaxSAT solver has the advantage of identifying unsatisfiable sets of tuples while restoring satisfiability.

The paper is organized as follows. The next section introduces the preliminaries, alongside with illustrative examples. Next, the new approach for restoring satisfiability in a CSP using MaxSAT is described. Experimental results show that the proposed approach is feasible. Finally, the paper concludes and points out future research directions.

2 Preliminaries

2.1 CSPs, MUCs and MUSTs

In what follows we assume that CSP variables have finite domains and that the constraints over variables correspond to conflicts rather than supports.

Definition 1. *(CSP) A CSP instance is a triple (X, D, C) where $X = \{x_1, \ldots, x_n\}$ is a set of n variables, $D = \{d(x_1), \ldots, d(x_n)\}$ is a set of n domains, where each domain $d(x)$ corresponds to the domain of variable $x \in X$, and $C = \{c_1, \ldots, c_m\}$ is a set of m constraints. Each constraint $c \in C$ is a pair $c = (S, R)$ where S is a sequence of variables of X, called the constraint scope, and R is a $|S|$-ary relation over D, called the constraint relation.*

An assignment to a CSP instance is a set $\{(v_1, a_1), \ldots, (v_k, a_k)\}$ where $\{v_1, \ldots, v_k\} \subseteq X$ is a set of variables and $a_i \in d(v_i)$ for all i with $1 \leq i \leq k$. In case $k = n$ the assignment is said to be complete. Assuming that a constraint relation specifies *disallowed* assignments, i.e. the conflicts, a solution to a CSP instance is a complete assignment such that no assignment to a constraint scope is a member of the corresponding constraint relation. A CSP instance for which there is a solution is said to be *satisfiable*; otherwise it is said to be *unsatisfiable*.

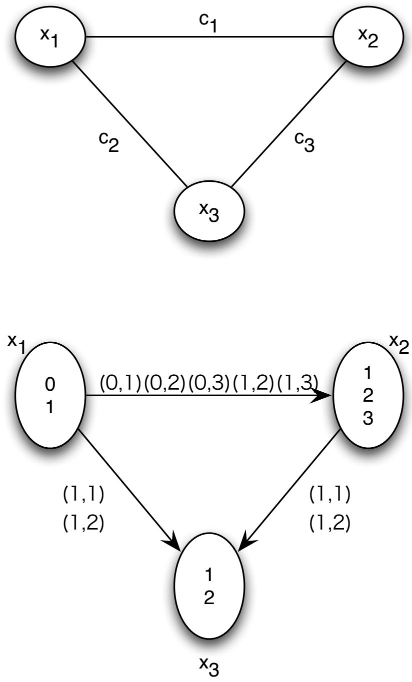


Fig. 1. Graphical representation of the CSP instance from Example 1

Example 1. (CSP) Let P be the CSP instance graphically represented in Figure 1. Formally, $P = (X, D, C)$, with $X = \{x_1, x_2, x_3\}$, $D = \{d(x_1) = \{0, 1\}, d(x_2) = \{1, 2, 3\}, d(x_3) = \{1, 2\}\}$ and $C = \{c_1, c_2, c_3\} = \{((x_1, x_2), \{(0, 1), (0, 2), (0, 3), (1, 2), (1, 3)\}), ((x_1,\ x_3), \{(1, 1), (1, 2)\}), ((x_2, x_3), \{(1, 1), (1, 2)\})\}$. Figure 1 represents P taking into account its variables and constraint scopes (top) and its variable domains and constraint relations (bottom). The arrows denote the order of the variable values in the conflict tuples.

We should note that the constraints of the previous example could be alternatively defined in terms of *allowed* assignments, i.e. the supports. In some cases such approach has the advantage of requiring less tuples. For example, c_1 would become $((x_1, x_2), \{(1, 1)\})$ instead. But in other cases it may be the contrary. For example, c_2 would become $((x_2, x_3), \{(2, 1), (2, 2), (3, 1)\ (3, 2)\})$.

Definition 2. *(UC) An unsatisfiable core (UC) of an unsatisfiable CSP $P = (X, D, C)$ is a CSP instance $P' = (X', D', C')$ such that (1) P' is unsatisfiable and (2) $X' \subseteq X$, $D' \subseteq D$ is the set of domains of variables in X' and $C' \subseteq C$ is the set of constraints with their scopes in X'.*

An *unsatisfiable core* (UC) of an unsatisfiable CSP instance is a CSP instance that is a subset of the former one with respect to its variables, domains and constraints, and is still unsatisfiable. An unsatisfiable CSP instance has at least one UC corresponding to itself.

An unsatisfiable core is expected to be minimal as it should be as precise as possible when identifying the causes of unfeasibility, thus extracting minimal unsatisfiable cores (MUCs). An UC has at least one MUC, and in case it is unique then the MUC is equal to the UC. Removing one constraint per MUC restores satisfiability. A smaller number of constraints suffices when constraints are shared by different MUCs.

Definition 3. *(MUC) A minimal unsatisfiable core (MUC) of an unsatisfiable CSP instance P is an UC of P, say P', such that removing any of the constraints of P' makes the resulting CSP instance satisfiable.*

Providing the user with explanations of unsatisfiability of CSPs has been first addressed with QuickXplain [8]. Moreover, the extraction of MUCs in CSPs has been recently made feasible [5]. The proposed algorithm performs successive runs of a complete backtracking search, using weights, in order to isolate an inconsistent subset of constraints.

Another approach for explaining unfeasibility consists in dealing with the tuples of constraint relations (in the sequel called tuples) rather than constraints defined as a pair with their scopes and relations [7]. This approach has clear advantages. Instead of identifying a set of constraints as the causes of unsatisfiability, a set of tuples is identified. Consequently, restoring satisfiability when considering tuples implies removing tuples rather than constraints, which may have a minor impact on the encoding. This assumes that, when making the encoding of a CSP instance, errors have been introduced when encoding a few tuples rather than when encoding a few constraints (including all its tuples).

Definition 4. *(MUST) A minimal unsatisfiable set of tuples (MUST) of an unsatisfiable CSP $P = (X, D, C)$, with $C = \{(S_1, R_1), \ldots, (S_m, R_m)\}$, is a CSP $P' = (X', D', C')$ such that (1) P' is unsatisfiable, (2) $\forall(S', R') \in C'\ \exists(S, R) \in C : S' = S \wedge R' \subseteq R$, $X' \subseteq X$ is the set of variables in S' and $D' \subseteq D$ is the set of domains of variables in X' and (3) removing any tuple from R' s.t. $(S', R') \in C'$ makes the resulting CSP instance satisfiable.*

It is worth mentioning that tuples in a MUST do not necessarily belong to constraints in a MUC. An illustrative example is given bellow.

Example 2. (MUC and MUST) Figure 2 illustrates MUCs and MUSTs of the CSP instance P introduced in Example 1. P has two MUCs: one with c_1 and c_2 and another one with c_1 and c_3. An intuitive explanation is the following: c_1 implies the assignment $\{(x_1, 1), (x_2, 1)\}$, whereas c_2 *forbids* the assignment $\{(x_1, 1)\}$ and c_3 *forbids* the assignment $\{(x_2, 1)\}$. Any of these MUCs contains seven tuples. Several MUSTs can be identified in P but only two of them are illustrated in the figure (bottom). One of them (left) corresponds to the smallest MUST and contains only five tuples. The point is that not all tuples from

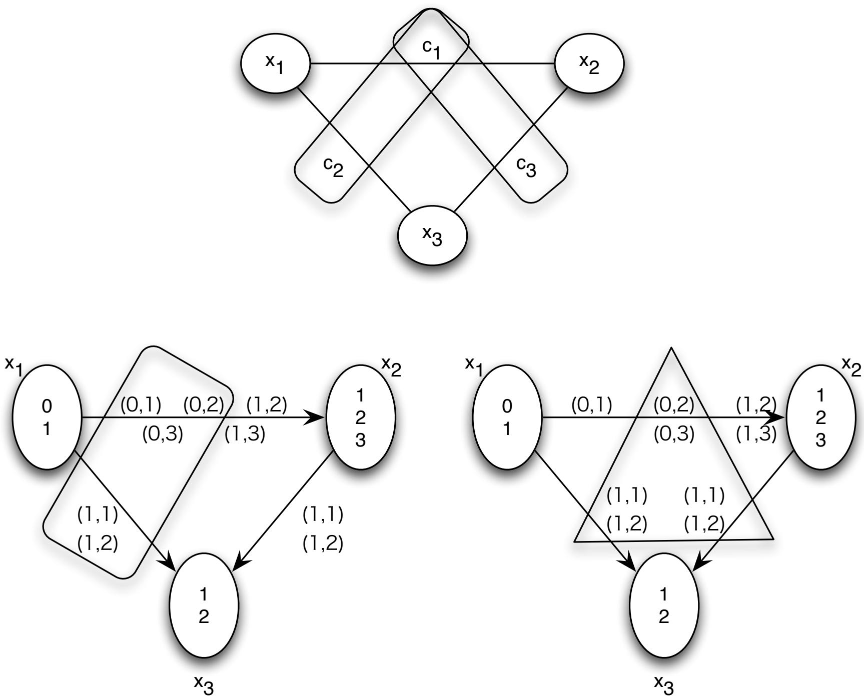


Fig. 2. MUCs and MUSTs of the CSP instance from Example 1

constraint c_1 are required to make the instance unsatisfiable when considering also c_2. The other MUST (right) contains six tuples with the particularity of covering all the constraints.

Satisfiability of a CSP can be restored by removing one tuple of each MUST. Similarly to MUCs, a smaller number of tuples may be removed when tuples are shared by different MUSTs. For the running example, removing a tuple representing either the conflict between assignments $x_1 = 0$ and $x_2 = 2$ or assignments $x_1 = 0$ and $x_2 = 3$ suffices to restore satisfiability.

2.2 SAT, MUSes and MaxSAT

We assume the reader is familiar with the SAT problem that consists in deciding whether there exists a satisfying assignment to a set of Boolean variables such that a given CNF formula is satisfied. A CNF formula is a conjunction of clauses, where a clause is a disjunction of literals and a literal is a Boolean variable or its complement (a positive or a negative literal, respectively).

A large body of research in SAT has been devoted to explaining unfeasibility of CNF formulas in the last decade. These explanations have many different applications and are of the utmost importance in model checking using SAT solvers. For each iteration, if the call to the SAT solver returns no solution, then an explanation is extracted to be incorporated in the next iteration. Clearly, this explanation should be as precise as possible. Explanations including irrelevant clauses are expected to have a negative impact in the subsequent iterations.

The design of tools to identify unsatisfiable subformulas (USes) in SAT formulas has been early addressed by Bruni [2] who used an heuristic approach to identify a subset of clauses that are still unsatisfiable. Later on, Zhang [18] and Goldberg [4] proposed a similar idea of extracting unsatisfiability proofs based on the resolution steps taken for deriving the empty clause. These resolution steps are implicitly behind the conflict clauses added to the formula as a result of the analysis of conflicts during the search. The identification of all the minimal unsatisfiable subformulas (MUSes) in a systematic way was first developed by Liffiton [11, 12] and later used for identifying the smallest MUS in an efficient way [10]. The identification of MUSes has also shown to be competitive recurring to a local search procedure as a preprocessing step to reduce the number of clauses that may be part of the MUS [6].

The MaxSAT problem consists in finding the largest number of clauses that can be satisfied in a CNF formula. The partial MaxSAT problem distinguishes between hard and soft clauses: hard clauses must be satisfied by any solution, while the number of soft clauses satisfied by a solution must be maximized. Partial MaxSAT can be seen as a compromise between SAT and MaxSAT where the hard clauses are treated as a SAT problem and the soft clauses are treated as a MaxSAT problem.

3 Restoring Satisfiability with MaxSAT

The goal of restoring satisfiability of a CSP instance with a minimal impact can be achieved by identifying the smallest size set of tuples to be removed from the CSP constraint relations. Removing these tuples guarantees that the instance is no longer unsatisfiable, whereas adding any of the removed tuples makes the instance unsatisfiable.

With the aim of restoring CSPs satisfiability, recent work by Grégoire *et al.* [7] introduced a two-step algorithm: (1) identify a MUC and (2) identify a MUST in that MUC. This algorithm is called Muster and is outlined in Algorithm 1. The extraction of a MUC and the extraction of a MUST is performed by using tools developed in the past which are publicly available.

This paper suggests restoring CSP satisfiability with MaxSAT, which guarantees restoring satisfiability. This is a clear advantage with respect to Muster, which only removes one MUST. The proposed algorithm involves three steps: (1) encoding the problem into partial MaxSAT, (2) solving the partial MaxSAT instance and (3) identifying the tuples to be removed from the CSP instance given the partial MaxSAT solution. The pseudo code for this procedure is illustrated in Algorithm 2. For each partial MaxSAT instance P' encoding a CSP instance P, there is a mapping between each MaxSAT variable and each domain value of

Algorithm 1 Identifying a MUST in a CSP

```
MUSTER(CSP P)
P' ← Extract MUC from P
P'_SAT ← Encode CSP P' into SAT
                                        ▷ M_{P'−P'_SAT} contains mapping SAT/CSP
Sol_{P'_SAT} ← Extract MUS from P'_SAT
P'' ← CSP with recovered constraint tuples from Sol_{P'_SAT} using M_{P'−P'_SAT}
return P''
```

Algorithm 2 Restoring CSP satisfiability with MaxSAT

```
RESTORECSPWITHMAXSAT(CSP P)
P_PMaxSAT ← Encode P as a partial MaxSAT instance
                                        ▷ M_{P−P_PMaxSAT} contains mapping SAT/CSP
Sol_{P_PMaxSAT} ← PMaxSATsolver(P_PMaxSAT)
P' ← CSP with recovered constraint tuples from Sol_{P_PMaxSAT} using M_{P−P_PMaxSAT}
return P'
```

a CSP variable. A partial MaxSAT solver is then called to solve P'. Its solution is used to identify the set of constraint tuples of P to be removed. In this section, we will explain how to encode a CSP instance into a partial MaxSAT instance to restore satisfiability.

3.1 A MaxSAT Encoding

We recall that given a CSP instance $P = (X, D, C)$ for which the constraints represent conflicts, it is encoded into a SAT instance P' as follows [3]:

- For each variable $x \in X$ and each value in the corresponding domain $d(x) \in D$ is created a Boolean variable.
- For each set of Boolean variables $\{b_1, \ldots, b_{|d(x)|}\}$ associated with the same CSP variable $x \in X$ is created one clause $(b_1 \vee \ldots \vee b_{|d(x)|})$ to ensure that each CSP variable is assigned at least one value and a set of binary clauses $(\neg b_i \vee \neg b_j)$ with $1 \le i < j \le |d(x)|$ to ensure that no more than one value is assigned to a CSP variable. In what follows, these clauses will be called *at least one clauses* and *at most one clauses*, respectively.
- For each tuple $t \in R$ of each constraint $c = (S, R) \in C$ is created a clause with $|S|$ negative literals to ensure that the corresponding values are disallowed. In what follows, these clauses will be called *conflict clauses*.

[3] Details about encoding CSP into SAT and vice versa can be found in [17].

We should note that there exist alternative encodings for guaranteeing that exactly one value is assigned to each CSP variable (see for example [13, 16]).

Example 3. (CSP to SAT) Consider a CSP instance $P = (X, D, C) = (\{x_1, x_2\}, \{d(x_1) = \{1, 2\}, d(x_2) = \{1, 3\}\}, \{((x_1, x_2), \{(1, 1), (2, 3)\})\})$. Let us consider that a Boolean variable b_{ij} corresponds to the domain value $j \in d(x_i) \in D$ of variable $x_i \in X$. The corresponding SAT instance has therefore the following set of clauses: $\{(x_{11} \vee x_{12}), (x_{21} \vee x_{23}), (\neg x_{11} \vee \neg x_{12}), (\neg x_{21} \vee \neg x_{23}), (\neg x_{11} \vee \neg x_{21}), (\neg x_{12} \vee \neg x_{23})\}$. The first two clauses correspond to *at least one clauses*, the next two clauses correspond to *at most one clauses* and the last two clauses correspond to *conflict clauses*.

The CSP to SAT encoding used by MUSTER does not include the *at most one clauses*. Provided that every *at least one clauses* belong to the MUST, computing one MUST amounts to compute one MUS in the CNF formula.

Consider an unsatisfiable CSP instance $P = (X, D, C)$ with constraints representing conflicts and for which the minimum set of tuples to be removed in order to restore satisfiability is to be found. This problem is encoded into a partial MaxSAT instance P' as follows:

– Each *at least one clause* produced by an encoding of P into SAT is a *hard clause* of P'.
– Each *conflict clause* produced by an encoding of P into SAT is a *soft clause* of P'.

Remark 1. When using the MaxSAT encoding to restore satisfiability of a CSP instance, there is no need of adding *at most one clauses* to guarantee a one-to-one correspondence between Boolean variables and CSP variable values.

Proof. We assume that the CSP instance is unsatisfiable. The hard clauses guarantee that at least one value is assigned to each CSP variable. The soft clauses guarantee the smallest number of violated constraints. Having more than one value assigned to a CSP variable, which means that any of those values can be selected to be in the solution, cannot reduce the number of violated constraints.

Proposition 1. *A solution to the MaxSAT instance P' corresponds to the minimum set of tuples to be removed from P in order to restore satisfiability.*

Proof. The hard clauses guarantee that any solution to the partial MaxSAT instance corresponds to a CSP complete assignment. Each soft clause represents one tuple in a constraint. Hence, a solution to the partial MaxSAT instance satisfies the largest number of tuples, which is equivalent to unsatisfying the smallest number of tuples.

As an additional remark, observe that MaxSAT has been used in the past to solve the MaxCSP problem [1], thus allowing to know which constraints have to be removed to restore satisfiability of a CSP. Although the motivation of this use of MaxSAT clearly differs from the one proposed here, the encoding is similar. Observe that a given assignment to the variables in a constraint scope can correspond to at most one tuple of the constraint relation, in which case we say that that that conflict tuple is *violated*. Hence, the number of violated constraints corresponds to the number of violated conflict tuples.

3.2 Unsatisfiability-Based MaxSAT Solvers

Unsatisfiability-based MaxSAT solvers [3, 15, 14] represent an approach for solving MaxSAT problem instances, which contrasts with the traditional MaxSAT solvers based on branch and bound algorithms [9]. Unsatisfiability-based MaxSAT solvers iteratively identify Unsatisfiable Subformulas (USs), not necessarily minimal, which are relaxed after being identified. Clauses are relaxed by adding one relaxation variable per clause. The use of cardinality constraints on the variables used to relax clauses in USs guarantees that a minimum number of clauses is relaxed. Hence, the MaxSAT solution is computed. The identification of clauses in USs is iterated until the resulting formula is satisfiable. In this case, one clause from each identified US is relaxed. A number of variants of MaxSAT algorithms have been proposed [3, 15, 14], which differ on the actual organization of the algorithm, and the way cardinality constraints are encoded to clausal form.

The use of unsatisfiability-based MaxSAT solvers has the advantage of having a solver which identifies unsatisfiable cores while running. Hence, in case a solution is not found within the allowed CPU time, there is still relevant information for restoring satisfiability.

In this paper we will be using the unsatisfiability-based MSUnCore [15, 14] version of April 2009. For this reason, we will use the name MSUnCore to denote the MaxSAT-based algorithm for restoring CSPs satisfiability.

3.3 Muster *vs* MSUnCore

A few interesting remarks are made bellow in order to stress that MSUnCore should be considered as an alternative approach to Muster.

We should first clarify how the Muster algorithm could be instrumented to restore CSPs satisfiability. First, a MUC is identified in a CSP. Second, MUSTs in the MUC are identified to restore satisfiability, i.e. the required tuples are removed from the MUC such that it becomes satisfiable. Now these same tuples are removed from the CSP and the whole process is repeated.

Remark 2. The solution provided by MSUnCore removes a number of tuples that is equal or smaller than the iterated application of Muster.

Proof. Muster first identifies a MUC and then MUSTs within the MUC. This does not guarantee removing tuples shared by MUSTs belonging to different MUCs. (Figure 2 illustrates this issue: unless Muster is lucky enough to remove tuple $(0, 2)$ or tuple $(0, 3)$, Muster will remove 2 tuples in two iterations, while MSUnCore will remove only one tuple.)

Remark 3. The number of Unsatisfiable Sets of Tuples (USTs) identified by MSUnCore during the search provides a lower bound for the number of tuples to be removed to restore satisfiability.

Proof. Unsatisfiability-based MaxSAT solvers (of which MSUnCore is a concrete example), iteratively identify and relax unsatisfiable subformulas (USs). Moreover, for each iteration for which a US is found, the number of clauses to relax is increased by 1. Consequently, at each step of the MSUnCore algorithm, the number of identified USs represents a lower bound on the MaxSAT solution, and so represents a lower bound on the number of tuples to be removed to restore satisfiability.

4 Experimental Evaluation

The experimental evaluation was performed over a set of instances from the First CSP Competition (`http://cpai.ucc.ie/05/CallForSolvers.html`). This set of instances corresponds to the set of instances used to test Muster [7].

Table 1 provides the characterization and results for each instance. Each instance is characterized in terms of the number of CSP variables (#vars) and constraints (#constr). In addition, the CPU time required by Muster to identify one MUC and one MUST in that MUC is given (1MUC+1MUST), as well as the CPU time required by an unsatisfiability-based MaxSAT solver (MSUnCore [14]) to identify the first UST (1UST), which is not necessarily minimal. Additional information gives the CPU time required by MSUnCore to restore satisfiability (AllMUSTs), which implies eliminating all MUSTs, as well as the total number of tuples to be removed to restore satisfiability (RemovedTuples). The CPU time was limited to 1000 seconds (TO means timeout). In case only a few USTs are identified, it is given a lower bound on the number of tuples to be removed to restore satisfiability.

The results included for Muster where taken from [7]. We should therefore note that the machine used for running MSUnCore is not significantly different from the one used for running Muster. Muster was run on a Pentium IV 3GHz under Linux Fedora Core 5, whereas MSUnCore was run on a Intel Xeon 5160 3GHz under RedHat Enterprise Linux WS4. The performance difference is therefore not significant for the conclusions to be drawn below.

First of all, the results illustrate the feasibility of the proposed approach. Many times MSUnCore is able to restore satisfiability faster than Muster is able to identify one MUST in a MUC. Also interesting is to note that most instances are restored in terms of satisfiability by removing very few tuples. This supports the claim that removing tuples has a very little impact. For the cases where MSUnCore is not able to completely restore satisfiability, it is able to identify a few USTs though, which gives a lower bound on the number of tuples to be removed. In any case, the identification of the first UST by MSUnCore is *always* faster than using Muster. Also, MSUnCore seems to require more time when the number of tuples to be removed is larger.

Table 1. Experimental results

Instance Name	CSP #vars	CSP #constr	Muster 1MUC+1MUST	MSUnCore 1UST	MSUnCore AllMUSTs	MSUnCore RemovedTuples
composed-25-1-2-0	224	4,440	23.80	0.01	0.04	1
composed-25-1-2-1	224	4,440	26.56	0.01	0.67	3
composed-25-1-25-8	247	4,555	15.10	0.01	0.18	2
composed-75-1-2-1	624	10,440	77.27	0.01	0.17	2
composed-75-1-2-2	624	10,440	81.18	0.01	0.19	2
composed-75-1-25-8	647	10,555	82.78	0.01	0.25	2
composed-75-1-80-6	702	10,830	69.65	0.01	0.16	2
composed-75-1-80-7	702	10,830	392.08	0.01	0.05	1
composed-75-1-80-9	702	10,830	95.17	0.01	0.18	2
qk_10_10_5_add	55	48,640	TO	74.62	92.89	1
qk_10_10_5_mul	105	49,140	2814.28	99.55	111.16	1
qk_8_8_5_add	38	19,624	548.03	6.04	13.44	1
qk_8_8_5_mul	78	19,944	531.90	10.58	19.36	1
graph2_f25	2,245	145,205	853.41	0.36	1.64	1
qa_3	40	800	8.64	0.01	0.06	1
dual_ehi-85-297-14	4,111	102,234	43.61	0.14	TO	≥8
dual_ehi-85-297-15	4,133	102,433	29.88	0.15	TO	≥8
dual_ehi-85-297-16	4,105	102,156	33.73	0.15	TO	≥8
dual_ehi-85-297-17	4,102	102,112	47.04	0.15	TO	≥8
dual_ehi-85-297-18	4,120	102,324	33.88	0.15	TO	≥8
dual_ehi-90-315-21	4,388	108,890	38.16	0.28	TO	≥8
dual_ehi-90-315-22	4,368	108,633	48.26	0.26	TO	≥8
dual_ehi-90-315-23	4,375	108,766	15.09	0.25	TO	≥8
dual_ehi-90-315-24	4,378	108,793	29.99	0.21	TO	≥7
dual_ehi-90-315-25	4,398	108,974	34.30	0.20	TO	≥8
scen6_w2	648	513,100	TO	2.51	TO	≥7
scen6_w1_f2	319	274,860	TO	17.25	31.83	1
scen11_f10	4,103	738,719	602.53	1.99	TO	≥4
scen11_f12	4,103	707,375	541.95	1.87	TO	≥3

5 Conclusions and Future Work

This paper suggests the use of MaxSAT to restore CSP satisfiability by removing the smallest number of constraint tuples. This solution contrasts with a MaxCSP solution as tuples instead of constraints are removed. We argue that removing tuples is more adequate for most of the problems due to having a minor impact.

Future work includes adapting MSUnCore for identifying *minimal* unsatisfiable cores. This will have the advantage of identifying MUSTs instead of USTs, which are more relevant in case the search does not terminate in the given timeout.

References

1. J. Argelich, A. Cabiscol, I. Lynce, and F. Manyà. Modelling Max-CSP as partial Max-SAT. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 1–14, 2008.
2. R. Bruni. On exact selection of minimally unsatisfiable subformulae. *Annals of Mathematics and Artificial Intelligence*, 43(1):35–50, 2005.
3. Z. Fu and S. Malik. On solving the partial MAX-SAT problem. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 252–265, August 2006.
4. E. I. Goldberg and Y. Novikov. Verification of proofs of unsatisfiability for CNF formulas. In *Design, Automation and Test in Europe Conference*, pages 10886–10891, 2003.
5. É. Grégoire, B. Mazure, and C. Piette. Extracting MUSes. In *European Conference on Artificial Intelligence*, pages 387–391, 2006.
6. É. Grégoire, B. Mazure, and C. Piette. Local-search extraction of MUSes. *Constraints*, 12(3):325–344, 2007.
7. É. Grégoire, B. Mazure, and C. Piette. MUST: Provide a finer-grained explanation of unsatisfiability. In *International Conference on Principles and Practice of Constraint Programming*, pages 317–331, 2007.
8. U. Junker. QUICKXPLAIN: Preferred explanations and relaxations for over-constrained problems. In *AAAI National Conference on Artificial Intelligence*, pages 167–172, 2004.
9. C. M. Li and F. Manyà. MaxSAT, hard and soft constraints. In A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors, *SAT Handbook*, pages 613–631. IOS Press, 2009.
10. M. Liffiton, M. Mneimneh, I. Lynce, Z. Andraus, J. Marques-Silva, and K. Sakallah. A branch and bound algorithm for extracting smallest minimal unsatisfiable subformulas. *Constraints, In Press*, 2009.
11. M. Liffiton and K. Sakallah. On finding all minimally unsatisfiable subformulas. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 173–186, 2005.
12. M. Liffiton and K. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. *Journal of Automated Reasoning*, 40(1):1–33, 2008.
13. J. Marques-Silva and I. Lynce. Towards robust CNF encodings of cardinality constraints. In *International Conference on Principles and Practice of Constraint Programming*, pages 483–497, 2007.
14. J. Marques-Silva and V. Manquinho. Towards more effective unsatisfiability-based maximum satisfiability alg orithms. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 225–230, 2008.
15. J. Marques-Silva and J. Planes. Algorithms for maximum satisfiability using unsatisfiable cores. In *Design, Automation and Test in Europe Conference*, pages 408–413, 2008.
16. C. Sinz. Towards an optimal CNF encoding of boolean cardinality constraints. In *International Conference on Principles and Practice of Constraint Programming*, pages 827–831, 2005.
17. T. Walsh. SAT v CSP. In *International Conference on Principles and Practice of Constraint Programming*, pages 441–456, 2000.
18. L. Zhang and S. Malik. Validating SAT solvers using an independent resolution-based checker: Practical implementations and other applications. In *Design, Automation and Test in Europe Conference*, pages 10880–10885, 2003.

An Evaluation of Heuristic Functions for Bicriterion Shortest Path Problems ⋆

E. Machuca, L. Mandow and J.L. Pérez de la Cruz

Dpto. Lenguajes y Ciencias de la Computación. Universidad de Málaga 29071 - Málaga (Spain).
{machuca, lawrence, perez}@lcc.uma.es

Abstract. The paper considers a set of heuristic search algorithms for the Bicriterion Shortest Path Problem. These are presented as particular cases of a more general algorithm. Their performance is evaluated over a random problem set using two general heuristic functions proposed by Tung & Chew (1992). The experimental results show that the heuristics reduce the space requirements of the algorithms considered. One contribution of this paper is to show that, contrary to intuition, no improvement in time performance is achieved in these cases by using heuristics. In fact, one of the heuristics appears to work against time efficiency. The paper provides also an explanation to the observed phenomena and points out possible lines of improvement.

1 Introduction

Heuristic search in Shortest Path Problems is a central field of study in Artificial Intelligence. The A* algorithm [1] [2][3] is an efficient solution for the case of admissible graph search guided by single-objective heuristic evaluation functions.

The Bicriterion Shortest Path Problem (BSP) is an extension of the Shortest Path Problem with practical applications in different domains, like routing in multimedia networks [4], route planning [5] satellite scheduling [6], or domain independent planning [7]. BSP problems require the evaluation of two different and frequently conflicting objectives for each alternative. These problems rarely have a single optimal solution. Most frequently, a set of *nondominated* (Pareto-optimal) solutions can be found, each one presenting a particular trade-off between the objectives under consideration. The number of nondominated solutions in BSP problems is known to grow exponentially with solution depth in the worst case [8]. Fortunately, several classes of interesting multiobjective problems do not exhibit this worst-case behavior [9].

The Artificial Intelligence and Operations Research communities have contributed several extensions of the A* algorithm to the multiobjective case. These include Tung-Chew [10], MOA* [11], and NAMOA* [12]. One of the fundamental differences between these algorithms is their path/node selection strategies. The selection strategy of NAMOA* was shown to have better formal properties than that of MOA*, and to improve in space requirements both formally and empirically [12] [13]. However, to the

⋆ This work is partially funded by/Este trabajo está parcialmente financiado por: Consejería de Innovación, Ciencia y Empresa. Junta de Andalucía (España), P07-TIC-03018

author's knowledge, there are no results comparing the performance of the heuristic selection strategies of NAMOA* and Tung-Chew.

The work of Tung & Chew [10] described also two general heuristic functions for multiobjective shortest path problems. The real impact of these heuristic functions in algorithm performance has never been tested with empirical results. This paper considers a general bicriterion search procedure that encompasses both NAMOA* and Tung-Chew. The impact of Tung and Chew's heuristics in the time and space requirements of several particular instances of this procedure is evaluated over a random problem set. Several of the algorithms considered exhibit a similar reduction in space requirements when compared to uninformed search. However, contrary to intuition, no improvement in time performance is achieved. In fact, one of the heuristics considered seems to work against time efficiency. The paper provides an explanation to the observed phenomena and points out possible lines of improvement.

The structure of the paper is as follows. Section 2 introduces some common terminology useful to understand BSP problems and a description of the heuristics analyzed in the paper. It also describes a general multiobjective search procedure that helps to identify the four particular algorithms used in the experiments. Section 3 describes the experimental setup, and analyzes the space and time requirements of the algorithms considered. An adequate explanation of the observed phenomena is presented. Finally some conclusions and future work are outlined.

2 Algorithms

2.1 Bicriterion Shortest Path Problems

Let us consider two q-dimensional vectors $\boldsymbol{v}, \boldsymbol{v}' \in \mathbb{R}^q$. A partial order relation $\prec$ denominated *dominance* is defined as follows, $\boldsymbol{v} \prec \boldsymbol{v}'$ iff $\forall i (1 \leq i \leq q)\ v_i \leq v'_i$ and $\boldsymbol{v} \neq \boldsymbol{v}'$, where v_i denotes the i-th component of vector $\boldsymbol{v}$.

Given two q-dimensional vectors $\boldsymbol{v}$ and $\boldsymbol{v}'$ (where $q > 1$), it is not always possible to say that one is better than the other. For example in a bidimensional cost space vector $(2, 3)$ dominates $(2, 4)$, but no dominance relation exists between $(2, 3)$ and $(3, 2)$.

Given a set of vectors X, we shall define $nd(X)$ as the set of non-dominated vectors in X, i. e., $nd(X) = \{\boldsymbol{x} \in X \mid \nexists \boldsymbol{y} \in X \quad \boldsymbol{y} \prec \boldsymbol{x}\}$

A total order relation $\prec_L$ denominated *lexicographic order* is defined as follows, $\boldsymbol{v} \prec_L \boldsymbol{v}'$ iff for some j, $v_j < v'_j$ and $\forall i < j$, $v_i = v'_i$. A useful property of the lexicographic order is that the lexicographic optimum in a set of vectors is also a non-dominated vector.

Let G be a locally finite labeled directed graph $G = (N, A, \boldsymbol{c})$ with $|N|$ nodes and $|A|$ arcs (n, n') labeled with positive vectors $\boldsymbol{c}(n, n') \in \mathbb{R}^q$. The cost of a path is defined as the sum of the costs of its arcs; obviously, the cost of a path is a q-dimensional vector. A multiobjective search problem in G is stated as follows:

Given a start node $s \in N$ and a set of goal nodes $\Gamma \subseteq N$, find the set of all *non-dominated* paths P in G, i. e., the set of all paths P such that

i) P goes from s to a node in Γ;

ii) the cost of P is non-dominated by the costs of any other path satisfying i).

The Bicriterion Shortest Path Problem (BSP) is a particular case of MSP in which $q = 2$, i. e., arc costs have two real components.

2.2 A general algorithm

The general algorithm ran in the experiments reported in this paper, is presented in table 1. This algorithm is a slight generalization of NAMOA* [12]. In fact, the only difference with NAMOA* lies in step 3. This difference will be explained in subsection 2.4.

The main features of the algorithm are as follows:

– The algorithm uses an acyclic search graph SG to record interesting partial solution paths. For each node n in SG two sets —$G_{cl}(n)$ and $G_{op}(n)$— denote the sets of non-dominated cost vectors of paths reaching n that have or have not been explored yet respectively (i.e. closed or open). Each cost vector in these sets labels one or more pointers emanating from n to its parents with that cost. Initially, s is the only node in SG.
– The algorithm keeps a list $OPEN$ of partial solution paths that can be further expanded. For each node n in SG and each nondominated cost vector $\boldsymbol{g} \in G_{op}(n)$ there will be a corresponding label $(n, \boldsymbol{g}, F(n, \boldsymbol{g}))$ in $OPEN$. Initially, the tuple $(s, \boldsymbol{g}_s, F(s, \boldsymbol{g}_s))$ is the only label in $OPEN$. Notice that contrary to what happens in single-objective search, many different nondominated paths may reach a given node. Therefore, it is the number of cost vectors stored in $G_{op}(n)$ and $G_{cl}(n)$ what determines memory requirements, while the number of nodes plays a minor role.
– At each iteration the algorithm will consider the extension of an open label $(n, \boldsymbol{g}, F)$ that stands for a partial solution path from s to n with cost $\boldsymbol{g}$.
– Two sets —$GOALN$ and $COSTS$— record all goal nodes reached and all non-dominated cost vectors to goal nodes respectively. Once a solution is known, its cost vector can be used to discard (filter) dominated open labels.
– Search terminates only when the $OPEN$ list is empty, i.e. when all open labels have been selected or filtered.
– Heuristic estimates will normally involve a set of vectors $H(n)$ for each node n. These vectors estimate the costs of nondominated paths from n to each goal node. Let $\boldsymbol{g}(P)$ be the function that returns the cost of path P, defined by the sum of the costs of all its component arcs. Therefore, for each path P_{sn} from s to n with cost $\boldsymbol{g}(P_{sn}) = \boldsymbol{g}_P$, there will be a set of heuristic evaluation vectors $F(P_{sn})$. This function is the multiobjective analogue of $f(n)$ in A*, $F(P_{sn}) = F(n, \boldsymbol{g}_P) = nd(\{\boldsymbol{f} \mid \boldsymbol{f} = \boldsymbol{g}_P + \boldsymbol{h} \ \wedge \boldsymbol{h} \in H(n)\})$. In this paper, we will consider the situation where $H(n) = \{\boldsymbol{h}(n)\}$, i.e., there is only one vector estimate $\boldsymbol{h}$. Therefore $F(P_{sn}) = \{\boldsymbol{f}\}$, $\boldsymbol{f} = \boldsymbol{g} + \boldsymbol{h}$.

2.3 Heuristic functions

The main goal of this paper is the comparative study of the performance of the general algorithm when instantiated with some heuristic functions. Three heuristic functions will be considered.

1. CREATE:
   - An acyclic search graph SG rooted in s.
   - List of alternatives, $OPEN = \{(s, \boldsymbol{g}_s, F(s, \boldsymbol{g}_s))\}$.
   - Two empty sets, $GOALN$, $COSTS$.
2. CHECK TERMINATION. If $OPEN$ is empty, then backtrack in SG from the nodes in $GOALN$ and return the set of solution paths with costs in $COSTS$.
3. PATH SELECTION. Select an alternative $(n, \boldsymbol{g}_n, F)$ from OPEN. Delete $(n, \boldsymbol{g}_n, F)$ from $OPEN$, and move $\boldsymbol{g}_n$ from $G_{op}(n)$ to $G_{cl}(n)$.
4. SOLUTION RECORDING. If $n \in \Gamma$, then
   - Include n in $GOALN$ and $\boldsymbol{g}_n$ in $COSTS$.
   - Eliminate from $OPEN$ all alternatives (x, g_x, F_x) such that all vectors in F_x are dominated by $\boldsymbol{g}_n$ (FILTERING).
   - Go back to step 2
5. PATH EXPANSION: If $n \notin \Gamma$, then
   For all successors nodes m of n that do not produce cycles in SG do:
   (a) Calculate the cost of the new path found to m: $\boldsymbol{g}_m = \boldsymbol{g}_n + \boldsymbol{c}(n, m)$.
   (b) If m is a new node
      i. Calculate $F_m = F(m, \boldsymbol{g}_m)$ filtering estimates dominated by $COSTS$.
      ii. If F_m is not empty, put $(m, \boldsymbol{g}_m, F_m)$ in $OPEN$, and put $\boldsymbol{g}_m$ in $G_{op}(m)$ labelling a pointer to n.
      iii. Go to step 2.

   else (m is not a new node), in case
   - $\boldsymbol{g}_m \in G_{op}(m)$ or $\boldsymbol{g}_m \in G_{cl}(m)$: label with $\boldsymbol{g}_m$ a pointer to n, and go to step 2.
   - If $\boldsymbol{g}_m$ is non-dominated by any cost vectors in $G_{op}(m) \cup G_{cl}(m)$ (a path to m with new cost has been found), then :
      i. Eliminate from $G_{op}(m)$ and $G_{cl}(m)$ vectors dominated by $\boldsymbol{g}_m$
      ii. Calculate $F_m = F(m, \boldsymbol{g}_m)$ filtering estimates dominated by $COSTS$.
      iii. If F_m is not empty, put $(m, \boldsymbol{g}_m, F_m)$ in $OPEN$, and put $\boldsymbol{g}_m$ in $G_{op}(m)$ labelling a pointer to n.
      iv. Go to step 2.
   - Otherwise: go to step 2.

Table 1. A general multiobjective search algorithm.

When no heuristic information is available, the trivial heuristic function is given by $\boldsymbol{h}_0(n) = \mathbf{0}$, for all nodes n.

Tung and Chew [10] proposed two nontrivial heuristic functions that are well defined for every Bicriterion Path Problem. The first heuristic [1] will be called in this paper $\boldsymbol{h}_{12}(n) = (h_1(n), h_2(n))$ and is computed as follows: consider a graph G_i $(i = 1, 2)$ with same nodes than the given graph G and arcs reversed. For each arc (n', n) in G_i a scalar cost is defined, given by the i-th component of the cost $\boldsymbol{c}(n, n')$ in G. Then apply Dijkstra's algorithm [14] to this graph to find the shortest path from a terminal node t to the usual starting node s, breaking the ties by a lexicographic order. In this way, we compute for each node n, a value $h_i(n)$, that is, the cost of the "best" path from t to n (i. e., from n to t), considering only the i-th component of the cost. It is obvious that

[1] Tung and Chew [10] call this heuristic q.

the vectorial heuristic $\boldsymbol{h}_{12}(n) = (h_1(n), h_2(n))$ is optimistic, i. e., $\boldsymbol{h}_{12}(n) \prec \boldsymbol{f}^*(n)$ for every node n.

The second heuristic [2] proposed by Tung and Chew will be called in this paper h_{mix} and is defined in a similar way, as the result from applying Dijkstra's algorithm to the graph resulting from reversing every arc, but the cost is now the sum of the components of the cost of the original graph, i. e., arc (n', n) is labeled with $c(n', n) = \sum_i c_i(n, n')$. Notice that h_{mix} is a scalar heuristic.

Algorithm	Selection rule	Additional selection rule	Filtering criteria
NAMOA-LEX-H0	best $\boldsymbol{g}$	lex order	$\boldsymbol{g}$ dominated
NAMOA-LEX	best $\boldsymbol{g} + \boldsymbol{h}_{12}$	lex order	$\boldsymbol{g} + \boldsymbol{h}_{12}$ dominated
TC-BS	best $\sum_i g_i$		$\boldsymbol{g} + \boldsymbol{h}_{12}$ dominated
TC-HS	best $h_{mix} + \sum_i g_i$		$\boldsymbol{g} + \boldsymbol{h}_{12}$ dominated

Table 2. Instantiations of the general algorithm.

2.4 Instantiation of the general algorithm

Heuristic functions $\boldsymbol{h}(n)$ do not appear explicitly in the pseudocode in table 1. However, there are four points where $\boldsymbol{h}(n)$ is implicitly invoked:

- Step 3, PATH SELECTION. Usually the node selected from *OPEN* is non-dominated according to certain heuristic function.
- Step 4, SOLUTION RECORDING (filtering). All alternatives $(x, \boldsymbol{g}_x, F_x)$, such that all vectors in F_x are dominated by the cost $\boldsymbol{g}_n$ of a newly found solution, are eliminated from *OPEN*. Since we are assuming that there is just an element $\boldsymbol{f}_x \in F_x$, $\boldsymbol{f}_x = \boldsymbol{g}_x + \boldsymbol{h}_x$, the heuristic function also plays a role at this step.
- Step 5(b).i and 5(b).iii.ii. These are two new occurrences of filtering when performing the step PATH EXPANSION. Filtering depends on the values of F_x and hence on the values of the heuristic function [3].

Algorithm NAMOA* [12] is just the algorithm of table 1 when the same heuristic function is applied both for path selection (step 3) and for filtering. Since h_{mix} is a scalar function, it could not be sensibly used for filtering. Therefore, considering the heuristic functions defined in subsection 2.3, there will be two instances of NAMOA: blind NAMOA* (with $\boldsymbol{h}_0$ for selecting and filtering) and informed NAMOA* (with $\boldsymbol{h}_{12}$ for selecting and filtering). In both cases, the selection procedure of step 3 could be stated as follows:

PATH SELECTION (version A). Select an alternative $(n, \boldsymbol{g}_n, \{\boldsymbol{f}\})$ from OPEN such that $\nexists (n', \boldsymbol{g}_{n'}, \{\boldsymbol{f}'\}) \in OPEN \quad | \quad \boldsymbol{f}' \prec \boldsymbol{f}$.

[2] Tung and Chew [10] call this heuristic with the somewhat confusing name h^*.

[3] Notice that the original *algorithm* proposed by Tung and Chew [10] does not perform all these filtering operations, or performs them in a different way

In other words, a nondominated $\boldsymbol{f}$ is selected from OPEN. This guarantees admissibility [12]. But this rule is not enough to determine the selected path; usually there are many nondominated $\boldsymbol{f}$ in OPEN. An additional rule is needed. The original description of NAMOA* [12] suggests LEX, the lexicographical order, as a suitable rule. Therefore, we consider two instantiations of the general algorithm with version A of path selection: NAMOA-LEX-H0 (blind NAMOA-LEX) and NAMOA-LEX (informed NAMOA-LEX).

If path selection is not dictated by the same heuristic function used for filtering, other possibilities appear. Tung and Chew [10] proposed the following selection rule that uses the h_{mix} heuristic (preserving $\boldsymbol{h}_{12}$ for filtering):

PATH SELECTION (version B). Select an alternative $(n, \boldsymbol{g}_n, \{\boldsymbol{f}\})$ from OPEN such that $\forall(n', \boldsymbol{g}_{n'}, \{\boldsymbol{f}'\}) \in$ OPEN, it holds that $h_{mix} + \sum_i g_i \leq h'_{mix} + \sum_i g'_i$.

The corresponding instance of the algorithm will be called TC-HS (for Tung & Chew with Heuristic Selection).

In order to evaluate the impact of the h_{mix} heuristic, we shall consider an additional blind selection rule:

PATH SELECTION (version C). Select an alternative $(n, \boldsymbol{g}_n, \{\boldsymbol{f}\})$ from OPEN such that $\forall(n', \boldsymbol{g}_{n'}, \{\boldsymbol{f}'\}) \in$ OPEN, it holds that $\sum_i g_i \leq \sum_i g'_i$.

It is easily proven that minimizing $\sum_i g_i$ yields a nondominated $\boldsymbol{f}$,(just consider that $\boldsymbol{h}_0(n)$ is used for selection). Since this rule is a “blind” version of that in Tung and Chew [10], the corresponding instance of the algorithm will be called TC-BS (for Tung & Chew with Blind Selection).

Table 2 sums up the features of the four algorithms studied in this paper.

3 Empirical evaluation

Bidimensional square grids without obstacles were used to test the algorithms. Biobjective cost vectors (c_1, c_2) were generated for each arc. The values c_1, c_2 were independently and randomly generated from a uniform distribution of integer values in the range $[1, 10]$. Ten different sets of grids with sizes from 10×10 to 100×100 were generated. The results presented for each size are averaged over the ten different test sets. Search was conducted in all of the problems from one corner of the grid to the opposite. Therefore, solution depth varies from 20 to 200 in steps of 20. The results discussed in the following describe the number of iterations, space, and time requirements of each algorithm as a function of grid size.

The tests were run on a Windows XP 32-bit platform, with an Intel Core2 Quad Q9550 at 2.8Ghz, and 4Gb of RAM. The algorithms were implemented to share as much code as possible, using the LispWorks Professional 5.01 programming environment. Nevertheless, important differences are needed. Both informed and blind NAMOA-LEX share all their code and only differ in the heuristic function provided. The G_{op} sets were ordered lexicographically. Both TC-HS and TC-BS share also all of their code. However, the G_{op} sets were ordered according to their respective linear evaluation functions. In all implementations, only the current best cost estimate of each node was kept in $OPEN$ at each iteration, as suggested in [12]. The $OPEN$ list was implemented as a binary heap.

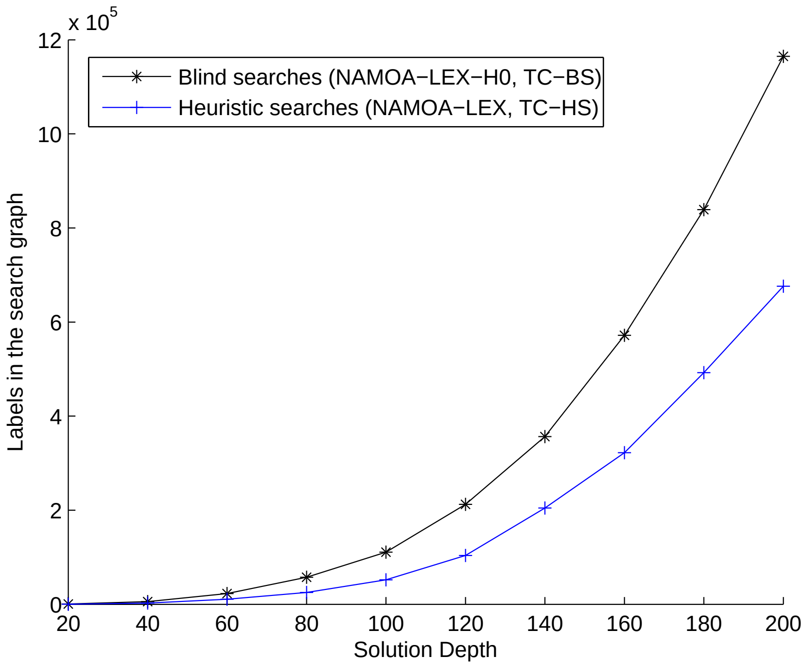


Fig. 1. Iterations/Space requirements at different solution depths, averaged over ten problem sets.

3.1 Number of iterations and space requirements

Most analysis of heuristic search algorithms concentrate on the number of iterations as a measure of their complexity. Figure 1 shows the number of iterations of the algorithms as a function of grid size. Both heuristic searches (NAOMA-LEX and TC-HS) obtained undistinguishable results. NAMOA-LEX and TC-HS exhibit an important reduction in the number of iterations (i.e labels considered) when compared with blind NAMOA-LEX-H0 and TC-BS. Again, blind searches (NAMOA-LEX-H0 and TC-BS) obtained indistinguishable results. The reduction in the number of iterations using heuristics is on average 45%. The number of labels stored in the search graph was found in all cases to equal the number of iterations. This is not surprising, since the label expanded by the algorithms at each iteration is permanently stored in the G_{cl} sets. Therefore, space requirements grow fast with solution depth. The number of nodes in each grid ($n \times n$) is comparatively much smaller than the total number of labels.

This is an important result for the heuristics under consideration, since most applications of bicriterion search described in the literature report difficulties with space requirements.

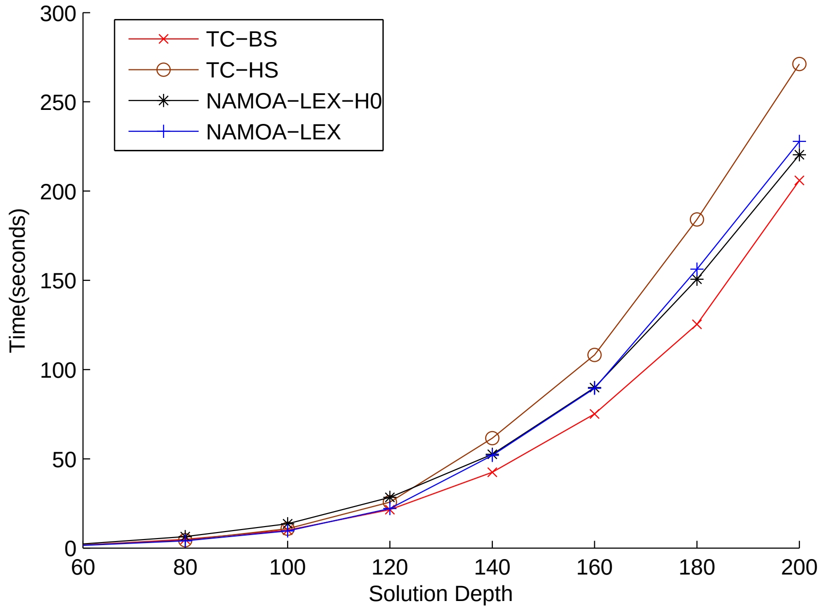


Fig. 2. Time requirements at different solution depths, averaged over ten problem sets.

3.2 Time requirements

A first conclusion from the experiments is that the time needed to calculate the heuristic values is not significant compared to total execution time (less than 5% on average, and less than 1% in the largest problems). This is due to the fact that the single objective search runs, used to precompute the heuristic values, have much smaller time requirements than bicriterion search.

Time requirements of all algorithms are shown in figure 2. Contrary to intuition, the experiments do not show the decrease in time that could be expected from the reduction in the number of iterations of the NAMOA-LEX and TC-HS algorithms. NAMOA-LEX has requirements similar to those of uninformed NAMOA-LEX-H0 in spite of the fact that the number of iterations is much smaller. Surprisingly, TC-BS takes significantly less time than TC-HS. In fact, TC-BS was the fastest algorithm and TC-HS the slowest. It is obvious from these results that search time cannot be easily extrapolated from the number of iterations. This is due to the fact that in bicriterion search, label expansion is by no means an atomic constant-time operation.

Bicriterion search algorithms share with scalar ones a variability in time per iteration due to the need to sort open alternatives. In bicriterion search the number of open alternatives is low at the beginning and end of the search, and high some time in between. Additionally, each new label selected for expansion generates a number of successor labels that need to be checked for dominance with the labels in the $G_{op}(n)$ and $G_{cl}(n)$

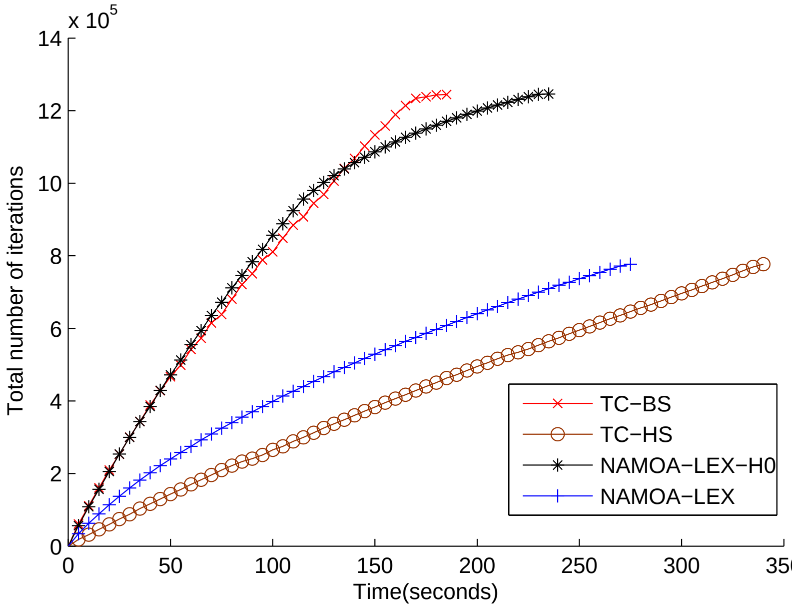


Fig. 3. Number of iterations vs. time for a particular search problem.

sets of the destination node n, as well as the set of nondominated labels found to the goal nodes (i.e. those kept in the $COSTS$ set). Particularly, the size of $COSTS$ can grow rapidly with solution depth. Therefore, the costly dominance checks used in the filtering steps Step 5(b).i and 5(b).iii.ii of the algorithm in table 1 seem to dominate time requirements.

Our analysis reveals that time per iteration is highly influenced by the number of solutions already found at each iteration (i.e. the size of the $COSTS$ set at that time). Figures 3 and 4 illustrate the behavior of the algorithms for one particular problem of size 100×100. Figure 3 shows that both TC-BS and NAMOA-LEX-H0 performed the same number of iterations in this problem. However, TC-BS performed a roughly constant number of iterations per second and abruptly slowed down at the end, while NAMOA-LEX-H0 slowed down more gradually and finally took more time to finish. Notice that TC-HS and NAMOA-LEX performed less iterations than the previous algorithms, but required more time per iteration from the beginning, specially TC-HS. The result is that these algorithms were slower while solving the same problem instance. Similar behaviour was found in all cases analyzed.

The explanation of this behavior can be found in figure 4. All algorithms found the same number of solutions. However, TC-BS found the solutions in the final seconds of search, coincidentally with the abrupt descent of iterations per second observed in figure 3. In a similar way, NAMOA-LEX-H0 found solutions more gradually but also in the final search stage, resulting in the second fastest alternative. NAMOA-LEX found

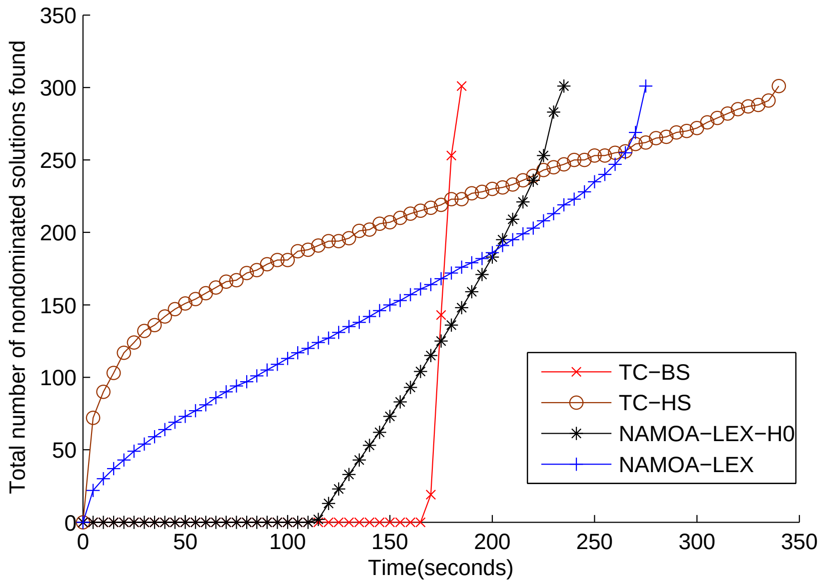


Fig. 4. Number of nondominated solutions found vs. time for a particular search problem.

solutions even faster and was therefore slower. Finally, TC-HS found solutions very quickly and was the slowest algorithm.

Figure 5 shows the time taken by each algorithm to find the *first* solution as a function of solution depth averaged for all problem sets. Again, TC-BS is the algorithm that starts to find solutions later, followed by NAMOA-LEX-H0. Both NAMOA-LEX and TC-HS find solutions very early and appear undistinguishable at this scale.

4 Conclusions and future work

The paper presents a general bicriterion search procedure that encompasses NAMOA* and Tung-Chew. This formal contribution highlights the differences between these algorithms and allows a clear comparison of different alternatives.

The effect of the heuristic functions proposed by Tung and Chew was evaluated empirically over four instances of the general search procedure (TC-HS, TC-BS, NAMOA-LEX, and NAMOA-LEX-H0). Several important conclusions can be drawn. In the first place, the experiments confirmed that the time needed to calculate the heuristics is not significant compared to total execution time. The heuristics were also found to improve the number of iterations, reducing the space requirements of the algorithms. The reduction of space requirements was similar in TC-HS (which used a combination of two heuristic functions) and NAMOA-LEX (which used only one of the heuristic functions).

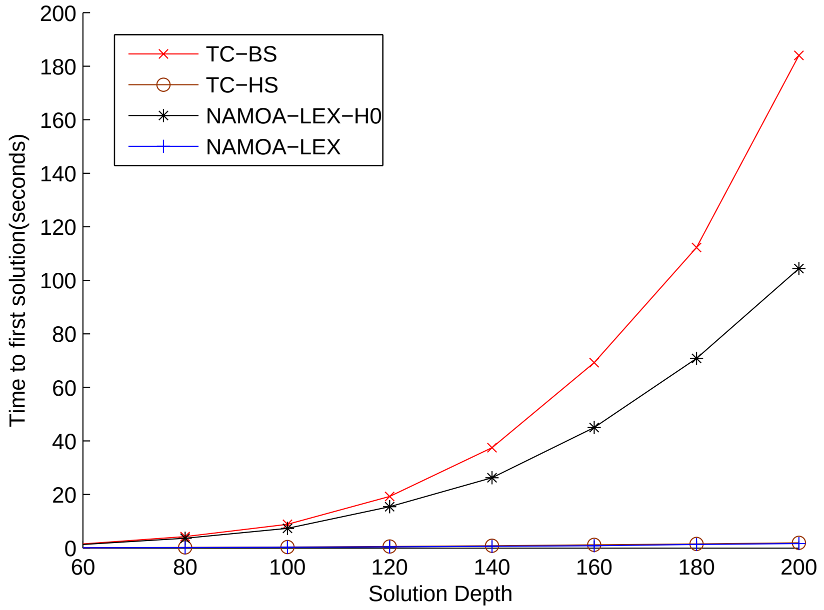


Fig. 5. Time requirements to reach the first solution averaged over all problem sets.

However, contrary to intuition, no time improvement could be observed in the algorithms with informed selection over the uninformed ones. The speed of the algorithms was found to be related to the discovery of nondominated solutions. Those algorithms that found solutions later in the search performed consistently faster. In this sense, the use of a specialized selection heuristic in TC-HS was found to work against the time efficiency of the algorithm.

The use of heuristics can effectively reduce the number of considered alternatives in bicriterion search algorithms. Most practical implementations of current algorithms perform linear or lexicographic orderings for label selection. The results presented in this paper suggest that the investigation of alternative orderings that combine heuristic search and delayed expansion of solutions could lead to more efficient algorithms.

References

1. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Systems Science and Cybernetics SSC-4 **2** (1968) 100–107
2. Pearl, J.: Heuristics. Addison-Wesley, Reading, Massachusetts (1984)
3. Dechter, R., Pearl, J.: Generalized best-first search strategies and the optimality of A*. Journal of the ACM **32**(3) (July 1985) 505–536
4. Clímaco, J.C.N., Craveirinha, J.M.F., Pascoal, M.M.B.: A bicriterion approach for routing problems in multimedia networks. Networks **41**(4) (2003) 206–220

5. Alechina, N., Logan, B.: State space search with prioritised soft constraints. Applied Intelligence **14**(3) (2001) 263–278
6. Gabrel, V., Vanderpooten, D.: Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite. European Journal of Operational Research **139** (2002) 533–542
7. Refanidis, I., Vlahavas, I.: Multiobjective heuristic state-space search. Artificial Intelligence **145** (2003) 1–32
8. Hansen, P.: Bicriterion path problems. In: Lecture Notes in Economics and Mathematical Systems 177, Springer (1979) 109–127
9. Müller-Hannemann, M., Weihe, K.: On the cardinality of the pareto set in bicriteria shortest path problems. Annals OR **147**(1) (2006) 269–286
10. Tung, C.T., Chew, K.L.: A multicriteria pareto-optimal path algorithm. European Journal of Operational Research **62** (1992) 203–209
11. Stewart, B.S., White, C.C.: Multiobjective A*. JACM **38**(4) (1991) 775–814
12. Mandow, L., Pérez de la Cruz, J.L.: A new approach to multiobjective A* search. In: Proc. of the XIX Int. Joint Conf. on Artificial Intelligence (IJCAI'05). (2005) 218–223
13. Mandow, L., Pérez de la Cruz, J.L.: Comparison of heuristics in multiobjective A* search. In: CAEPIA'05 -LNAI 4177, Springer (2006) 180–189
14. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik **1** (1959) 269–271

Bringing Relevant Topics to Foster Learning with the Aid of Ontologies

Elaine Harada Teixeira de Oliveira[1], Erika Handa Nozawa[1], Rosa Maria Vicari[2], Alberto Nogueira de Castro Júnior[1]

[1] Programa de Pós-Graduação em Informática, Universidade Federal do Amazonas (UFAM), Brazil
[2] Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul (UFRGS), Brazil
elaine@dcc.ufam.edu.br, erika.handa@gmail.com, rosa@inf.ufrgs.br, alberto@ufam.edu.br

Abstract. Ontologies are powerful structures that may help organize and share knowledge considering its contexts. This work addresses the issue of modeling the Japanese Language Proficiency Test (JLPT) domain through an ontology. The ontology was developed as an attempt to provide options for personalized learning oriented to context, based on some relevance principles. The most recent results come from the evaluation process in which the presented ontology was submitted to evaluators to assure its computationally consistency and adequacy to the Relevance Theory.

Keywords: Ontology, cognitive model, JLPT, Relevance Theory, context.

1 Introduction

Ontologies have been useful specifications to share conceptualization in a certain context [1]. They are especially useful to avoid misunderstandings and ambiguities. With them, experts may create a collaborative structure that may organize, share and make it a consensual knowledge. Ontologies are not restricted to specific area; they may be developed and applied to any area of knowledge.

The idea of using ontologies on the educational domain is not new, as we can see in [2, 3]. The use of domain ontologies for educational environments requires a semantic mapping of the knowledge from a taxonomy of formalized goals and concepts (information structure), developing a cognitive model that carries, implicitly, heuristics. For heuristics, we think on the understanding of the strategies (techniques) that comprise the individual who is guided by his pre-determined goals, purposes and values, taking into account what seems significant in order to get to new discoveries and learning.

For that, we have sought for systematic exploitation of finite semantic rules and properties of inferences of ontologies for the mining of relevant and guided contexts. This provides evidence to show the current state of the environment and allow the pedagogical agent to make decisions to positively support students in their study.

The proposed scheme in this work becomes interesting as it tries to develop more focused learning environments. Learning Environments, Learning Management Systems and Intelligent Tutoring Systems are some of the main interests in our research group as can be seen at [4, 5, 6]. Most times, contextual navigation is overloaded with information, and if not well directed may become boring or harmful to the student's goals. The pedagogical, animated and affective agent used for this work uses rules of perception and reaction to change emotional states such as sadness and joy, empathy and sympathy. Our research on pedagogical agents is also well established as it can be seen in [7, 8]. This work shows the process of learning from the integration of semantic-cognitive and affective domains, applied on the field of Artificial Intelligence.

For ontology, we mean an explicit specification of a shared conceptualization in a particular context. Our context is the Japanese Language Proficiency Test domain. In this case, the shared conceptualization models the necessary definitions and relations to enable people working in that domain to clearly communicate and exchange knowledge.

It is certainly useful and, most of the times, essential to have students paying attention to relevant information and studying relevant topics. We do not want students lost in a great amount of educational material, searching unnecessarily for relevant material, loosing time with irrelevant information. Considering this, our proposal is to bring relevant topics of certain subject to foster learning. This fostering of learning will be done with the aid and support of domain ontologies.

This scenario, validated within the multidisciplinary environment, presents interesting contributions, which motivated this work: a computational implementation of the formal representation of an ontology for JLPT opens possibilities for new services for the environment; the prospect of Maximum Relevance in the context of JLPT is essential, both to motivate the student to persist in his goal as to foster and direct his learning, important issues in proficiency courses. The use of pedagogical agent with an approach of affective tactics may strengthen the motivation of the student in the environment.

So the focus of this paper is to present the evaluation process of the conceived JLPT ontology. For now, it is our main result as it has taken close contact, interviews and collaboration with different domain experts, in a total of eight evaluators.

Next section presents the problem addressed in this work, the fostering of Japanese language proficiency. Section 3 explains briefly the Relevance Theory as part of the solution for our problem. Section 4 presents the ontology modeled to solve the issue of representing knowledge and some use cases. Section 5 presents the evaluation process achieved so far along with the evaluation obtained from different experts. Last section comes with some final considerations of this work with suggestions for future works.

2 Fostering Japanese Language Proficiency

As for other languages, Japanese language has a specific proficiency test, the Japanese Language Proficiency Test (JLPT). JLPT is held once a year in several large cities worldwide. This test aims to assess and certify the foreigner's proficiency level of knowledge of the Japanese language. The test is applied only to those whose Japanese is not the native language.

The test is divided into four different levels. The applicant shall submit to the test equivalent to his Japanese language level. In each level, the test is divided into three parts, so called subtests: writing/vocabulary, listening and text comprehension/grammar.

There is not an official public domain agenda for JLPT, but the characteristics of each subtest were verified through examples of previous tests, within the context of content and independent of JLPT level:

- Writing/vocabulary: evaluation in the field of reading and writing Japanese ideograms (*kanji*) as well as of vocabulary domain;
- Listening: evaluation on the ability of listening comprehension of spoken Japanese. Language tricks and expressions are common in this type of test;
- Text comprehension/grammar: evaluation in the field grammar, reading and text comprehension.

There are many preparatory courses for JLPT. The majority of these courses are offered in a traditional paper-based form and only a few of them are available online. In order to solve this problem, [9] proposed the eJLPT simulator, an educational hypermedia tool for students willing to practice their skills and knowledge of the Japanese language through an online simulation of JLPT.

eJLPT system has been utilized by a community of users interested in JLPT and some need for improvements and new functionalities, such as automatic scenarios, has become necessary. The main concerns were towards having some guidance through learning process, resources to adapt learning needs to JLPT context, special tools for teachers and students to avoid unnecessary effort on certain subjects which may not be important to get approved in JLPT.

To address these and other difficulties a new architecture for eJLPT is proposed. The existing system should not be presented as a unique hypermedia document for every user. It should be as flexible as an adaptive learning management system. In order to provide adaptativity of content and interface, one of the necessary issues to be considered is having an adaptation model which supports the user model. In this work, the user model is based on a cognitive perspective. The intention is to model the domain in a way that it should represent part of the cognitive context of an individual. Therefore, our approach aims to model the domain using an ontology based on a cognitive perspective, referred to as the Relevance Theory [10]. Another issue that needs to be considered is the adaptation mechanism. In our schema, this will be done by an intelligent pedagogical agent. Details of this mechanism will be treated in future works.

Next section focuses on the usage of the Relevance Theory as the support to understand, model and represent, in some way, cognitive contexts. It will also give directions on the attempt to achieve the maximum relevance of a certain subject according to pre-defined goals.

3 Relevance Theory

One of the keywords of this work is the word ‘context’. It may recall a lot of different definitions, ideas and meanings and may be used in almost all areas of knowledge. Many definitions of context may be found in the literature [11, 12, 13, 14, 15]. In Computer Science it is usually used in the area of Ubiquitous Computing, in which most of the usage of contextual information regards to circumstantial context. The research relies mostly on how to manage contextual information of the individual’s identity, his location and the moment in which the individual is interacting with the system. This is not the intention of this work neither the main concern of the adaptativity.

One of the first challenges is the attempt to model a knowledge domain to reflect, in some way, the representation of the cognitive context of a person over such domain. For cognitive context, we mean the set of assumptions used to interpret a statement and, for cognitive context of a person over a particular domain, we mean the person’s subset of assumptions on such domain.

Therefore, there is not the intention to model a circumstantial context neither the entire cognitive context of an individual, which would probably be impossible, but only those relevant to the domain of JLPT. An ontology was developed to represent this domain of JLPT. It consists of a class that represents the major context and subclasses representing subordinated contexts. In this case, the ontology plays both the role of representing the general area, acting as a course agenda, as something more specific, such as the representation of the knowledge already acquired by a particular individual. In this case, the ontology of the individual is a subset of the general ontology of the domain.

Besides the representation of the concepts, the ontology should allow navigation in a context from the perspective of the Relevance Theory. According to this theory, for any assumption to be relevant in a context, there should be connections among new assumptions and existing ones, which are already part of the context.

Relevance can be characterized in terms of contextual effects. To modify and improve a context is to have some effect in that context. There is no change in the context where the information is completely duplicated or when it is not related to any old information. There must be an interaction between old and new information. “The context used to process new assumptions is a subset of the old assumptions of an individual, with the new assumptions which combine to generate a variety of contextual effects” [10].

To ensure relevance, certain conditions must be met, which leads us to a comparative definition. An assumption is relevant in a context in the proportion of: 1) the increase of the contextual effects and, 2) the decrease of the effort required to process it in that context.

One of the main goals of this work is to meet the principle of maximum relevance by assessing the relationship of best cost-benefit between the contextual effect and processing effort.

Considering the graphical representation of ontology, assuming that we are in the original node, the more we move towards the children nodes, the greater the contextual effect obtained. To have this move forward, and consequently, the contextual effect, the move should occur by the links between nodes. Each move is valued in accordance to the specified value in each link in the ontology modeling.

The processing effort, similarly, also increases as we move through the graph. As the value of contextual effect, the value of the effort of processing is not an absolute value, does not have a unit of measure, but is a relative value, for comparative purposes, given by experts who, based on domain principles and parameters, can identify topics with greater weight and importance to the learning of subsequent topics.

Currently, the values for each move for both contextual effect and processing effort are unitary values. In the future, we intend to attach values to the links in accordance to principles and parameters of second language learning.

Next section will present the fundamentals of ontology, specially the application of this knowledge conceptualization in education and the concepts of the JLPT ontology.

4 JLPT Ontology

It is important to evidence that there is no official and available public agenda neither from Japan Foundation[1] nor Japan Educational Exchanges and Services[2], entities responsible for the administration, organization and dissemination of JLPT.

Once it was decided that an ontology would be a good solution for our problem, we started a search on the main ontology repositories (Swoogle[3], Protégé[4], DAML[5], Tones[6]). We've found some ontologies on Japanese Language, but none of them was complete or related to JLPT.

Therefore, the concepts for the development of this ontology were obtained through the consulting on the grammar adopted by Japanese school books, available preparatory courses and interviews with an expert, a Japanese language teacher for JLPT certification. The presented ontology is, then, an agenda, result of the compilation of several materials related to the Japanese language grammar because it expresses and defines the rules for a language as well as for the vocabulary. Figure 1 represents the class 'Grammar' and its *is-a* relations.

Japanese language is rich in vocabulary and expressions. Japanese grammar clearly defines its elements and rules essential to the Japanese. Gobi is an example that we can cite as trivial. Trivial in natural language, from the perspective on human reasoning. But how to express and make Gobi instances to be automatically computationally identified as trivial? Gobi is our use case and its use is explained further. One of the possibilities is to use description logic to facilitate the representation of knowledge based on logic.

The main characteristic of description logics is that concepts (or classes) can be defined in terms of descriptions. These descriptions specify the properties whose

[1] http://momo.jpf.go.jp/jlpt/overseas/index_en.html
[2] http://www.jees.or.jp/jlpt/en
[3] http://swoogle.umbc.edu
[4] http://protegewiki.stanford.edu/index.php/Protege_Ontology_Library#OWL_ontologies
[5] http://www.daml.org/ontologies
[6] http://owl.cs.manchester.ac.uk/repository/browser

objects must satisfy the domain of the concept. The language to be used shall allow the construction of descriptions of composition, including restrictions on binary relations that connect objects. This language defines a set of instances with syntax, to build the description, and semantics, representing concepts in multiple hierarchies, organizes in a structure of subconcepts, or taxonomy. Such formalization allows identifying attributes such as multiple inheritance, restriction of values, limits and also roles (transitivity, inversion, etc.).

In first order logic, the predicate is a language feature that can be used to create a statement or give a property. Thus, properties and instances (individuals) can be represented in predicate logic. Predicate logic has expressiveness from existential (at least some individual/body) and universal (all subjects/body) quantifiers. A quantifier is a logical symbol that quantifies the instances (individuals).

Next, we present part of the taxonomy of JLPT ontology, the Grammar class. Just after, we present some use cases from the stage of formalization of the ontology.

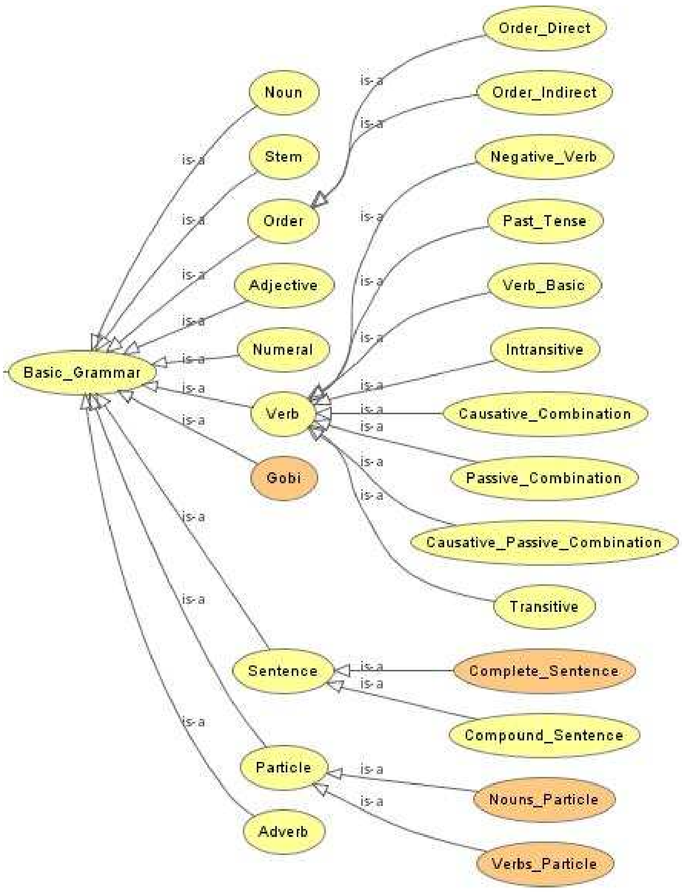


Fig. 1. Classes of JLPT ontology.

Use case: Gobi class

The Gobi class describes a common and useful resource in sentence endings. Gobi is very used in the Japanese language. It literally means “language tail” and it simply refers to anything that comes at the end of a sentence or a word, such as a verb (the minimum of a complete sentence in Japanese language). Most common Gobi usage is presented in this class. Some examples: yo-gobi, ne-gobi, yone.

Necessary and sufficient condition:

Instances (Individuals) are inferred as Gobi instances if they are instances of the subclass Basic_Grammar:

$$\text{Gobi} \equiv \text{Basic\_Grammar}$$

A necessary condition can be identified in this class:

Gobi is a subclass of Basic_Grammar:

$$\text{Gobi} \subseteq \text{Basic\_Grammar}$$

Instances of the mapped classes allow the inference of relevant contexts to JLPT agenda, from the Japanese Grammar formally expressed as:

$$\text{YO-GOBI} \subseteq \text{Gobi} \therefore \text{YO-GOBI} \equiv \text{Basic\_Grammar}$$
$$\text{NE-GOBI} \subseteq \text{Gobi} \therefore \text{NE-GOBI} \equiv \text{Basic\_Grammar}$$

Use case: Verb class and its subclasses

Verb class covers the basic properties of verbs. In Japanese language, verbs can be divided in:

- Intransitive Verb
- Causative Verb
- Causative Passive Verb
- Passive Verb
- Transitive Verb

Some necessary conditions may be identified in this class:

Verbs are **Basic_Grammar** subclasses:

$$\text{Verbs} \subseteq \text{Basic\_Grammar}$$

Instances of **Verbs** class relates to each other through **hasParts** property with at least one instance of **Particle** class:

$$\text{Verbs} \subseteq \exists\ \text{hasParts.Particle}$$

Instances of **Verbs** class relates to each other through **hasParts** property with at least one instance of **Gobi** class:

$$\text{Verbs} \subseteq \exists\ \text{hasParts.Gobi}$$

Instances of **Verbs** class relates to each other through **isPartOf** property with at least one instance of **Expression** class:

$$\text{Verbs} \subseteq \exists\ \text{isPartOf.Expression}$$

Instances of **Verbs** class relates to each other through **isPartOf** property with at least one instance of **Sentence** class:

$$\text{Verbs} \subseteq \exists \text{ isPartOf.Sentence}$$

Instances of **Verbs** class relates to each other through **isPartOf** property with at least one instance of **Essential_Grammar** class:

$$\text{Verbs} \subseteq \exists \text{ isPartOf.Essential\_Grammar}$$

The types of verbs used in Japanese Grammar classification (Intransitive Verb, Causative Verb, Causative Passive Verb, Passive Verb) are subclasses of **Verb** class. As consequence they inherit its properties and necessary condition.

Intransitive_Verb are **Verbs** subclasses:.

$$\text{Intransitive\_Verb} \subseteq \text{Verbs}$$

If there a instance of **Instransitive_Verb**, it consequently cannot be a instance of **Causative Verb**, **Causative Passive_Verb** or **Passive_Verb**. Formally, this can be expressed using disjoin:

$$\text{Intransitive\_Verb} \subseteq (\neg\text{Causative}_{\text{Verb}}, \neg\text{Causative}_{\text{Passive Verb}}, \neg\text{Passive}_{\text{Verb}})$$

Some properties used to specify the classes:

hasParts is an inverse property of object of **isPartOf**, which domain and codomain is the class **Context**.

$$\text{hastParts} \in P_0$$

$$\text{hasParts} \equiv \text{isPartOf}^{-R} \subseteq \forall\text{hasParts.Particle} \;\; \therefore\; R \subseteq \forall \text{ (hasParts.Particle}$$

$$\text{hasParts} \equiv \text{isPartOf}^{-R} \subseteq \forall\text{hasParts.Gobi} \;\; \therefore\; R \subseteq \forall \text{ (hasParts.Gobi)}$$

$$\text{hasParts} \equiv \text{isPartOf}^{-R} \subseteq \forall\text{hasParts.Verb} \;\; \therefore\; R \subseteq \forall \text{ (hasParts.Verb)}$$

$$\text{hasParts} \equiv \text{isPartOf}^{-R} \subseteq \forall\text{hasParts.Expression} \;\; \therefore\; R \subseteq \forall \text{ (hasParts.Expression)}$$

$$\text{hasParts} \equiv \text{isPartOf}^{-R} \subseteq \forall\text{hasParts.Sentence} \;\; \therefore\; R \subseteq \forall \text{ (hasParts.Sentence)}$$

$$\text{hasParts} \equiv \text{isPartOf}^{-R} \subseteq \forall\text{hasParts.Basic\_Grammar}$$

$$\therefore\; R \subseteq \forall \text{ (hasParts.Basic\_Grammar)}$$

Use case: Complete_Sentence class

A complete sentence is a specific sentence in Japanese grammar. It has, at least, one verb.

At least one necessary condition can be identified in this class:

Complete_Sentence is a subclass of Sentence:

$$\text{Complete\_Sentence} \subseteq \text{Sentence}$$

Some necessary and sufficient conditions to identify Complete_Sentence class can be expressed:

Instances of **Complete_Sentence** classes are related through **hasVerb** property with at least one instance of the class **Verb**:

$$\text{Complete\_Sentence} \subseteq \exists \text{ hasVerb.Verb}$$

One instance of **Complete_Sentence** relates to the the property **hasVerb** at least once:

$$\text{Complete\_Sentence} \subseteq (\geq 1\ \text{hasVerb})$$

As class **Complete_Sentence** is a subclass of **Sentence**, it <u>inherits</u> from **Sentence**:

$$\text{Complete\_Sentence} \subseteq \exists\ \text{hasOrderDirect.OrderDirect}$$
$$\text{Complete\_Sentence} \subseteq \exists\ \text{hasOrderIndirect.OrderIndirect}$$

Some properties in this class were defined and are used:

hasVerb is a property of functional objects, which domain is the **Sentence** class and the codomain is the **Verb** class:

$$\text{hastVerb} \in P_0$$
$$R \subseteq (\leq 1\ \text{hasVerb})$$
$$R \subseteq \forall\ (\text{hasVerb.Sentence}) \therefore R \subseteq \forall\ (\text{hasParts}^{-}.\ \text{Verb})$$

In this case, the ontology plays both the role of representing the general domain, acting like a course agenda, as something more specific, such as the representation of the knowledge already acquired by a particular individual. In this case, the ontology of the individual is a subset of the general ontology of the domain. The shared attribute of the conceptualization also lead to a validation process which is explained on Section 5.

Finally, we refer to explicit specification as the formal language used to develop the ontology computationally. We adopted OWL (Web Ontology Language) as the formal language, since it is developed by the World Wide Web Consortium and is a de-facto standard for building web ontologies. Research on OWL and its inference properties lead to several implementations of fast and reliable reasoners. The availability of this kind of tool is important since the ontology is the base to our adaptive learning environment.

5 Evaluation Process

So far, there has been a partial evaluation process, accomplished in three stages. First the ontology was evaluated by four experts on the domain, through interviews. Their expertise is on Japanese language proficiency tests. This stage of the process helped to achieve the “shared” property of the ontology. It demonstrated that the ontology is somehow reflecting the common body of knowledge of the Japanese proficiency domain.

We are aware that more experts must evaluate the ontology, but this is part of the work evolution. As the ontology and our adaptive system are made public, we will be able to gather more feedback and improve the conceptualization.

Secondly, we based the evaluation stage on the work of [16]. That work defines and formalizes several metrics to evaluate the ontology from quantitative and qualitative perspectives. The definition is represented with a meta-ontology and the formalization establishes the mathematical formulae to calculate the quantitative part.

Three ontology experts have evaluated JLPT ontology using modularity, depth, breath and accuracy parameters. From this partial evaluation, the ontology is valid. Our next step is to evaluate the ontology against the full set of parameters defined in [16]. At this point we used more quantitative aspects since we are more concerned with the practical utilization of the ontology. The pragmatics aspect is related to the fulfillment of the requirements established at the specification stage with support from the utilization scenarios (more end-user-related issues).

Finally, the assessment of the extent to which the ontological knowledge base reflects the constructs prepared by the experts in the domain, is, in essence, the assessment of a conceptual ontology.

The approach adopted was the one proposed by [16], known as assessment of the functional dimension of an ontology, with respect to its main purpose, i.e., the specification of a particular conceptualization or contextual assumptions on the domain or area of interest. According to [16] such specifications are always approximations, because the relationship between ontology and conceptualization (cognitive semantics) always depends on what the rational agent has conceived and on the semantic space that formally codifies it (formal semantics). Thus, according to the authors the functional assessment should focus on examining how these dependencies are implemented considering the ontology as a language that includes the object information and the required conceptualization.

In this context, [16] suggest evaluating the expressiveness of the ontology, with a computational perspective, according to its suitability and fitness to the concepts and relationships present in a referenced body of knowledge, or with experts of the involved areas. Regarding expressiveness, the ontology was evaluated in two phases, with experts in ontology engineering and cognitive science.

The ontology was formalized through description logic, which was implemented through the standard language called OWL DL (Web Ontology Language – Description Logic), developed by the World Consortium. Thus, since OWL was used for the development we were able to use two reasoners that logically verify OWL ontologies. The purpose of a reasoner is beyond this simple evaluation, but this is the part that better fits this stage of the work. Logical consistency tests were performed with the Pellet and RACER reasoners, and the results, analyzed by ontology experts. Both of them generated the same results confirming the validity of the ontology. Refinements in the ontology related to naming standards, modularity and connectivity between classes were identified and considered.

It should also be noted that the ontology was developed according to the concepts of the JLPT, using prior proficiency tests, grammar books and preparatory courses material widely recognized in the area, confirming the semantics of the ontology on JLPT domain considering the particularities of the Japanese language.

After validating the scenarios established in the use cases, through the symbolic description represented by the definitions of classes and instances in OWL-DL language, a validation of the domain ontology from the perspective of the Relevance Theory was accomplished. This was done through a document presenting the problem, the chosen approach and the proposed solution, with a questionnaire at the end, given to the cognitive science expert. In this step, the evaluation was done according to the glass-box method of functional dimension of [16]. The criteria were related to competency adequacy, such as compliance to expertise in the area, flexibility, precision and recall related topic with specificity. The result of the evaluation of the ontology was considered highly satisfactory, responding to the four basic points based on the Theory of Relevance: the possibility of mapping the cognitive domain for the JLPT, reflection of ontological notions of contextual effect and processing effort, and finally, the conclusion that the ontology can meet the principle of maximum relevance through navigation on the ontology.

Therefore, given the assessments made by experts, it was possible to demonstrate that the ontology is appropriate and promising for the domain and desired goals. Suggestions to extend the purpose of the ontology to attend to other aspects of the Relevance Theory were identified by the expert, and contribute to future works.

After the current phase of implementation and partial evaluation, the intention is to have a prototype in order to verify experimentally the optimistic responses given by the experts, to compare, in groups of users, the use of the previous environment against the environment with the new implementation and the adapted navigation through relevant ontologies.

6 Final Considerations

The development of an ontology is a very hard task by itself. It requires a lot of research, dedication and, above all, close contact to knowledge experts of the proposed domain. The development process is exhausting and requires many interviews, many back and forth in each stage.

Another challenge was to develop it according to a perspective of the Relevance Theory. In many ways, it should reflect the notion of cognitive contexts. It was very important to have an expert who could evaluate it.

As the validation process shows, it seems that, so far, we have achieved a mature state of the ontology. With it, we may feel more confident that the domain is well modeled in many aspects – computationally, conceptually and cognitively.

With the preliminary validation of the experts, the knowledge base of the system is formalized. The adaptation of content, through the implicit heuristics in the structures of the ontology, allows the pedagogical agent to support and assist the student. These characteristics, necessary for an ITS, through an approach of adaptation of relevant content with semantic taxonomy, are interesting expected contributions to the AI research area.

For future works, the intention is to develop ontologies services so it becomes a real semantic web service, as a semantic repository, with other domain ontologies. Given the explicit semantics of knowledge representation, it is possible to instantiate a web service provided with features such as: semantic repository, recommendation services from expert perspective, or even though, with the integration of intelligent agents, among others.

There is also the intention to do some formalization on the Relevance Theory in order to make replication possible. Besides that, there should be the evaluation with other domain experts and an analysis of the results.

Acknowlegments

This work was partially supported by FAPEAM, CAPES, CNPq, Brazil.

References

1. Schreiber, G.: Knowledge Engineering. In: Handbook of Knowledge Representation, Elsevier (2008)
2 Bittencourt, I.I., Isotani, S., Costa, E., Mizoguchi, R.: Research Directions on Semantic Web and Education. Journal Scientia - Interdisciplinary Studies in Computer Science, vol. 19, n. 1, pp. 59--66 (2008)
3. Mohammed, P.; Mohan, P.: Contextualizing Learning Objects Using Ontologies. In: Computational Intelligence, v. 23, n. 3, pp. 339--355. Blackwell Publishing (2007)
4. Viccari, R.M.; Flores, C.D.; Seixas, L.; Gluz, J.C.; Coelho, H.: AMPLIA: A Probabilistic Learning Environment. International Journal of Artificial Intelligence in Education, v. 18, n. 4 (2008)
5. Jaques, P.A.; Viccari, R.M.: A BDI approach to infer student's emotions in an intelligent learning environment. Computers & Education, Inglaterra, v. 49, pp. 360--384 (2007)
6. Viccari, R.M.; Gluz, J.C.: An Intelligent Tutoring System (ITS) view on AOSE. International Journal of Agent-Oriented Software Engineering (IJAOSE), v. 1, pp. 295--333 (2007)
7. Jaques, P.A.; Viccari, R.M.: A BDI Approach to Infer Student's Emotions. In: Ibero-American Conference on Artificial Intelligence (IBERAMIA), 2004, Puebla. Lecture Notes in Artificial Intelligence. Berlin: Springer-Verlag, v. 1, pp. 1--10 (2004)
8. Viccari, R.M. (Org.); Jaques, P.A. (Org.); Verdin, R. (Org.): Agent-Based Tutoring Systems by Cognitive and Affective Modeling. Hershey, PA: IGI Global, v. 1, 300 p. (2008)
9. Nozawa, E.H.; Oliveira, E.H.T.; Costa, M.L.F.; Santos, E.R.; Isotani, S.; Castro Júnior, A.C.: Towards an Adaptive Learning Environment Supported by Ontologies. In 9th IFIP WCCE – World Computers in Education, July 27–31 (2009)
10. Sperber, D.; Wilson, D.: Relevance: Communication and Cognition. Cambridge, MA: MIT Press (1986)
11. Chen, G.; Kotz, D.: A survey of context-aware mobile computing research. Tech. Rep. TR2000-381, Dartmouth, November (2000)
12. Dey, A.K.: Understanding and using context. Personal and Ubiquitous Computing, Special issue on Situated Interaction and Ubiquitous Computing, vol. 5, n. 1 (2001)
13. Schilit, B.N.; Adams, N.L.; Want, R.: Context-aware computing applications. In IEEE Workshop on Mobile Computing Systems and Applications (Santa Cruz, CA, US) (1994)
14. Schmidt, A.; Beigl, M.; Gellersen, H.-W.: There is more to context than location. Computers and Graphics, vol. 23, n. 6, pp. 893--901 (1999)
15. Strang, T.: Service Interoperability in Ubiquitous Computing Environments. PhD thesis, Ludwig-Maximilians-University Munich, Oct. (2003)
16. Gangemi, A.; Catenacci, C.; Ciaramita, M.; Lehmann, J.: Modeling Ontology Evaluation and Validation. Y. Sure (ed.), Proceedings of the Third European Semantic Web Conference, Springer (2006).

Chapter 7

IROBOT - Intelligent Robotics

Intelligent Robotic Mapping and Exploration with Converging Target Localization

João Certo[1,2], João Lobato Oliveira[1] and Luís Paulo Reis[1,2]

[1] FEUP – Faculty of Engineering of the University of Porto, Portugal;
[2] LIACC – Artificial Intelligence and Computer Science Lab., University of Porto, Portugal;
joao.certo@fe.up.pt, jmldso@gmail.com, lpreis@fe.up.pt

Abstract. This paper presents different methodologies for a maze solving agent in unknown environments. The developed methodologies include an odometry based localization system, a converging target localization based on the centre of mass of recursive triangulations, and a regressive obstacle distance function, along with an obstacle setting and sweeping mechanism for mapping. A quad-tree solution was used for map representation, and an innovative exploring A* for path planning was developed. A quasi-reactive, wall-following agent serves as a comparison basis and a simulation environment called ciber-rato was used to test the implemented architecture and validate the developed methods within gradually increasing difficult scenarios. Besides the reactive agent being typically less time consumptive, the exploring architecture granted greater consistency and robustness. The results also evinced the time-improvement of using the exploring A* and the efficiency of this target localization method as gradually converging to the real target position.

1 Introduction

Cyber-Mouse (Ciber-Rato) is a modality included in the "Micro-Rato" competition, directed to teams interested in the algorithmic issues and software control of mobile autonomous robots [1]. This modality is supported by a software environment, which simulates both robots and a labyrinth [2]. The Cyber-Mouse has a body diameter of 1 mouse unit (*Um*) and seven available sensors although only two, user selectable, can be used at any given time. The final purpose is to reach the cheese, identified by a ground sensor and detectable through a direction providing beacon sensor visible trough low walls. Mouse's performance is evaluated through success on reaching the cheese, the time it took and the number of collisions.

The cyber-mouse competition has been used, amongst other applications as a testbed for long-term planning [3], as a scenario for the detection and avoidance of dangerously shaped obstacles [4], or even as a tool for the teaching of Artificial Intelligence and Robotics [5]. In this paper we evaluate the problems of mapping, localization and path planning by building a deliberative agent that can find its way from a starting position to the target without prior knowledge of the maze. Ultimately this architecture's performance is compared to a reactive approach.

The paper structure is as follows. The next section discusses the problems of mapping and self-localization, navigation and path planning together with some related state of the art algorithms leading to the chosen approaches. Section 3 presents the developed methodologies. Section 4 contains a description of a comparative reactive agent, the testing environments and the respective results. Finally, section 5 concludes this paper and points to future work.

2 Robotic Mapping and Planning Overview

Mapping is the process of building an estimate of the metric map of the environment [6]. The mapping problem is generally regarded of most importance in the pursuit of building truly autonomous mobile robots, but still mapping unstructured, dynamic, or large-scale environments remains largely an open research problem.

Planning is the process of deciding which route to take based on and expressed in terms of the current internal representation of the terrain. Typically this process calculates the cost of each motion decision towards the target, based on a given heuristics, and chooses the “cheapest” one.

2.1 Mapping and Localization Problem

To acquire a map, robots must possess sensors that enable it to perceive the outside world. Sensors commonly brought to carry out this task include cameras; range finders (using sonar, laser or infrared technology), radars, tactile sensors, compasses, and GPS. However, all these sensors are subject to errors, often referred to as measurement noise, and to strict range limitations.

So, considering these issues several different challenges can arise to robotic mapping: statistically dependent sensors measurement noise, high dimensionality of the entities that are being mapped, data association problem (determining if sensor measurements taken at different points in time correspond to the same physical object), environments changing over time and robot exploration.

The motion commands issued during environment exploration also carry important information for building maps, since they convey information about the locations at which different sensor measurements were taken. Robot motion is also subject to errors and the controls alone are therefore insufficient to determine a robot’s pose (location and orientation) relative to its environment. If the robot’s pose was known all along, building a map would be quite simple. Conversely, if we already had a map of the environment, there are computationally elegant and efficient algorithms for determining the robot’s pose at any point in time. In combination, however, the problem is much harder.

Considering the map representation problem, which has a significant impact on robot control [7], we can account for three main methods: Free space maps (road mapping), as spatial graphs, including Voronoi diagrams, and generalised Voronoi diagrams; object maps; and composite maps (cell decomposition) as point grids, area grids and quad trees.

Virtually all state-of-the-art algorithms for robotic mapping in the literature are probabilistic. They all employ probabilistic models of the robot and its environment relying on probabilistic inference for turning sensor measurements into maps [6].

2.2 Navigation and Path Planning

In artificial intelligence, planning originally meant a search for a sequence of logical operators or actions that transform an initial world state into a desired goal state [8]. Robot motion planning focuses primarily on the translations and rotations required to navigate, considering dynamic aspects, such as uncertainties, differential constraints, modelling errors, and optimality. Trajectory planning usually refers to the problem of taking the solution from a robot motion planning algorithm and determining how to move along the solution in a way that respects the mechanical limitations of the robot.

The classic path planning problem is then finding a collision-free path from a start configuration to a goal configuration, in a reasonable amount of time, given the robot's body constitution and the map representation, as retrieved in the mapping process.

In an unknown environment the mapping and motion planning must be processed in parallel through exploration and dynamic navigation decisions. This structure requires plans updating. A natural way of updating plans is to first select a path based on the present knowledge, then move along that path for a short time while collecting new information, and re-planning the path based on new findings.

Considering the application many algorithms have been proposed for path planning: A and A Star (A*), Dijkstra, Best-First, Wavefront Expansion, Depth-First Search, Breadth-First Search.

3 Exploring Agent Methods and Architecture

The architecture of our exploring agent is presented in four independent modules, concerning the self-localization, target (goal) localization, mapping and navigation, and path planning problem. These modules were integrated to solve various mazes facing an unknown environment without any previous knowledge.

3.1 Self-Localization

The self-localization is based on the robots' odometry which is defined by a dynamic inertial movement model [9]. Due to Gaussian noise the simulator model induces a linear motion maximum error given by Eq. 1.

$$\delta \leq \frac{Max(MotorPow) * NoiseDeviation + MotorResolution/2}{Max(MotorPow)} \ (\%) \qquad (1)$$

As such, for each position estimate there is a maximum δ deviation for the Cartesian coordinates and 2*δ for the rotation angle. The simulator defines *Max(MotorPow)=0.15*, *NoiseDeviation=1.5%* and *MotorResolution=0.001*; which infers δ≈1.83% and a rotation error of 3.66%, acceptable for this application. In order to correct cumulative odometry rotation errors, the compass is read every 50 cycles, always accounting for the compass sensor latency of 4 cycles.

3.2 Target Localization– Centre of Mass from Recursive Triangulations

In order to define an initial plan of attack two methodologies were considered for marking the initial target position. If the beacon isn't visible a random function defines its disposition as being within any of the four map corners, changing every user-defined number of cycles (usually 500) to one corner different from the former. As soon as the beacon is visible an initial rough estimation of the target is calculated by intercepting the line defined by the mouse and the beacon points with the map bounds. From the resultant 4 points the initial target position is considered as being the convergent point (in the beacon direction – see Eq. 2 closer to the mouse.

A more accurate target position is recursively estimated by computing the centre of mass resultant from successive triangulations; as illustrated in Fig. 1. To achieve this, within every 25 cycles, if the beacon is visible, a line Y_A is traced along the target direction. In order to retrieve the real beacon direction, α, this process accounts for the beacon sensor's latency of 4 cycles. Each line is then intercepted with each of the former ones resulting in a conjunction of intercept points P_i (X_i, Y_i). In order to restrict these points to the ones converging to the target Eq. 2 was applied:

$$\begin{cases} P_i = P_i \; if\big((-90° < \theta < 90°) \; AND \; X_i > X_A\big) \\ P_i = P_i \; if\big((-180° < \theta < -90° \; OR \; 90° < \theta < 180°) \; AND \; X_i < X_A\big) \\ else \; Remove \; Pi \end{cases} \quad (2)$$

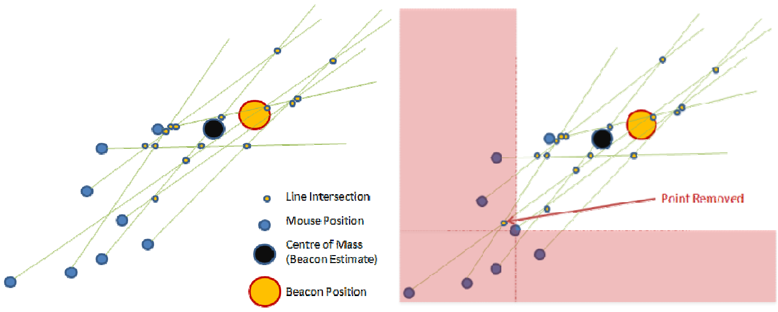


Fig. 1. Target localization from recursive triangulations (left to right): method overview **(a)**; removing intercept points displaced from the mouse-target square, at a mouse position **(b)**.

All the points outside the map bounds are also filtered. After having 5 valid intercept points, while the mouse moves towards the estimated target, every point displaced from the mouse-beacon square (see Fig. 1b), along with the ones distancing more than 3*Um* from the current target estimate, are removed from the list.

$$\begin{cases} X_C = \left(\sum_{i=0}^{N_{points}} X_i\right) / N_{points} \\ Y_C = \left(\sum_{i=0}^{N_{points}} Y_i\right) / N_{points} \end{cases} \quad (3)$$

Finally the target position, given by Eq. 3, is calculated as the centre of mass of the intercept points in the given time. Whenever a new line, Y_A, is traced the target position is re-calculated, granting a robust target estimation, which is enhanced as the mouse moves towards the target.

3.3 Mapping

Our mapping algorithm is based on a deterministic model representing the distance to an obstacle given the obstacle sensor value. The map is represented using a multi-resolution quad-tree decomposition since this representation grants good performance for this application, with low processing cost.

In the presence of an obstacle we used a quad-tree gridding to subdivide each of the obstacle cells. Our quad-tree strategy uses an adaptive division depth to a deepest cell size (granularity) of 0.1*Um*. Fig. 2 illustrates different granularities for different known maps.

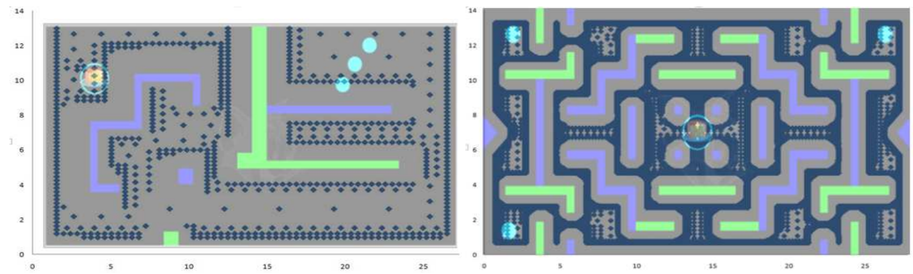


Fig. 2. Real map overlapped with quad-tree representation (left to right): RTSS06Final with 0.7*Um* depth **(a)**; 2005Final with 0.1*Um* depth **(b)**.

3.3.1 Obstacle Detection

The navigation and consequent mapping is based on the obstacles disposition along the map. To calculate the robot's distance to an obstacle relative to its sensor values, a series of successive experimental measurements were taken as seen in Fig. 3.

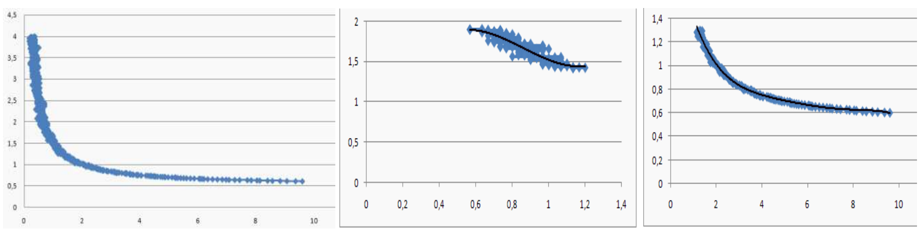


Fig. 3. Obstacle distance distribution – obstacle sensor values in horizontal axis (units); obstacle distance in vertical axis (mouse units) (from left to right): the full distribution **(a)**; distribution for sensor values ranging from 0.9 to 1.0 **(b)**; distribution for sensor values ranging from 1.1 to 4.5 **(c)**.

Through linear regression it was possible to obtain the following equation (Eq. 4) where *d* is the distance and *x* the sensor values.

$$\begin{cases} d = 4.1981x^3 - 10.84x^2 + 8.1978x - 0.0403, \quad if\ 0.9 < x < 1.0 \\ d = -0.0001x^5 + 0.0046x^4 - 0.0561x^3 + 0.3435x^2 - 1.095x + 2.2027, \\ \quad if\ 1.1 < x < 4.6 \\ d = 0.6, \quad if\ x > 4.6 \end{cases} \tag{4}$$

These functions estimate the obstacle distance (from the mouse's body centre) with a maximum error, δ, of 0.213Um for sensor values in the range of 0.9-1.0, and 0.189Um for values in the 1.1-4.5 range.

3.3.2 Obstacle Setting and Sweeping

The sensor available in the cyber-mouse simulation system has an aperture angle of 60º, as depicted in Fig. 4a.

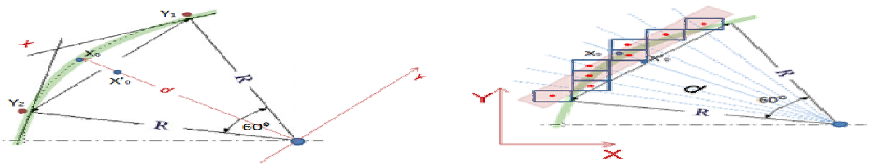


Fig. 4. Obstacle sensor coverage **(a)**; quad-tree cells identified as obstacles **(b)**.

As the furthest detectable obstacle has a distance of approximate 1.6Um (derived from Eq. 4) for the maximum *x* equal to 4.5Um. This corresponds to a possible obstacle between Y_1 and Y_2 (Fig. 4) of 1.6Um. As the detected obstacle can be in any part of that space, a set of cells between Y_1 and Y_2 are marked as an obstacle. So, in order to assure that all cells are marked (depicted in Fig. 4b) a set of points is generated. The points are placed along the arc defined by the distance to obstacle minus a user defined safety distance (at least 0.5Um for the mouse body, but usually 0.65Um). As the number of points depend both on this distance and the deepest cell maximum width (*w*), the number of marked points is given by Eq. 5:

$$N(d,w) = \begin{cases} \frac{\pi}{3} \Big/ \sin^{-1}\left(\frac{w/2}{d}\right), & \frac{w/2}{d} < 1 \\ 1, & \frac{w/2}{d} \geq 1 \end{cases} \tag{5}$$

The previous method can lead to cells wrongly marked as obstacles. In order to solve this problem a sweeping mechanism was developed to clean blocked (obstacle) cells making them passable again. This mechanism is very similar to the obstacle setting mechanism with the differences of marking passable cells (instead of obstacles), fixating *d* (user definable) and restricting the aperture of sweeping to half (30º). These restrictions were made so that the robot wouldn't be cyclically marking the same cells as blocked or passable.

An exemplar resultant internal map representation of the robot is shown in Fig. 5.

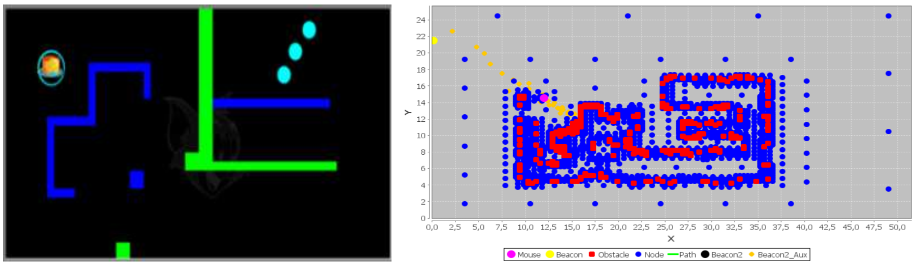


Fig. 5. Robot internal map after solving a maze.

3.4. Path Planning - Exploring A*

To find a path between a previously computed objective and the agent, the A-Star (A*) algorithm is used. This decision was made by balancing implementation cost with a guarantee of a solution. This algorithm is implemented over the quad-tree map representation, defining the shortest path towards the target by marking waypoints in the correspondent map cell centres.

Eq. 6 represents the cost between the source point and the target point, which passes through node n.

$$f(n) = g(n) + h(n) \quad (6)$$

Here *g(n)* is the real cost from the source to node *n*, and *h(n)* is the estimated cost between node *n* and the target. *f(n)* is the total cost of the path that passes through node *n*. The customary heuristic function would be the Euclidean distance between the source and the target position. However, when solely using the Euclidian distance as the cost for reaching a target, the robot would spend a great amount of time mapping around the same obstacle. In order to solve this problem, a different heuristic was developed where the waypoint cost is the Euclidian distance affected by the quad-tree cell depth and a weight (*f*), as defined in Eq. 7.

$$Cost\ (A, B) = Dist(A, B) * Depth(B)^f, f\ \geq 1 \quad (7)$$

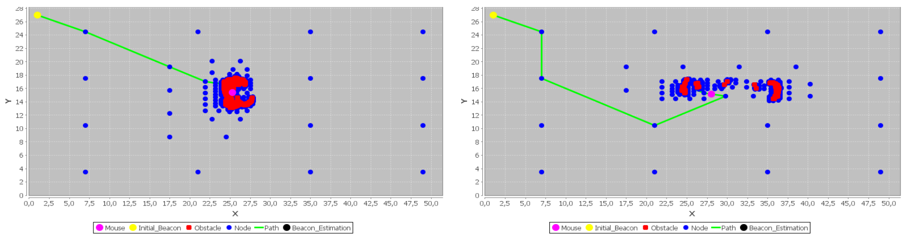


Fig. 6. Comparison of exploring algorithms: classic A*(**a**); Exploring A* (**b**).

This method assures that the mouse explores the map, towards the beacon, preferably following big cells. Consequently it increases the exploring step, as can be seen in Fig. 6b, and the mouse's velocity. The guarantee of solution given by the classic A* is maintained as the exploring A* would revert to passing trough small cells after exploring the larger cells.

3.5 Navigation and Control

The robot is controlled by following each waypoint centre given by the A* algorithm towards the target. The robot rotates to each waypoint centre and accelerates in that direction. The waypoint centre is considered reached if the robot's coordinate values are within a certain error margin of that centre.

The agent navigation speeds are dynamically adjusted and are dependent on several factors. Simple control optimizations include a speed increase in rotation if there are big differences between the current angle and the waypoint direction or increasing speed if a waypoint is far away. More advanced implemented speed optimizations take into account subsequent waypoints and their relative direction in order to further increase the mouse' performance.

3.6 Visualization System

In order to visualize the current internal robot's map representation, in real-time, a visualization system was developed. This interface, shown in Fig. 7, was designed upon the Java open-source JFreeChart API [10]. Its zooming capabilities, as well as its Cartesian disposition, extend this system to debugging purposes. As observable, 5 different series, with different colours, were designed; each representing an individual estimation: pink for the mouse position, yellow for the initial (attack) beacon position, red for the obstacle's cell centres, blue for the passable cell centres, green for the planned path, and black for the current beacon estimation.

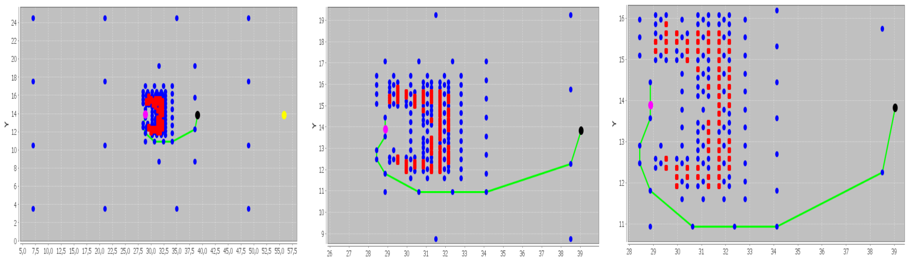


Fig. 7. Visualization system for different, incremental, zoom levels.

4. Experiments and Results

In order to test and validate the proposed approach, this section is divided in three: a presentation of a reactive agent architecture serving as a comparative base, a description of different evaluation scenarios and a presentation of the results.

4.1 Reactive Agent Architecture

Although this quasi-reactive (wall-following) architecture isn't the focus of this paper, it is well suited to serve as a comparative base for results evaluation. The designed reactive agent consists on a behaviour-based state-machine.

When the simulation starts, the robot begins with the state *Find Beacon*, rotating the robot around itself until it finds the beacon, or walking randomly until a wall is found. When the beacon is found, the robot changes its state to *Follow Beacon* and goes forward until it reaches the ground beacon area or finds a wall. If the beacon is reached, the state changes to *Beacon Area Reached* and the simulation ends. On the other hand, if a wall is found instead, the robot changes state to *Change Direction*, rotating itself to the side which has no detectable walls. Once the robot stops detecting a wall directly in front, it changes to the state *Follow Wall*. On this state the robot simply goes forward until it stops detecting the side wall. When it stops detecting the side wall, or detects another wall in front it changes back to the *Find Beacon* state.

Besides the state, two additional non-reactive elements were included: the time since the mouse was near a wall and a memory of the relative side of the wall being followed. The notion of time allowed the mouse to wander randomly to a wall when no beacon is found (high-walls) and to maintain a direction for a short time after leaving the wall. Remembering the wall being followed allowed the avoidance of a problem where the robot would cyclically alternate between close opposing walls.

4.2 Evaluation Scenarios

In order to evaluate each experiment the following, gradually increasing difficult scenarios, were chosen.

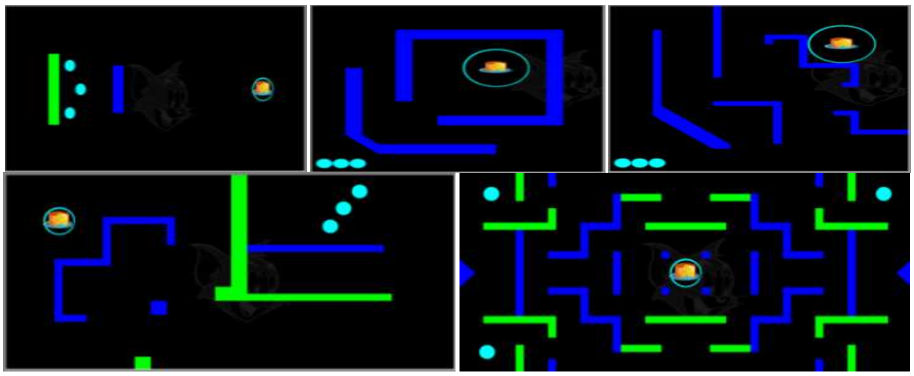


Fig. 8. Evaluation maps (from top-left to bottom-right): Basic **(a)**; MicRato98 **(b)**; 2001Final **(c)**; RTSS06Final **(d)**; 2005Final **(e)**.

As can be seen in Fig. 8: the *Basic* map has a small wall between the mouse and the beacon (a); *MicRato98* is an easy map with only low walls (b); *2001Final*, a medium difficulty map with only low walls (c); *RTSS06Final*, a hard map with low and high walls (d); *2005Final*, a very hard map with low and high walls (e).

The evaluation was done by observing if the mouse reached the cheese or not and the time it took to do it. For the exploring agent, since collisions impose errors in the self-localization procedure which would make the robot fail the waypoints (given by A*) towards the target, the number of collisions weren't considered. Additionally, when relevant, an observational description of the mouse' behaviour during the experiment may be included to further evaluate and compare the approaches.

4.3 Results

Each map was tested with the two described maze solving experiments. For results comparison, in both experiments the tests were made with two self-localization systems: through odometry measurement (see 3.1) and with GPS. Different deepest cell's maximum sizes (considered in the quad-tree decomposition) were used: a fixed resolution of 0.1*Um* that guarantees map solving (for the maximum 1.5*Um* obstacle distance) and one variable, granting the best performance for each map. Since the simulator adds some noise in the sensors and actuators, three different runs for each map and agent were performed. As such, conclusions can be made from averaging the results and thus overcoming the stochastic nature of the simulator.

4.3.1. Maze solving in an Unknown Map - Reactive Agent Evaluation

In this experiment we tested our (quasi-)reactive agent, *Smart-Follower*, for paradigm comparison. The results are shown in Table 1.

Table 1. Experimental results for the *Smart-Follower* agent.

Follower	Experiments									
	1		2		3		Average			Observations
	Time	Collisions	Time	Collisions	Time	Collisions	Successful Exp.	Time	Collisions	
Map1 - Basic	1260	0	228	0	1244	0	3	911	0	NA
Map2 - MicRato98	950	0	450	1	886	0	3	762	0	NA
Map3 - 2001Final	638	0	790	0	NA	NA	2	714	0	Enclosure Conflict
Map4 - RTSS06Final	NA	NA	1366	0	NA	NA	1	1366	0	Wall-Beacon Conflict
Map5 - 2005Final	NA	NA	NA	NA	NA	NA	0	NA	NA	Wall-Beacon Conflict

The *enclosure* conflict happens when the mouse is surrounded by obstacles very close to each other. Although the side sensors detect an obstacle there was enough room for the mouse to pass. The *wall-beacon* happens when the mouse is near a U-shaped wall. At each wall end the mouse turns towards the beacon coincident with the obstacle centre. As such, the mouse follows the same wall in the opposite direction.

4.3.2 Exploring Agent Evaluation

The following Table 2 then presents the exploring agent performance in resolving the proposed evaluating scenarios, for the self-localization and GPS methods. In order to visually depict the agent's deliberations, namely its mapping, planning and target estimation abilities, while solving some of the tested mazes (and as proof of concept), Fig. 9 presents some screen shots of a successful experiment.

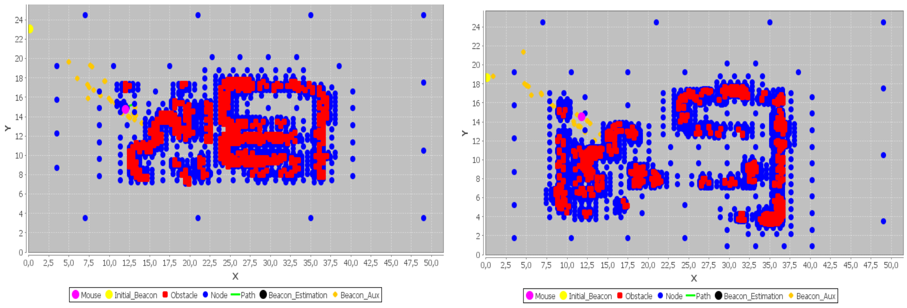


Fig. 9. Exploring agent runs on RTSS06Final map (from left to right): GPS **(a)**; Odometry **(b)**.

Table 2. Exploring Agent Map Results.

Unknown Map		Experiments 1 Time	2 Time	3 Time	Average Time	Success. Exp.	Observations
Map1 - Basic	GPS	260	658	562	493	3	NA
	Odometry	260	364	682	435	3	NA
Map2 - MicRato98	GPS	2544	NA	2634	2589	2	Corner Collision
	Odometry	NA	2328	NA	2328	1	Small Obstacles' Aperture
Map3 - 2001Final	GPS	3432	2598	NA	3015	2	Corner Collision
	Odometry	NA	NA	4266	4266	1	Corner Collision
Map4 - RTSS06Final	GPS	10234	7322	NA	8778	2	Corner Collision
	Odometry	5344	5546	6954	5948	3	NA
Map5 - 2005Final	GPS	NA	NA	NA	NA	0	Small Obstacles' Aperture
	Odometry	NA	NA	NA	NA	0	Small Obstacles' Aperture

In this experiment the corner collision is self-explanatory as representing the collisions at the obstacles' corners level. We found some problems at solving the *Final2005* and the *MicRato98* maps due to the distance between walls, which is kept at the lowest margin (1.5*Um*) in many situations, blocking the mouse, due to its mapping methodology limitations.

5. Conclusions and Future Work

As observable in the quasi-reactive experiment (4.3.1), approaches featuring some deliberations can quite effectively resolve most of the simpler maps (first 3 in Table 1) and situations with simple algorithms.

During the simulations, the localization method gradually converges to the real target position as the new measurements add points clustering around the beacon, resulting in a better centre of mass that gives a better beacon estimative.

The initial approach of using classic A* was too time consuming to be practical. As an example, comparing to the worst time from Table 2 in Map 4 (10234 cycles), the agent using classic A* was still very far from the target, due to navigation and exploration very close to obstacles. Nevertheless we believe that classic A* would eventually allow the agent to reach the beacon. As such, using the exploring A* with an *f* factor of 1.2 (trial-error adjust) lead to a great time improvement. Yet, in worst case scenarios, with large clusters of small obstacles, the exploring A* can take longer but still guarantees a solution.

In comparison to the reactive agent, the exploring mouse evinced better map solving capabilities. In terms of consistency, results with the exploring agent were more favourable as it was capable of repeatedly solving the same maps (*Basic*, *2006Final* and *2001Final* – Table 2), contradicting the reactive mouse's behaviour (*Basic* and *MicRato98* in Table 1). This was possible due to the exploring agent's abilities to bypass U-shaped obstacles and recognizing its position, thus leaving already explored areas.

As a limitation, for the maps that the reactive agent could solve, the time taken for the exploring agent to conclude each map (except for the Basic map) was greater than with the reactive implementation. This was to be expected as map navigation (cell marking) isn't as effective as sensor navigation (wall following).

Within the reactive agent when comparing odometry to GPS navigation (Table 2) we conclude that the GPS's had a superior rate of success but odometry was more time effective. Besides the less need for adjustments to reach the cell centre, the reason for time effectiveness of odometry is the small error in obstacle detection that eliminates some hysteresis caused by setting and sweeping obstacles.

In the future the use of both lateral proximity sensors along with a probabilistic model for obstacles' detection should greatly improve the mapping efficiency. Parameters like the exploring weight, sensor apertures for marking and cleaning obstacles, distance for cleaning obstacles can be set to optimal by using reinforcement learning mechanisms. The maximum cell depth can also be dynamically adjusted using a greedy algorithm, like hill-climbing, in order to improve performance.

References

[1] Almeida, L., Fonseca, P., Azevedo, J.L.: The Micro-Rato Contest: a popular approach to improve self-study in electronics and computer science. SMC'2000, IEEE Int. Conference on Systems, Man and Cybernetics, Vol. 1, Nashville, USA (2000) 701 – 705.

[2] Lau, N., Pereira, A., Melo, A., Neves, A., Figueiredo, J.: Ciber-Rato: Um Ambiente de Simulação de Robots Móveis e Autónomos. Revista do DETUA 3 (2002) 647 - 650.

[3] Ribeiro, P.: YAM (Yet Another Mouse) - Um Robot Virtual com Planeamento de Caminho a Longo Prazo. Revista do DETUA 3 (2002) 672-674

[4] Luís, P., Martins, B., Almeida, P., Silva, V.: Detecção de Configurações de Obstáculos Perigosas: Aplicação no Robô EnCuRRalado. Revista do DETUA 3 (2002) 659-661.

[5] Reis, L.P.: Ciber-FEUP - Um Agente para Utilizar o Simulador Ciber-Rato no Ensino da Inteligência Artificial e Robótica Inteligente. Revista do DETUA 3 (2002) 655-658.

[6] Thrun, S. – Robotic Mapping: A Survey CMU-CS-02-111 (2002).

[7] Murphy, R.R.: An Introduction to AI Robotics. Bradford Book, MIT Press : Cambridge, Massachussets, London England (2000)

[8] LaValle, S.: Planning Algorithms. Cambridge University Press, 2006.

[9] Lau, N., CiberRato 2008 Rules and Technical Specifications, online at: http://microrato.ua.pt/main/Docs/RegrasMicroRato2008_EN.pdf accessed 15 April 2009.

[10] Gilbert, D.: The JFreeChart Class Library Reference Documentation. Simba Management Limited (2002) 158.

Robustness and Precision Analysis in Map-Matching based Mobile Robot Self-Localization

Manuel Gouveia[1], António Paulo Moreira[1,2], Paulo Costa[1,2], Luís Paulo Reis[1,3] and Marcos Ferreira[1]

[1] FEUP – Faculdade de Engenharia da Universidade do Porto, Portugal
[2] INESC-P – Instituto de Engenharia de Sistemas e Computadores do Porto, Portugal
[3] LIACC – Laboratório de Inteligência Artificial e Ciência de Computadores da Universidade do Porto, Portugal
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
{mgouveia, amoreira, paco, lpreis, ee04128}@fe.up.pt

Abstract. In this paper an accuracy analysis is presented for a localization algorithm based on maps correspondence. Initially the algorithm was developed by Martin Lauer [1], for robotic soccer purposes, and it has now been adapted to work on indoor robots localization, using walls and other known fixed objects as references. Infrared range sensors (IR) - for low cost applications - and Laser Scan Range Finder (LS) - for other applications - were used in our approach and compared experimentally. Precision and robustness analysis was made using increasing amount of laser points (which simulates a varying number of sensors) and results were compared to using similar number of IRs. Several tests were conducted with a real robot along a well defined trajectory, following a line previously painted on the floor. Visual markers were placed along the line allowing to keep track of the robot's real position with high precision. The experimental results achieved showed that the implemented algorithm works well with a variety of sensors. The results also show that the scan laser range finder is by far the best type of sensor to measure the range of objects indoors. Using only few points the algorithm achieved good results and demonstrated a very high robustness rate and precision.

Keywords: Mobile Robot Localization, Localization Algorithms, Sensor Fusion, Global Localization

1 Introduction

The main objective of mobile robotics is to develop autonomous and intelligent systems which can operate in real environments and unknown situations. Among the various lines of research carried out in the area of mobile robotics, the functions relating to mobility, such as localization and navigation, have great importance in the development of autonomous and intelligent systems. The localization of mobile robots has involved intense scientific and technological research, and a wide variety of techniques, based on different physical principles and localization algorithms, have already been developed, implemented and studied.

Mobile robot localization is probably one of the most important problems in robotics. Mobile robots are currently used in several applications, as guides in museums [2], [3], for entertainment purposes [4], for indoor cleaning [5], for monitoring patients in a hospital environment [6], [7], as well as in a range of other applications.

In any of these applications, if the mobile robot does not know where it is, how will it be able to complete the task assigned to it? Indeed, it needs to be able to ascertain its location using complex localization algorithms, which require information about the robot's surrounding environment. This information is obtained from sensors, which must have appropriate characteristics for the environment in which they are operating and the information that is needed, while the accuracy of such information is a key issue in ascertaining location.

Localization in a structured environment is based generally on external elements called landmarks. We can use the environment's natural landmarks, or deliberately placed artificial landmarks. The challenge of localization can be easily solved if artificial landmarks or beacons are used [8], [9] together with expensive sensors [10], such as a scan laser range finder [11], which allow large amounts of data to be obtained and which make high precision measurements. Other researchers use natural landmarks, such as walls to solve this problem [12], [13], others have explored other techniques based on vision systems that enable certain visual elements to be identified from the image obtained [14], [15].

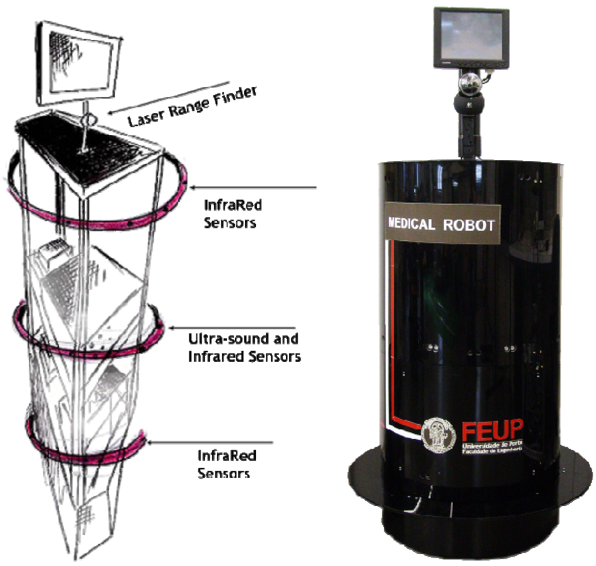


Fig. 1. a) Robot structure and sensor distribution (sketch); b) Medical Robot (foto)

One of the most challenging problems is the difficulty of estimating, with reliability and accuracy, the overall position of the robot in a specific environment.

For example, when the robot is placed in a known environment, it must be able to locate itself without knowing its original position, and also keep track of its local position. Knowing its overall position, the robot can navigate with reliability and, tracking its position locally, the robot can navigate efficiently and perform the tasks assigned to it. To make this possible it is necessary to estimate the status of the robot in real time, based on its initial status and on all the information obtained by its sensors. Therefore, localization is a major obstacle to giving robots real capacity for autonomy.

The aim of this paper is to ascertain whether it is possible to use a method of map matching for robot localization, in real time, using natural landmarks and using low cost sensors. The localization algorithm used is based on [1], it was adapted to be used with other types of sensors than those for which it was originally designed.

The analysis of these algorithms accuracy will be carried out using two kinds of sensors with different range characteristics: Laser range finder and Infra Red. These sensors are not used simultaneously, since the objective is to check their performance independently, especially the Infra Red. The aim of this work is to verify the error and precision in the localization process using these type of sensors.

The experiments were performed in a corridor using the two sensors with data fusion with odometry. Medical Robot (Fig.1) was used [6] as the robot. This robot was built based on the 5dpo soccer robots [16].

This paper is organized as follows: Section 2 presents the main algorithm and the adjustments made to it; Section 3 presents the experimental results achieved; Section 4 presents the work related to localization algorithms with different types of sensors and environments; and Section 5 presents the conclusions of this paper and points out some future work directions.

2 Self-Localization Algorithm

The algorithm implemented was first developed by M. Lauer et al. [1] for the Tribots robot soccer middle-size team [22] to obtain the localization of the robot in a well structured environment, a robotic soccer field, using the field lines. This approach uses an omni-directional camera as the robot's main sensor. The image processing result is a list of positions relative to the robot position and the robot's heading of detected line points.

We have adapted this approach for indoor environments and thus propose some adjustments in the global localization. Instead of the known fields lines we use the walls and other objects in the environment as reference marks represented in a map.

First, the algorithm was implemented and tested in the same conditions verified in [1], on a robotic soccer field, where we verified its reliability and robustness. This also allowed us to adjust the parameters of the algorithm to obtain better results.

For global localization we propose solving the problem in a different way, by estimating the localization for **n** random positions of the robot on the map, for every iteration of the program. For each position the algorithm is applied using the current sensorial information, and **n** localization estimates are calculated, as alternatives to the current localization. Making the comparison of the cost function of each estimate, it is possible to determine which one is better. Because in each iteration, the current robot localization is estimated from its last known position, to obtain global localization and to ensure that it does not jump constantly, the best of all random estimates is saved and compared continuously with all newly obtained random estimates. If that estimate is better than the current estimate, within a given range (5% better, for example), for N consecutive iterations, the robot localization jumps to the new estimated localization. By using repeated random alternative estimates we successively scan the whole map. After several trials and adjustments we concluded that the number of random points, **n**, should be 10, to ensure global localization and obtain appropriate processing time; the new random estimates have to be 10% better for 10 consecutive iterations, to allow the localization to jump to that new estimate. Note that the algorithm implemented in [1], only checks the random positions when the robot start running, and when the localization is assumed to be wrong.

The fusion of data between the estimated localization given by the algorithm and the robot odometry is accomplished by analyzing their uncertainty. Despite the good results we can obtain using only the algorithm, odometry information will always be important, in order to obtain the perception of movement and to guarantee correct localization, especially in the cases of ambiguity which may occur and/or noisy measurements.

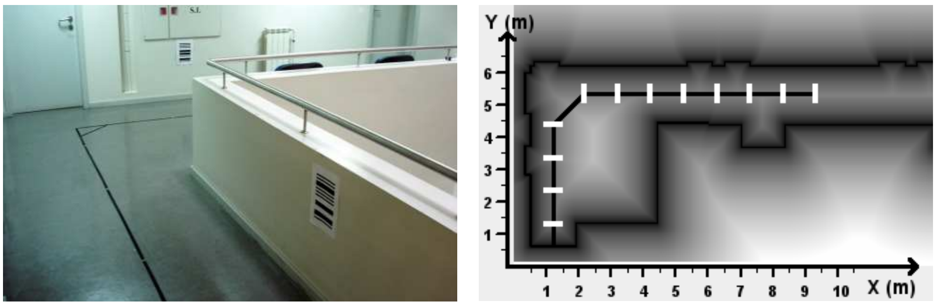


Fig. 2. a) Corridor perspective where the tests were made, with the line on the ground. b) Graphical representation of matrix map used by the algorithm for the corridor and the path and markers used to get real positioning

The map used is also another important issue, because it has the peculiarity of being a matrix, which, in each cell, contains the closest distance to the object on the map. That is, being able to represent each object of the environment with lines and arcs, and considering a cell size for the map, it is possible to gauge the distance of each cell to the nearest object (line, arc) and save its value in a matrix of distances. In Fig. 2b) we can see a graphical representation of the matrix map, in grayscale, where black represents the smallest distance and white is the maximum distance found in the matrix map.

The black line (in the middle of the map) represent the path that the robot follows in every run test we made, and the white marks (rectangles), perpendicular to the black line, are a representation of the markers placed on the ground to give the robot its real localization.

Moreover, we also use other types of sensors able to obtain information used by the algorithm: range of objects present in the environment and represented in the matrix map of distances. Therefore, the main differences of the algorithm presented by [1] are the environment and the type of sensors used to gather information about that environment.

3 Experimental Results

We used two different type sensors: the scan laser range finder [21] and the infrared sensor.

To perform the experiments and get all the information we need, like the robot real position in the environment, we put a line on the ground for the robot to follow it, see Fig 2. The path run 5m in y turns 90º and then runs 7m in x, performing a total of 12m traveled in each run. That line is seen by a camera and with image processing we get its angle and distance relative to the center of robot. With a "FollowLine" control, the robot follows the path we made. Then at every meter we put a marker that can be identified with the image processing, Fig 2. That known marker position, and its orientation relative to the robot body, gives us the exact position and orientation in the moment the robot passes through it. That method of obtaining the robot real position is independent of the algorithm used and was made only to have an external reference to obtain the algorithm error.

Several experiment runs, eleven used for data gathering, were performed and in each one, the data from the odometry, every 11 infrared sensors, 97 laser sensor ranges, were logged at every 40ms, all at the same time and robot real positions was logged when a marker was identified. That will permit a better comparison between the sensors. In every run we used the first mark to define the exact position of the robot.

The program was made in a way that we can run all that logs saved in offline mode (without the robot) as if we were running the robot, remember that we saved all the data needed to do it. That gives us the possibility of repeatability, so that we can run every test how many times we want, and process the data we want to get, like the error of the algorithm.

The matrix used by the localization algorithm, has a resolution of 4 centimeters per cell, although this does not mean that the algorithm give discrete results spaced by 4cm, it is graphically represented in Fig. 2b).

This matrix is pre-calculated and can be stored in a file on disk. This is loaded to memory when the program starts running, along with other required matrices, such as the gradient matrix also used in the algorithm. This allows the algorithm to be fast and efficient, because there is no need to recalculate the matrix of distances for each iteration, the program merely reads the cells of interest in the matrix map from memory, which is an extremely fast process.

The study of experimental data obtained and stored in logs allowed us to compare the error and precision of the algorithm in a real environment for the two sensors used.

3.1 Results using only Odometry

First we run all the data using only odometry, and as we expected the error was high. In global coordinates we get an average error in X of 0.103m, in Y of 0.476m and in orientation of 7,281 degrees. The error grows everytime the robot moves, Fig. 3 and Fig. 5a).

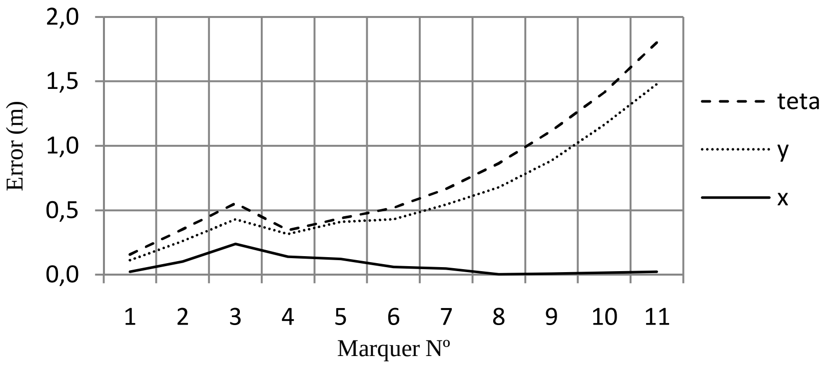


Fig. 3. Evolution of error for a run experiment, odometry only

3.2 Laser Range Finder

The Laser Range Finder, can range in a semicircle of 240º, up to 4m with high accuracy. With this sensor, we can select the number of equally spaced points whose range we want to measure. Accordingly, we measured a maximum of 97, out of 681, equally spaced points in the aperture angle of the sensor. Note that if in one orientation of measurement there is no obstacle it results in a null range. For testing proposes we run the algorithm with 97 points, 49 points, 25 points, 13 points and 7 points in order to examine how the number of points measured impacts the precision and error of the implemented algorithm. We also wanted to compare them directly with the 11 infrared sensors used, although the directions were not the same, as beyond the number of sensors we also want to study the influence of their position, orientation and the spacing between them.

The test runs with 97 points showed that the algorithm can localize the robot even when the robot's odometry is not available for data fusion, even with 49 and 25 points we get the same results, but we want to test it with odometry fusion. Table 1 shows the final results, the average error for the position and orientation given by the algorithm.

Table 1. Average (for all 11 tests, and 11 markers) value of the error for x, y and theta for the laser case, comparing with different number of points

Number of Points	97	49	25	13	7
Error_x (m)	0,0393	0,0404	0,0419	0,0415	0,0492
Error_y (m)	0,0572	0,0539	0,0454	0,0428	0,0376
Error_theta (degrees)	1,3452	1,2251	1,6023	1,2720	1,4882

Although it was predicted a sharp increase in precision with the number of points used, strangely this does not always happen, which may denote a path, environment (map) and physical localization of sensors, dependency. Note that although in X coordinates the average error always increases with the decrease in the number of points, Fig. 4, this is not the case with the angle and surprisingly the average error in Y coordinates decreases with the number of points. Is expected that the robustness of the algorithm decreases as we use fewer points. I.e. the probability of a total loss of global localization is higher.

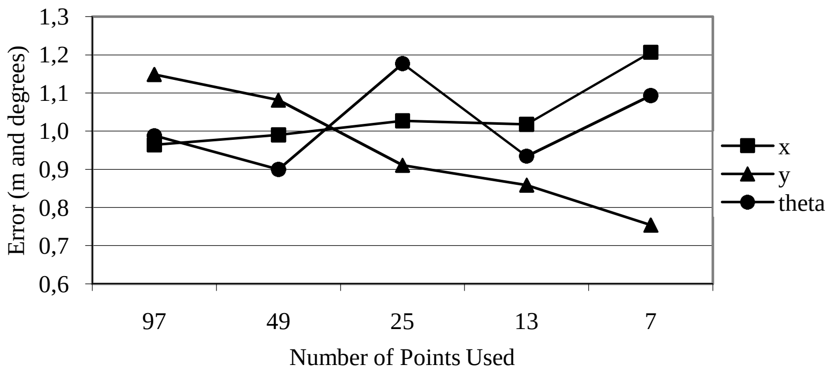


Fig. 4. Error for different number of laser sensor points used

3.3 Infrared range sensors

The Infrared sensor, can range up to 1.5m, with less accuracy and much higher levels of error than the laser range finder. We have 11 sensors mounted around the robot, measuring distances at every 30º. Because of the physical position of these sensors in the robot, Fig 1, the map is different. Fig. 5a).

These 11 sensors are sufficient to properly locate the robot in the sense that we don't have catastrophic failures or errors. Obviously the precision is less than the one obtained with the scan laser. To compare them we interpolate the results from the test runs with 7 and 13 points.

The error obtained from the tests result that in global coordinates we have a average error in X of 0.059m, in Y of 0.089 and in orientation of 4.24º. In one of the markers the robot was in every run a higher error when compared with the other positions, not considering that marker the maximum error in Y drops to half. And so we have a maximum error in X 0.126m, in Y of 0.111m and in theta of 5.58º.

3.4 Comparison

In Fig. 5 we can see an example of a test run (a representation). In each detected mark we can see the robot real position and orientation (at yello) and the localization given by the algorithm evaluated (at red), the yellow dots represent the sensor points position.

As can be seen with odometry only, Fig. 5a), after some meters of movement the robot localization starts to be wrong, when we apply the algorithm with the infrared sensors, Fig. 5b), the localization is much better, maintaining the robot near its real localization. If we look to the Fig. 5c) and 5d) we can see that the error is even less, in the case of the laser range finder. This happens because the sensor accuracy is better than the infrared, and the range of 4m when compared to the 1.5m range of the infrared also influences the localization.

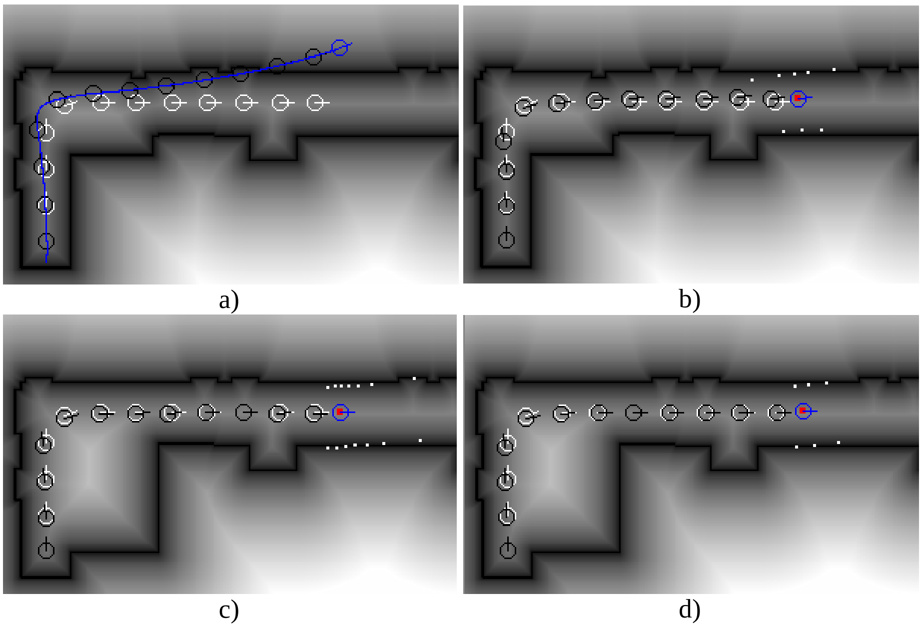


Fig. 5. Representation of path followed by the robot and error of the algorithm: a) with odometry only; b) applying the algorithm with the 11 Infrared sensors; c) applying the algorithm with 13 points of the laser; d) applying the algorithm with 7points of the laser.

Looking at Fig. 5c) and 5d) the difference between them is not noticeable, but comparing with the infrared is. If we look to the values, Table 2, there is a difference of a few millimeters; the error in orientation is more noticeable.

Table 2. Average value of the obtained error of all test runs

Error	X(m)	Y(m)	Theta(degrees)
Odometry	0,1026	0,4762	7,2812
IR	0,0594	0,0887	4,2491
Laser 13	0,0415	0,0428	1,2720
Laser 7	0,0492	0,0376	1,4882

4 Related Work

Several methods have been proposed, ranging from methods based on maps [17], constructed by the robot or pre-programmed, to methods based on recognition of beacons [8] or labels with pre-set positions. The localization problem can easily be solved by placing references in the environment which are distinguishable from all other previously existing objects. However, this solution has its drawbacks, given that the introduction of some changes to the environment may have a negative impact on the environment (for humans). We therefore propose that the environment's own characteristics are used to localize the robot, such as walls and other objects that already exist in the environment.

Typically, these methods have a number of negative characteristics such as being computationally demanding, requiring considerable exploratory time and being inflexible. The localization algorithms [18] that have achieved the best results in robot localization, when compared to other existing algorithms, are those based on the Monte Carlo algorithm [12].

The proposed algorithm in [1] uses a minimization algorithm, Resilient Propagation [19], that is also used in neural networks. This allows rapid convergence of the localization estimate, achieving a more probable location of the robot, with a minimum of error. This method is based on map matching and the major innovation of the method is the fact that the map is pre-calculated, meaning that the values of the distances which are used by the map matching are pre-calculated. These distances are stored in memory in a matrix, which is the map of the algorithm. The gradients of these distances, which are used to minimize the localization error, are also pre-calculated. All of this results in fast processing time and also quickly gives the second derivative of the cost function, allowing us to ascertain the level of uncertainty and the estimated variance of the localization given by the algorithm and therefore merge it correctly with the odometry. As we can read in [1] there are other approaches that are closely related to this one. For example Cox [20] that also uses range finders to detect the walls. But this new approach tackles and improves its bad aspects by giving global localization, a better optimizer, dealing better with the outliers, the robot can move faster, the process time is smaller and any slippage in the odometry is compensated with the algorithm itself.

5 Conclusions

With this research, we verified that the used algorithm works well with a variety of sensors, which are capable of measuring distances to the defined landmarks, whether natural or artificial.

The results achieved are quite satisfactory and enable us to conclude that, for the predicted application, a service robot in a non-industrial environment without the need for high precision, with a minimum of eleven low cost sensors, we have enough robustness and precision, even considering that some of these sensors are giving wrong measurements.

As demonstrated, the algorithm can successfully be used in other applications and with other types of sensors beyond robotic soccer and the video camera sensor, for which it was originally designed.

The results show that the scan laser range finder is by far the best type of sensor to measure the range of objects indoors. Using only few points the algorithm achieved good results and demonstrated a higher robustness rate and precision than the infrared sensor, but at a higher price, which makes it unavailable for most applications.

With this study, we also concluded that the number of points whose range is measured is important; higher numbers of points result in better robustness for global localization, only needing a few points for local localization with odometry. The accuracy and range of measurements is also important because more accurate measured points allows better map matching.

Future work includes adjusting the algorithm to use the maximum number of points possible for global localization but fewer points for adjust of local localization with the fusion with odometry. to verify the accuracy and precision of the algorithm and to evaluate the localization error obtained with it. Also, we intend to tune the weight of the fusion between location and odometry for the IR as the optimal value may be different for the Laser. Other line of future work includes improving the algorithm in the way to better reject the “outliers”, meaning the bad measurements. Finally, we intend to improve the robustness when the robot faces objects that are not covered in the map, like for example moving objects such as people or animals.

Acknowledgments

This research was supported by the Portuguese Science and Technology Foundation (FCT) under the project "ACORD – Adaptative Coordination of Robotic Teams" (PTDC/EIA/70695/2006), funded through the POCI/FEDER program.

References

1. M. Lauer, S. Lange and M. Riedmiller., "Calculating the Perfect Match: an Efficient and Accurate Approach for Robot Self-Localization." *In RoboCup 2005: Robot Soccer World Cup IX , LNCS.* Springer, 2005, pp 142-153

2. S.Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hnel, C. Rosenber, N. Roy, J. Schulte and D. Schulz, "Minerva: a Second-Generation Museum Tour-Guide Robot" *In Proceedings of the 1999 IEEE International Conference in Robotics and Automation*, Detroit, MI, 1999 vol.3, pp.1999-2005
3. D. Fox, S. Thrun, W. Burgard, and F. Dellaert, "Particle Filters for Mobile Robot Localization." *In Sequential Monte Carlo Methods in Practice.* 2000. pp. 470-498.
4. B. Traub A. John D. Schraft and R. D. Graf. "A Mobile Robot Platform for Assistance and Entertainment". *In Industrial Robot Journal*, 28:83-94, 2001.
5. A. Sousa, P. Moreira and P. Costa, "Multi Hypotheses Navigation for Indoor Cleaning Robots", *In Proceedings of the 3rd International Workshop on Intelligent Robotics* IRobot 2008 October 14th, 2008, Lisbon University Institute ISCTE, Portugal, pp 71-82.
6. T. Costa., "Medical Robot." Tese de mestrado em Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Portugal, 2008.
7. Yulun Wang, Steven E. Butner, and Ara Darzi. "The Developing Market for Medical Robotics" Vol. 94, No. 9, September 2006, pp.1763-1769.
8. J. S. Esteves, A. Carvalho and C. Couto., "Generalized Geometric Triangulation Algorithm for Mobile Robot Absolute Self-Localization." *In IEEE International Symposium on Industrial Electronics (ISIE).* Rio de Janeiro: s.n., 2003. Vol.1, pp. 346-351
9. M. Betke e L., Gurvits., "Mobile Robot Localization using Landmarks." *IEEE Transactions on Robotics and Automation.* 1997. Vol. 13. vol.13, no.2, pp.251-263
10. J. G. Mota and M. I. Ribeiro, "Mobile Robot Localization on Reconstructed 3D Models," *Journal of Robotics and Autonomous Systems*, 2000. Vol. 31, no. 1, pp. 17-30,
11. Hee Jin Sohn, Byung Kook Kim, "A Robust Localization Algorithm for Mobile Robots with Laser Range Finders", *In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA* 2005. 18-22 April 2005, pp. 3545 – 3550.
12. F. Dellaert, W. Burgard and S. Thrun. "Monte Carlo Localisation for Mobile Robots." *In IEEE Int. Conf. on Robotics and Automation (ICRA99).* 1999. Vol.2, pp.1322-1328
13. J. J. Leonard e H. F. Durrant-Whyte, "Directed Sonar Sensing for Mobile Robot Navigation." *Kluwer Academic.* Boston : s.n., 1992.
14. A. Sousa, P. Costa, A. Moreira, A. Carvalho. "Self localization of an Autonomous Robot: Using an EKF to Merge Odometry and Vision Based Landmarks." *Proceedings of the IEEE Int. Conf. on Emerging Technologies and Factory Automation.* 2005. pp. 227-234.
15. D. Cabecinhas, J. Nascimento, J. Ferreira, P. Rosa, P. Lima. "Self-Localization based on Kalman Filter and Monte Carlo Fusion of Odometry and Visual Information." *Robótica.* 2006. No. 65
16. 5dpo Robotic Team Home Page, "5DPO Robotics". [Online] available at: http://www.fe.up.pt/~robosoc, consulted on June 2009.
17. N. Karlsson, E. Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian and M. Munich. "The vSLAM Algorithm for Robust Localization and Mapping." *In International Conference on Robotics and Automation.* ICRA 2005. pp. 24-29
18. S. Thrun, W. Burgard and D. Fox., "Probabilistic Robotics" *The MIT Press.* 2005.
19. M. Riedmiller and H. Braun, "A Direct Adaptative Method for Faster Backpropagation Learning: the RPROP Algorithm." In Proceedings of the IEEE International Conference on Neural Networks. 1993. pp. 586-591.
20. I. J. Cox, "Blanche – an Experiment in Guidance and Navigation of an Autonomous Robot Vehicle", IEEE Transactions on Robotics and Automation, 7(2):193–204, 1991.
21. Hokuyo, Laser range finder Hokuyo URG-04LX. [Online] Avaible at: http://www.hokuyo-aut.jp/02sensor/07scanner/urg.html, consulted on April 2009.
22. Brainstormers, Tribots, "Brainstormers, Tribots Homepage" [Online] Available at: http:\\www.ni.uos.de\index.php?id=2, consulted on April 2009.

MODEN: Obstacle-driven Elastic Network for Line-of-Sight Communication*

Francisco S. Melo and Manuela Veloso

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{fmelo,veloso}@cs.cmu.edu

Abstract. In this paper, we address the problem of connecting two distant communication nodes by deploying a number of mobile robots that act as gateways between the towers. To address this problem, we propose ODEN, a path-planning algorithm that relies on an *elastic network* to produce an obstacle-free path between the two towers. This algorithm builds over a previous elastic-network-based path-planning algorithm and overcomes some of the major limitations of this algorithm. We also propose an extension of ODEN to address the more complex situation in which multiple pairs of towers must be connected in a common environment. We illustrate the results obtained with both ODEN and its multi-path counterpart, MODEN, in several test-scenarios.

1 Introduction

In this paper, we consider the problem of connecting two distant communication nodes (henceforth referred as "towers") placed in such a way that no communication is possible between them. To overcome this difficulty, a number of mobile robots can be deployed in the environment. These robots possess the ability to communicate with the towers (when in range) and with each other. In the most general setting, we assume that the robots have only local knowledge about the configuration of the environment (*e.g.*, about the existence and position of obstacles) obtained by means of any local on-board sensors. The purpose of the robots is to navigate the environment until a configuration is reached in which the two towers are connected, as illustrated in Fig. 1(a).

In this paper, we model this problem as a *path planning* problem between the positions of the two towers and use an "elastic network" to model the movement of the gateway robots deployed in the environment. Our algorithm is an extension of a previous method proposed in [1], in which the path is represented as an elastic network initialized as a straight-line between the two target positions. The mobile

* This work was partially supported by the Information and Communications Technologies Institute, the Portuguese Fundação para a Ciência e a Tecnologia, under the Carnegie Mellon-Portugal Program. A preliminary version of this work was previously presented at 7th Portuguese Robotics Festival and Scientific Meeting, 2007.

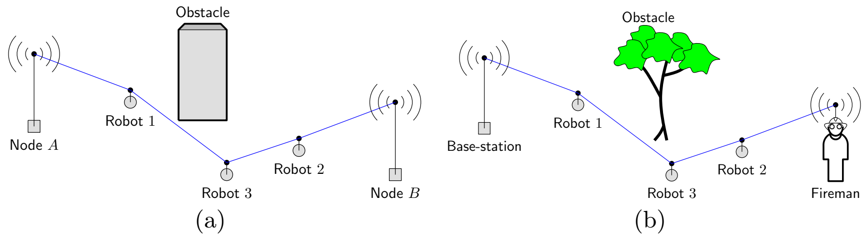


Fig. 1. Illustration of the typical scenario considered in the paper and a motivating example.

nodes in the network (the mobile robots in our scenario) are "attracted" toward free-space and "repelled" away from the obstacles, yielding an obstacle-free path.

The contributions of this paper are two-fold. On one hand, we improve on the original algorithm from [1] by proposing a new update rule that yields smoother and more efficient deployment of the robots, in a sense soon to be made clear. This leads to the ODEN algorithm, our first contribution. On the other hand, we propose an extension of ODEN to address the more complex situation in which multiple pairs of towers must be connected in a common environment. In this situation, and to minimize interference, it is desirable that the deployment of the robots is conducted in such a way as to minimize intersection between the different communication paths. This leads to MODEN, the "multi-path" version of ODEN and the second contribution of this paper.

To motivate this work, consider, for example, a forest fire situation in which firemen must maintain contact with a base-station. In order to do so, they can carry small, portable mobile units that they drop as they move toward their assigned position. These units will ensure connectivity between each fireman and the base-station by locally adjusting their position so as to improve connectivity and minimize interference (see Fig. 1(b). Aimed at such applications, the approach in this paper explicitly considers the existence of obstacles in the environment, driving this work away from other works addressing open space deployment and connectivity [2]. Furthermore, we are particularly interested in modeling scenarios in which the robots move in an unstructured (outdoor) environment. In these scenarios, signal obstruction can arise from obstacles such as groups of trees that can actually be *traversed* by the robot, but cause severe decrease in the ability of the robot to communicate with its neighbors. Also, in the setting considered herein, we are not concerned with radio-based *navigation* or *localization*, a topic of recent current research [3–7]. As will be apparent from our algorithm, the robots need only *minimal navigation capabilities*. In particular, the algorithm will not require them to use any localization or path planning algorithm, but simply execute very simple movement primitives (such as moving straight in a given direction). Finally, the problem considered herein is, in a sense, closely related to the problem addressed in [8]. The latter work proposes an algorithmic procedure to determines *when* the robots can move in order not to loose the overall connectivity of the network. In this paper we propose an algorithmic procedure to determine *how* the robots can move in order to improve the overall connectivity of the network.

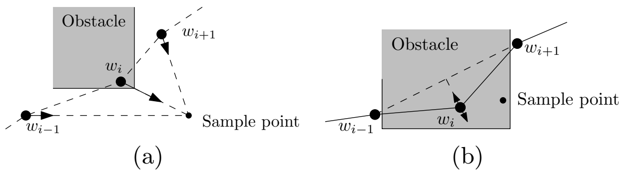


Fig. 2. Several updates of the closest PU, depending on the sampled point.

2 Path Planning Using an Elastic Network

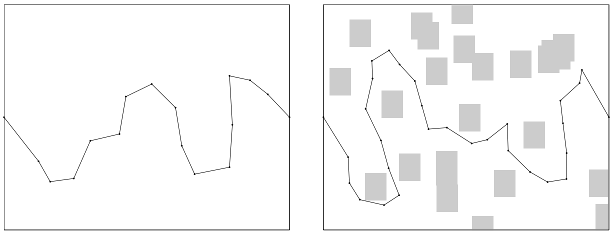
The algorithm described in [1] makes use of an elastic network of "processing units" (PU) linking the two target towers throughout the environment (henceforth abusively referred as the *configuration space* for the network). The network is initialized as a straight line between the two towers and we denote by $\mathcal{U}^t$ the set of all units in the network at iteration t. The algorithm proceeds at each iteration t by sampling a random point within a distance of r around the network, *i.e.*, a point in the "perceptual range" of some robot in the network. It then updates the position of the nearest node in the network (the best matching unit, BMU) according to the following heuristic:

– If the sample point is in free space, it "attracts" the BMU toward it (see Fig. 2(a));
– If the sample point is in occupied space, it "repels" the BMU away from it;
– If both the sample point and the BMU lie in occupied space, the BMU moves orthogonally to the line segment whose extreme points are BMU's neighboring units (Fig. 2(b)).

Once the BMU is updated, its immediate neighbors are updated accordingly. The points used in the updates are randomly sampled around the nodes in the network, in a region with radius r that is successively decreased to a final minimum value of r_F. The algorithm starts by "pulling the network" toward large regions of free space and then locally adjusts the network to the "details" of the environment.

It is worth noting that, in terms of the robots in the network, these "abstract" operations can be implemented in a remarkably simple way. At each iteration, one of the robots in the network uses its sensor information to move toward a randomly chosen position in free-space. This will typically cause the signal strength between the robot and its neighboring robots to change. The neighboring robots then adjust their positions so as to compensate for the change in the signal strength. We also note that, in practice, this operation can be performed by multiple robots simultaneously, as long as none of the robots are contiguous or share a neighbor.

Finally, every λ time instants, a new unit is added in between the two most distant units in the network until a maximum number of units is reached. This condition could easily be replaced by similar condition of adding a new unit (a new robot) whenever the distance between two nodes in configuration space was above a pre-specified threshold. However, the use of a temporal condition allows the initial updates to be more efficient.



(a) No obstacles. (b) 25 obstacles.

Fig. 3. Trajectories obtained by the original algorithm.

The term *elastic network* arises from the fact that each unit exerts some conservative tension on its neighboring units so as to maintain their relative positions. In other words, the updated PU is "pulled" by its neighbors and also "pulls" its neighboring units. More details on the algorithm can be found in the referred work [1]. This "elastic tension" between neighboring units can be seen as a local adjustment that each of the units performs whenever one of its neighbors moves in order to improve connectivity.

The method has several interesting properties. It is extremely simple to implement. It only requires a sampling mechanism around the network and the evaluation of the *attraction function.* This attraction function merely determines whether a point is in free-space or in occupied space and the update of a unit reduces to a few extremely simple operations that translate into very simple motion commands. Also, given its simplicity, it is a surprisingly efficient algorithm, being in fact able to find obstacle-free paths between the desired points with very little computational effort. The examples reported in [1] were able to determine obstacle-free paths in several complex environments in few thousand iterations. Furthermore, it does not require any search or any model of the environment, but just a sampling function able to determine whether a point lies in free-space or not. Finally, the method is *local* in that each unit is updated considering only its immediate neighborhood. This is a very interesting feature of the algorithm for the particular set of problems we are interested in. In our setting, each unit is actually an independent robot that must adjust its individual position communicating only with its neighboring units and sampling the surrounding space, but the method still provides a way of determining a *global* obstacle-free network.

However, it also presents several inconveniences. First of all, the algorithm will often spend a lot of iterations updating units that need not be updated. This happens, for example, in an obstacle-free environment or any environment where there is a lot of free space, where most of the units will already start away from any obstacle. Furthermore, because sample points in free-space keep attracting the network to free-space, this may lead to peculiar trajectories as depicted in Fig. 3. Finally, the algorithm was designed to run for a fixed (usually large) number of iterations and takes generally this number of iterations to terminate before a path is produced. In the next section we propose a modification of the algorithm that, while maintaining the simplicity of the original algorithm and its functional principle, alleviates the reported inconveniences.

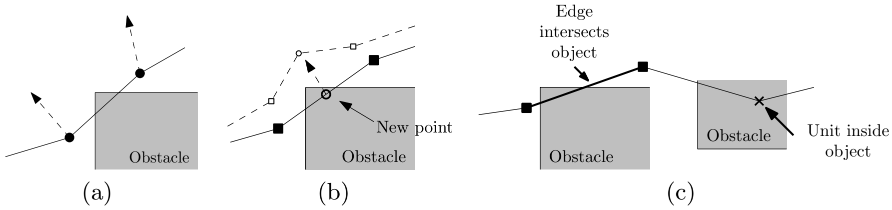


Fig. 4. Situations addressed by the new update mechanism. (a) Situation where further updating is desirable. (b) The extra point solves the problem. (c) Two different classes of "updatable" units.

3 The ODEN Algorithm

In this section we describe two fundamental changes to the algorithm in the previous section: a new update mechanism for the units and an additional condition to introduce new units in the network. These modifications yield the ODEN algorithm,[1] and readily overcome the inconveniences pointed out in the previous section. We also present the results obtained with ODEN that show the introduced modifications to actually improve the overall efficiency and performance over the original algorithm.

3.1 The Update Mechanism

Recall that the two main inconveniences reported in the previous section are the many iterations spent updating units that need not be updated and the fact that the *whole network* is attracted toward free-space, this leading to undesirable trajectories. This undesirable behavior is mainly due to continuous updating of units that are already in free space. To overcome such phenomenon we introduce a simple modification to the algorithm: we partition the set $\mathcal{U}^t$ into two sets, $\mathcal{U}_U^t$ and U_F^t. The set $\mathcal{U}_U^t$ contains the "updatable" units and the set $\mathcal{U}_F^t$ contains the "fixed" units. In a first approach, we consider the set U_F^t as the subset of $\mathcal{U}^t$ lying in free-space. Therefore, we update only the units that lie inside objects.

There is a purpose in continuously updating the different PUs even when they are in free space. Consider the situation depicted in Fig. 4(a). In this situation there is an obstacle between the two depicted units that may hamper the corresponding communication channel. In the original algorithm, the two units depicted would be continuously updated, eventually pulling their common edge out of the obstacle. If we consider the set $\mathcal{U}_F^t$ as containing all units in free space, none of the two units depicted will ever be updated again, and the path thus obtained will not be obstacle-free, even if all units may eventually lie in free space. To overcome this situation, we consider a second class of updatable units: those whose contiguous edges intersect any of the obstacles. In Fig. 4(c) we illustrate the two classes of updatable units.

One final remark to refer that, in order to determine if a given edge intersects any object, we sample several random points along the edge and test whether they lie in free space. The number of points depends on the actual length of the edge. This method generally produces quite reliable results even without

[1] Obstacle-Driven Elastic Network.

sampling too many points. In the case of actual robots, the same information can generally be determined from the link between the two robots.

3.2 Introducing New Units

Recall that in the original algorithm a new unit is added to the network every λ iterations up to a maximum pre-determined number of units. The addition of such extra units, as well as a decrease in the sampling radius around the network, allows the network to converge to smoother trajectories and forces the updates to consider increasingly *local* data around each unit (see [1] for complete details).

Considering the modified version of the algorithm described insofar, it is possible and indeed likely that all units reach free-space before λ iterations have occurred. This implies that the algorithm will "stall" until a new unit is introduced. As such, we include an additional condition in the algorithm, and a new unit is added to the network if any of two conditions is verified:

- Whenever λ iterations have occurred; or
- Whenever $\mathcal{U}_U^t$ is empty.

If the algorithm introduces a new unit in the network before λ iterations have occurred, the sampling radius is, nevertheless, decreased accordingly.

We note that the introduction of new units also alleviates the problem described in Fig. 4(a). In fact, in many situations such as the one in Fig. 4(a), a new point will actually be added between the two points in Fig. 4(a), leading to a solution like the one on Fig. 4(b).

3.3 The ODEN algorithm

We refer to the modified version of the algorithm as ODEN, standing for obstacle-driven elastic network. ODEN is summarized in Algorithm 1, where α is a randomly chosen number such that $-\beta \leq \alpha \leq \beta$ and F is the attraction function.[2]. The parameters $t_{\max}$, λ, β, η_0 and η_1 are common to the algorithm in [1]. The first two parameters represent, respectively, the maximum number of iterations for the algorithm and the number of iterations between two insertions of a new point. The last three parameters define the "update rates" and "elastic coefficient" used in the updates of the various components of the algorithm. The input parameters x_I and x_F represent the positions of the two towers and N_0 and $N_{\max}$ represent the initial and final number of nodes in the network.

3.4 Experimental Results with ODEN

We tested ODEN in several random environments with different degrees of complexity. In all results displayed, the algorithm was run with the same parameters as those reported in [1].[3]

Figure 5 presents the results obtained in the same environments as those in Fig. 3, illustrating how ODEN is able to overcome the inconveniences of

[2] The attraction function is just an indicator function for an obstacle, taking the value of 1 if x is in the free-space and -1 otherwise.

[3] In particular, we use $N_0 = 10$, $N_{\max} = 100$, $\beta = 0.0025$, $\eta_0 = 0.05$, $\eta_1 = 0.01$, $r_I = 2$, $r_F = 0.7$ and $t_{\max} = 40,000$.

Algorithm 1 The ODEN algorithm.

Require: $x_I, x_F, N_0, N_{\max}, r_I, r_F$;
1: Set $w_0 = x_I$ and

$$w_i = x_I + i \cdot \frac{x_F - x_I}{N_0 - 1}, \quad i = 1, \ldots, N_0 - 1.$$

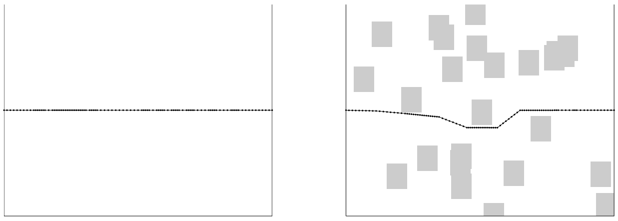
2: Set $t = 1$, $L_{ins} = 0$, $N = N_0$;
3: Compute $\mathcal{U}_U^t$;
4: **if** $\mathcal{U}_U^t = \emptyset$ **then**
5: Set $t = L_{ins} + \lambda$ and goto 17;
6: **end if**
7: Randomly choose $w_i \in \mathcal{U}_U^t$;
8: Set $r = r_I \cdot \left(\frac{r_F}{r_I}\right)^{t/t_{\max}}$;
9: Randomly choose $x \in X$ such that $\|x - w_i\| < r$;
10: Set $w_j = \arg\min_{w \in \mathcal{U}_U^t} \|x - w\|$;
11: **if** $F(x) > 0$ **then**
12: $w_j \leftarrow w_j + \eta_0(x - w_j) + \beta(w_{j-1} + w_{j+1} - 2w_j)$;
13: $w_{j\pm1} \leftarrow w_{j\pm1} + \eta_1(x - w_{j\pm1})$;
14: **else**
15: $w_j \leftarrow w_j + \alpha \frac{(w_{j+1} + w_{j-1})^{\perp}}{\|w_{j+1} + w_{j-1}\|}$;
16: **end if**
17: **if** $t - L_{ins} \geq \lambda$ **then**
18: $L_{ins} = t$;
19: Insert w_{new} such that $w_{new} = \frac{w_k + w_{k+1}}{2}$, where

$$\|w_k - w_{k+1}\| = \max_{w_i \in \mathcal{U}_U^t} \|w_i - w_{i+1}\|.$$

20: $N = N + 1$;
21: Update $\mathcal{U}_U^t$;
22: **end if**
23: **if** $t < t_{\max}$ and $(I \neq \emptyset \vee N < N_{\max})$ **then**
24: Set $t = t + 1$ and goto 7;
25: **else**
26: Exit;
27: **end if**

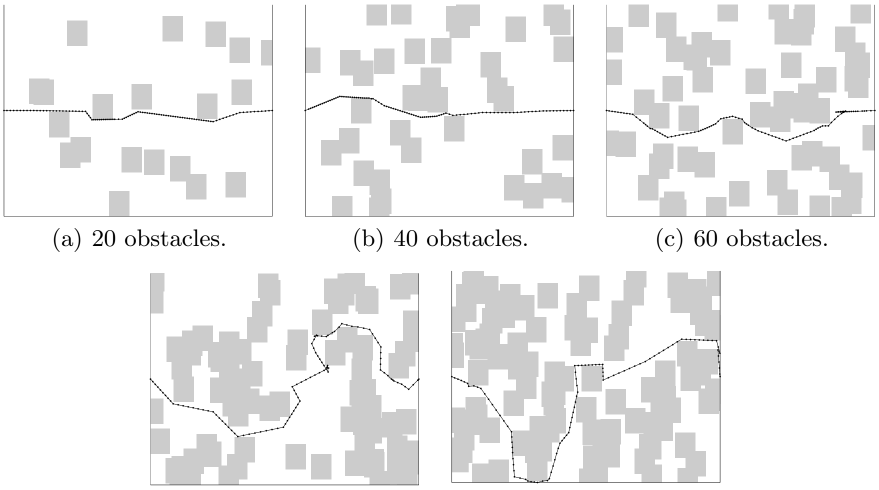
the original algorithm reported in the previous section. Figure 6 presents the results obtained in several random environments ranging from 20 to 100 random obstacles. It is worth noting that even if the algorithm does not take into account the minimization of the length of the path, the obtained paths often exhibit small perturbations from the initial straight path. Note also that the intersection effect in Fig. 4(a) is not observed in the trajectories, even if in many situations the path is close to these.

Finally, Fig. 7 presents the results obtained in environments with non-random obstacles. The distributions of the obstacles in these environments can be seen



(a) No obstacles. (b) 25 obstacles.

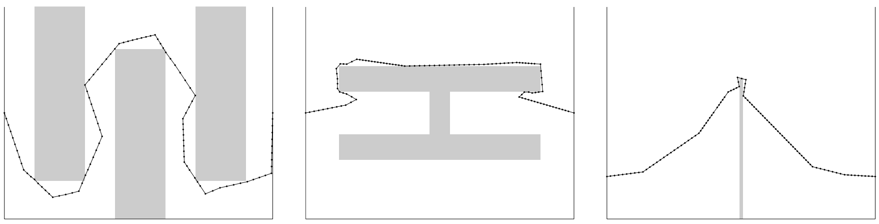
Fig. 5. Trajectories obtained with ODEN in the environments of Fig. 3, corresponding to 100% and approximately 77% of free-space.



(a) 20 obstacles. (b) 40 obstacles. (c) 60 obstacles.

(d) 80 obstacles. (e) 100 obstacles.

Fig. 6. Sample network outlines obtained with ODEN in five random environments with diverse number of obstacles, respectively corresponding to approximately 82%, 64%, 46%, 28% and 10% of free-space.



(a) (b) (c)

Fig. 7. Trajectory obtained with ODEN in environments with non-random obstacles.

as possible worst-case situations for the algorithm. Notice that the algorithm is, in fact, able to deliver the expected collision-free paths.

We note that, in the first set of environments, the algorithm is generally able to deliver a solution *and stop* within few iterations. This is in clear contrast with the original method, which would run for a predefined number of iterations. Also, ODEN generally performs better on more sparse environments than in environments with little free space. This is easily explained if we consider that the algorithm proceeds by sampling the space around the network and pulling the network toward free-space and away from the obstacles. If few sample points lie in free space, more iterations will be required for the algorithm to find a proper path. Finally, "thin" obstacles may be hard to sample and, therefore, hard to avoid. We produce one further experiment to test the performance of the algorithm when a single thin, long obstacle is found in the environment. The result is presented in Fig. 7(c). Even though the environment has a single obstacle and is, in general, a sparse environment, the algorithm took a considerable number of iterations (≈ 720) before a solution was found.

4 MODEN: ODEN in Multi-path Domains

We now address the more complex problem of connecting n pairs of towers in a common environment. Given the positions in configuration space for all towers in the set, we are interested in determining *individual paths* joining the two towers in each of the n pairs, so that the paths are non-intersecting. The requirement for non-intersecting paths is mainly to avoid undesirable interference between the corresponding communication channels. We henceforth refer to the nodes between the pair of towers i as the *network* EN_i and refer to the positions of the corresponding towers as the "initial" and "final" points of EN_i (with no particular convention as to which is the initial and which is the final).

Given the initial and final points in configuration space for each of the n networks, we can apply the ODEN algorithm to determine the path describing each network individually, yielding n distinct paths, $EN_1, \ldots, EN_n$. However, this approach does not take into consideration the existence of the other networks in the same environment and will probably lead to intersecting paths. A simple idea to overcome such difficulty is to consider each PU in the path of other networks as the center of some obstacle when running ODEN for each network EN_i.[4] The size of the obstacle can be adjusted as a parameter. This strategy turns the other networks's path into "obstacles" and ODEN should thus produce individual trajectories that do not intersect, if possible.

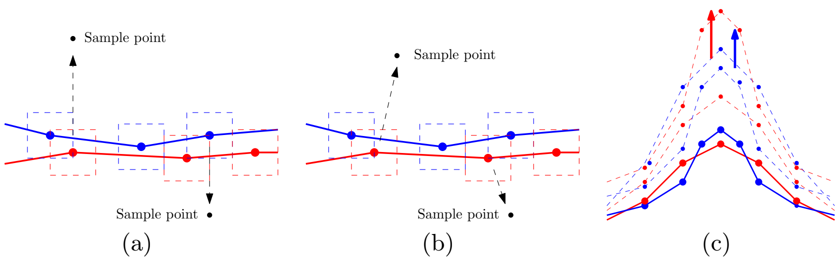


Fig. 8. Situations that may lead to intersection. (a) Different networks sample on opposite sides. (b) Same network samples on opposite sides of neighboring network. (c) Two networks keep "chasing" each other.

[4] The consideration of the other networks as an obstacle can also be implemented using the connectivity information between the robots in the networks.

Algorithm 2 The MODEN algorithm.

Require: $\{x_I^1, \ldots, x_I^n\}, \{x_F^1, \ldots, x_F^n\}, N_0, N_{\max}, r_I, r_F$;

1: Initialize X_{obs} with the obstacles from the environment;

$$w_0 = x_I;$$
$$w_i = x_I + i \cdot \frac{x_F - x_I}{N_0 - 1}, \quad i = 1, \ldots, N_0 - 1.$$

2: **for all** i=1,…,n **do**
3: $P_i = ODEN(x_I^i, x_F^i, N_0, N_{\max}, r_I, r_F)$;
4: Append P_i to X_{obs} (the extreme units in the network are not considered);
5: **end for**

However, if ODEN updates all trajectories simultaneously, this will lead to intersecting paths. Consider the situation depicted in Fig. 8(a). If the samples used to update each path continue to be sampled in opposite sides of the other path, the final paths will unavoidably intersect. Furthermore, situations may occur in which the paths keep "chasing" each other, as they keep sampling free points in one direction while trying to avoid the other path (Fig. 8(c)).

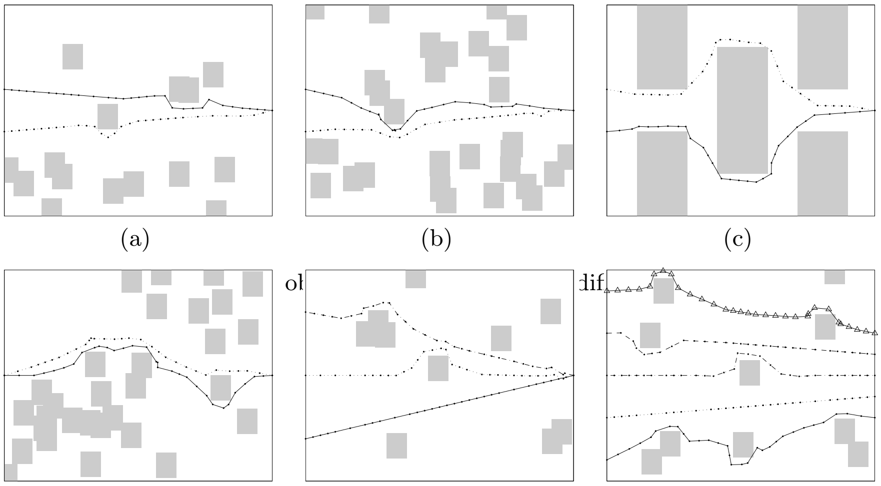
To minimize such phenomenon, we introduce an ordering among the networks and run the ODEN algorithm sequentially according to that order. According to this ordering, it is possible to refer to each of the networks in the set as $EN_1, \ldots, EN_n$, where EN_i stands for the ith network in the given ordering. We now successively apply the ODEN algorithm to each of the networks $EN_1, \ldots, EN_n$ individually: we determine the path for network EN_1 using the ODEN algorithm and considering only the original obstacles in the environment. Then, for each other robot EN_i, $i = 2, \ldots, n$, we determine the corresponding path using the same ODEN algorithm, but considering as obstacles the paths of the networks $EN_1, \ldots, EN_{i-1}$ besides the natural obstacles in the environment. Finally, to alleviate the intersections arising from cross-sampling (Fig. 8(b)), we include a small adaptive bias in the random sampling process.

4.1 The MODEN algorithm

MODEN is summarized in Algorithm 2, where $\{x_I^1, \ldots, x_I^n\}$ and $\{x_F^1, \ldots, x_F^n\}$ denote the sets of initial and final points for the n networks. P_i is the path generated for EN_i. As for the bias in sampling, recall that ODEN uses points sampled in a region of radius r around the network to update the units in the network. A sample point x is given by

$$x = w_i + r_x \angle \theta_x,$$

where w_i is a unit in the network, r_x is a random number between 0 and r, and θ_x is a random angle between 0 and 2π. Let $I_1, \ldots, I_k$ be a uniform partition of the interval $I = [0, 2\pi]$. A uniform sampling procedure chooses an angle θ_x in I_i with probability $p_i = 1/k$. Suppose that the chosen angle at some iteration was α_x and the corresponding point x is in free space. Then, the probabilities p_i are



(a) (b) (c)

(a) (b) (c)

Fig. 10. Sample trajectories obtained for problems with coincident endpoints and with multiple networks (3 and 5).

biased toward the direction α_x according to

$$p_i = \begin{cases} p_i + \epsilon(1 - p_i) & \text{if } \alpha_x \in I_i; \\ (1 - \epsilon)p_i & \text{otherwise} \end{cases}$$

and then normalized to yield $\sum_i p_i = 1$.

4.2 Results using the multi-path algorithm

We now present the results of MODEN in different scenarios. In Fig. 9 we present the trajectories obtained in environments with 15 and 30 random obstacles and 5 non-random environments, where the networks joint different initial positions to a common final position. We then applied the algorithm to the extreme situation where two paths must joint the same initial and final positions. Notice that this implies that the paths for both networks are coincident in the initial iteration. Figures 10(a) depicts the results obtained in environments with 30 obstacles. In Fig. 10(b) and 10(c) we depict the results obtained when 3 and 5 networks are considered.

As in Section 3, the algorithm was run with the same parameters as those reported in [1]. The dimension of the "virtual obstacles" around each PU is $\frac{1}{3}$ of the size of the actual random obstacles.

5 Discussion

From our results, we claim ODEN to be an simple but efficient algorithm, being able to provide obstacle-free paths even in environments with a large percentage of occupied area. In a practical application, ODEN simply requires a function that samples a point in free-space around each robot in the network. This information can easily be retrieved from sensorial data and makes this method simple to implement. On the other hand, the sampling mechanism supporting ODEN implies that the algorithm is less effective in environments with little free-space or with very thin obstacles. In the former case, it will be hard to sample points in free-space to pull the network away from the obstacles. In the latter case, it will be hard to sample the obstacles so as to drive the network away from them.

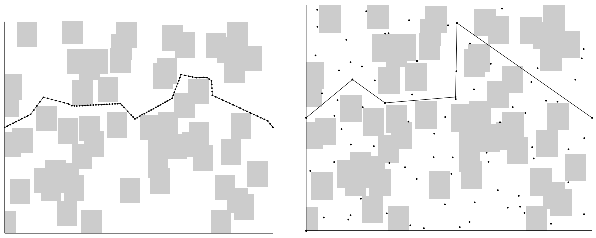
MODEN extends the simple principle behind ODEN to multi-path scenarios, while remaining a local method driven by obstacles. MODEN attests the applicability of ODEN in more complex problems and suitably illustrates the effectiveness of ODEN's underlying working principle. Nevertheless, and in spite of the encouraging results, we should remark that if the initial straight-line path of the several networks intersects, it is generally not possible to ensure a non-intersecting solution. And it may also happen that the path for the first network renders the task of finding an obstacle and intersection-free path infeasible.

As future work, it would be interesting to explore the use of richer attraction functions F. If the attraction function F is more than an "obstacle-indicator", the algorithm may use the extra information to drive the updates in a more informed fashion and thus improve its performance. In particular, the indicator function can provide information about signal strength, not only pulling each node toward free-space but inclusively drive each node/robot toward directions in which the signal strength is improved the most. This would bring our approach closer to potential-field/vector-field methods [9].

It is important to point out that, due to the local nature of the methods, neither ODEN or MODEN are optimization methods. This means that the network is adjusted *locally* to reach an obstacle free path. Since it departs from a straight-line condition, it is expectable that it may be able to reach a network configuration that is "close" to the straight-line path. However, and as other probabilistic path-planning methods (see, for example, Fig. 11), the configuration attained will generally not be optimal in any particular sense.

Another interesting discussion arises from the consideration of ODEN and MODEN as *path planning* algorithms, in which each network corresponds to the actual path of *one* robot. In this context, it is important to note that most path-planners are *global planners* in that they use information from the whole network which may not be practical (or even feasible) in the scenarios considered in this paper. Also, they are typically off-line methods, thus less suited for on-line implementation as intended in the applications envisioned here. On the other hand, and from a multi-robot path planning perspective, the requirement of non-intersecting paths in MODEN is not such a usual approach, as more elaborate ways exist to coordinate multiple robots wandering in a common environment (see, for example, the approaches in [10–12]). Nevertheless, non-intersecting paths alleviate the need for any knowledge of the robot dynamics/communication capabilities. The paths generated by the algorithm would be immediately usable by the robots, without considering any coordination or synchronization mechanism to prevent on-path collisions.

We note that, as with ODEN, environments with little free space also cause difficulties to probabilistic path planning methods such as PRMs [13]. Also, the



(a) ODEN. (b) PRM.

Fig. 11. Trajectories obtained with ODEN and PRM.

fact that ODEN uses all nodes in defining its path implies that, in general it will be able to locally adjust to the obstacles, producing potentially shorter paths. Figure 11 compares the paths obtained with ODEN and PRM in an environment with 60 random obstacles. A more extensive comparison of the performance ODEN againsta that of other state-of-the-art planning methods is still necessary to further understand the applicability of ODEN in more general scenarios than those considered here.

References

1. Moreno, J., Castro, M.: Heuristic algorithm for robot path planning based on a growing elastic net. In: 12th Port. Conf. Artificial Intelligence. (2005) 447–454
2. Ludwig, L., Gini, M.: Robotic swarm dispersion using wireless intensity signals. In: Int. Symp. Distributed Autonomous Robotic Systems. (2006) 135–144
3. Howard, A., Siddiqi, S., Sukhatme, G.: An experimental study of localization using wireless ethernet. In: 4th Int. Conf. Field and Service Robotics. (2003)
4. Peng, C., Shen, G., Han, Z., Zhang, Y., Li, Y., Tan., K.: A beepbeep ranging system on mobile phones. In: Conf. Embedded Networked Sensor Systems. (2007)
5. Poduri, S., Sukhatme, G.: Constrained coverage for mobile sensor networks. In: IEEE Int. Conf. Robotics and Automation. (2004) 165–171
6. Priyantha, N., Chakraborty, A., Balakrishnan, H.: The CRICKET location-support system. In: 6th ACM Conf. Mobile Computing and Networking. (2000) 32–43
7. Ziparo, V., Kleiner, A., Nebel, B., Nardi, D.: RFID-based exploration for large robot teams. In: IEEE Int. Conf. Robotics and Automation. (2007) 4606–4613
8. Reich, J., Misra, V., Rubenstein, D., Zussman, G.: Spreadable connected autonomic networks (SCAN). TR CUCS-016-08, CS Dept., Columbia Univ. (2008)
9. Barraquand, J., Langlois, B., Latombe, J.: Numerical potential field techniques for robot path planning. IEEE Trans. Systems, Man, and Cybernetics **22**(2) (1992) 224–241
10. Bennewitz, M., Burgard, W., Thrun, S.: Optimizing schedules for prioritized path planning of multi-robot systems. In: IEEE Int. Conf. on Robotics and Automation. (2001) 27–276
11. Švestka, P., Overmars, M.: Coordinated path planning for multiple robots. Robotics and Autonomous Systems **23** (1998) 125–152
12. Clark, C., Rock, S., Latombe, J.: Dynamic networks for motion planning in multi-robot space systems. In: Proc. Int. Symp. Artificial Intelligence, Robotics and Automation in Space. (2003)
13. Kavraki, L., Švestka, P., Latombe, J., Overmars, M.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. Robotics and Automation **RA-12**(4) (1996) 566–580

Environment Mapping using the Lego Mindstorms NXT and leJOS NXJ

Gerardo Oliveira[1], Ricardo Silva[1], Tiago Lira[1], Luis Paulo Reis[1,2]

[1] FEUP – Faculdade de Engenharia da Universidade do Porto, Portugal
[2] LIACC – Laboratório de Inteligência Artificial e Ciência de Computadores da Universidade do Porto, Portugal
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
{ei04106, ei03087, ei04085, lpreis}@fe.up.pt

Abstract. This paper presents a project of simultaneous localization and mapping (SLAM) of an indoor environment focusing on the use of autonomous mobile robotics. The developed work was implemented using the Lego Mindstorms NXT platform and leJOS NXJ as the programming language. The NXJ consists of an open source project which uses a tiny Java Virtual Machine (JVM) and provides a very powerful API as well the necessary tools to upload code to the NXT brick. In addition, the leJOS NXJ enables efficient programming and easiness of communication due to its support of Bluetooth technology. Thus, exploiting the mobile robotics platform flexibility plus the programming language potential, our goals were successfully achieved by implementing a simple subsumption architecture and using a trigonometry based approach, in order to obtain a mapped representation of the robot's environment.

Keywords: Lego Mindstorms, NXT, leJOS, NXJ, Java, Robotics, Mapping, Subsumption Architecture, Behavior-Based, Client-Server.

1 Introduction

One of the major problems in the field of robotics is known as SLAM (Simultaneous Localization and Mapping) [3], consisting of a technique used by robots and autonomous vehicles to build up a map within an unknown environment while at the same time keeping track of their current position. This is not as straightforward as it might sound due to inherent uncertainties in discerning the robot's relative movement from its various sensors. Thus, the elaboration of a project in this area has proved to be highly motivating for us since “solving” a SLAM problem is considered a notable achievement of the robotics research.

The robotic platform chosen for this project is the Lego Mindstorms NXT [4], a major improvement over their previous platform, the RCX [5]. It contains a more powerful processor, more memory, and employs enhanced sensor and actuator suites, therefore being considered the easiest way to enter in the robotics world.

For programming the NXT robot we decided to use leJOS [1, 6, 16, 17, 18], a firmware replacement for Lego Mindstorms programmable bricks, considered by many as the best platform in the moment to use software engineering ideas. The NXT brick is supported by leJOS NXJ [7], a project that includes a Java Virtual Machine (JVM) [8], so allows Lego Mindstorms robots to be programmed in the Java programming language. The leJOS platform proved to be the best choice for this project since it provides extensive class libraries that support various interesting higher-level functions such as navigation and behavior-based robotics, therefore simplifying the task of robotic piloting through the implementation of a subsumption architecture [9]. Also the API [10] support for the Bluetooth technology was essential in this work since we also implemented a client-server architecture, using the NXT robot as server and sending information to a client application via Bluetooth. Therefore, the whole logic is hosted and executed by the robot, which exchanges information with the client application, running on a computer. The data sent by the NXT is then processed and used in the environment map building.

In order to present the work developed for this project, this paper is organized as follows: Initially, in Sections 2 and 3 the hardware and software platforms are detailed, which is the Lego Mindstorms NXT and the leJOS NXJ, correspondingly. In Section 4 we give an overview of the architectures behind the developed robotic agent, namely the subsumption and client-server architectures. Section 5 details the environment mapping procedures that were used. Finally, in Sections 6 and 7, we give details about the tests we ran and their results, stating the conclusions we achieved and thereby concluding the article.

2 Lego Mindstorms NXT

Lego Mindstorms NXT is a programmable robotics kit released by Lego in late July 2006. The NXT is the brain of a Mindstorms robot. It's an intelligent, computer controlled Lego brick that lets a Mindstorms robot come alive and perform different operations.

Fig. 1. The Lego Mindstorms NXT. In the center, the NXT brick *1*. Above, the three servo motors *6*. Below, the four sensors: touch *2*, sound *3*, light *4* and ultrasonic *5*.

The NXT has three output ports for attaching motors – Ports A, B and C and four input ports for attaching sensors – Ports 1, 2, 3 and 4. It's possible to connect a USB cable to the USB port and download programs from a computer to the NXT (or upload data from the robot to the computer). As an alternative the wireless Bluetooth connection can be used for uploading and downloading.

In the specific case of our project we simply needed two motors (connected at ports A and C), but we used all the input ports for sensors (Ultrasonic Sensor – port 1, HiTechnic [20] Compass Sensor – port 2, HiTechnic Color Sensor – port 3, Sound Sensor – port 4).

3 leJOS NXJ

The name leJOS was conceived by José Solórzano, based on the acronym for Java Operating System (JOS), the name of another operating system for the RCX, legOS, and the Spanish word "lejos".

The leJOS NXJ [7] is a project to develop a Java Virtual Machine (JVM) for Lego Mindstorms NXT, as a replacement of the original firmware of the NXT brick, thereby allowing the NXT robots to be programmed in the Java programming language, providing extensive class libraries that support various interesting higher level functions such as navigation and behavior based robotics.

Started out as a hobby open source project, by José Solórzano in late 1999, the leJOS NXJ has been evolving along the time with new features. Currently leJOS Research Team has launched the release 0.7, available for both Microsoft Windows and Linux/Mac OS X operating systems. This version includes a Windows installer to make installation a breeze for new users. Also, in order to facilitate the tasks of uploading the latest firmware into the NXT brick, uploading NXJ programs to the NXT brick and converting any Java project into NXT project, the leJOS Research Team developed a plug-in for the Eclipse IDE [11], which turn these tasks into a children's play.

From our personal experience with the leJOS NXJ platform, it proved to be an excellent choice, not only for this work, but for any kind of robotic projects using the Lego Mindstorms NXT [1], because of all the offers of the leJOS NXJ, like its preemptive threads, synchronization, multi-dimensional arrays and its well-documented robotics API, for example.

4 Architecture

In this project we made use of two software architectures: subsumption and client-server architectures, which are subsequently detailed.

Subsumption architecture is a methodology for developing artificial intelligence robots. It is heavily associated with behavior-based robotics. The term was introduced by Brooks [2] (1991), which enforces a hierarchy among behaviors so that some of them in certain conditions would inhibit other lower priority ones. A subsumption architecture is a way of decomposing complicated intelligent behavior into many "simple" behavior modules, which are in turn organized into layers. Each layer implements a particular goal of the agent, and higher layers are increasingly more abstract. Each layer's goal subsumes that of the underlying layers. As opposed to more traditional artificial intelligence approaches, subsumption architecture uses a bottom-up design.

The choice of this architecture allowed us to address each type of action independently in different behaviors and give them a set priority. This allows for more organized and alterable code, and avoids problems in the interaction between the different behaviors.

In order to make a better research work, we approached the mapping problem in two fronts, both using a subsumption architecture. Basically one using the compass sensor and another without using it, where calculations are based on the initial position of the robot and its following movements.

For the first approach mentioned above, we used the subsumption architecture to implement a basic navigation mechanism by developing a lowest layer “obstacle avoidance”, which corresponds to the rotate (turn around its center) and retreat (drive backward) behaviors, and on top of it lays the “environment exploration” layer, implemented by a drive forward behavior. Each of these horizontal layers access all of the sensor data and generate actions for the actuators – the main caveat is that separate tasks can suppress (or overrule) inputs or inhibit outputs. This way, the lowest layers can work like fast-adapting mechanisms (e.g. reflexes), while the higher layers work to achieve the overall goal. Feedback is given mainly through the environment.

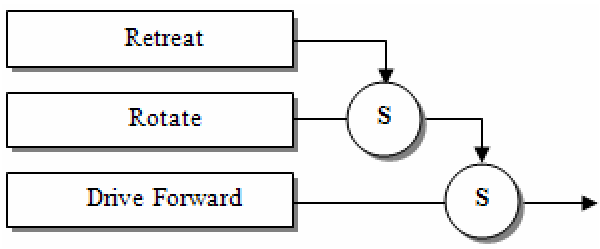


Fig. 2. Representation of the subsumption architecture implemented for the first approach. “Retreat” and “Rotate” correspond to the layer “Obstacle Avoidance” and the “Environment Exploration” is implemented by the “Drive Forward” behavior.

For the second approach, we applied a subsumption architecture also based on the sensor read values, but this time the behaviors are more independent, given that even a behavior with a lower priority will end its turn before another takes place. This happens because the conditions that decide the change of behavior are calculated in the end of that behavior. All the measures and calculations are merged in order to map the environment with all the data collected by that time.

This approach has two main behaviors, the Measure behavior and the Navigate behavior. When on the Measure behavior, the robot rotates 360 degrees and at the same time it measure the distances given by the sonar sensor on the four quadrants. When on the Navigate behavior, the robot drive forward a previous defined distance (for our tests we used distance = 30 units, where each unit correspond to a unit given by the sonar sensor). This way, when we join these two behaviors we get a step-by-step very reasonable map.

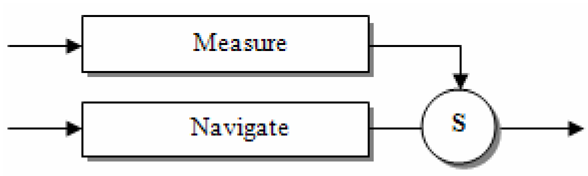


Fig. 3. Representation of the sumbsumption architecture implemented for the second approach. “Measure” and “Navigate” complement each other even if “Measure” has priority over “Navigate”.

This kind of architecture (subsumption) could be implemented by the packages provided by the leJOS NXJ, which has an interface (Behavior) and a class (Arbitrator) to control the whole process:

lejos.subsumption.Behavior

An interface that represents an object embodying a specific behavior belonging to a robot. Each behavior must define the following three methods:

– boolean takeControl()

Returns a boolean value to indicate if this behavior should become active. For example, if the ultrasonic sensor indicates the robot is too close to an object, this method should return true.

– void action()

The code in this method initiates an action when the behavior becomes active. For example, if takeControl() detects the robot is too close to an object, the action() code could make the robot rotate and turn away from the object.

– void suppress()

The code in the suppress() method should immediately terminate the code running in the action() method. The suppress() method can also be used to update any data before this behavior completes.

lejos.subsumption.Arbitrator

The arbitrator regulates which behaviors should be activated, once all the behaviors are defined.

The client-server architecture model distinguishes client systems from server systems, which communicate over a computer network. A client-server application is a distributed system comprising both client and server software.

Thereby, given the fact that the robot doesn't have a great capacity for data processing, not all the calculations were made by the server (NXT brick). In our approaches we needed to send some sensorial data and processed data for a computer client application to better analyze and expose that information. The nature of that data focus on the orientation angle of the robot (obtained with the compass sensor), the distances to the obstacles (acquired by the ultrasonic sensor) and the values of a matrix representing what it "sees", that the robot keeps updating. This process of data exchange between different applications was implemented using the leJOS NXJ, via the Bluetooth wireless protocol.

Fig. 4. The client-server architecture used for communication, by means of the Bluetooth wireless protocol, between the computer application (client) and the NXT brick (server).

While the client server offered a wide range of possibilities, as remote controlled commands for example, we chose to keep the robot as autonomous as possible. Therefore this architecture was used only as a means of reporting observational data to the computer for visualization.

5 Environment Mapping

Robotic mapping [12] has been extensively studied since the 1980s. Mapping can be classified as either metric or topological. A metric approach is one that determines the geometric properties of the environment, while a topological approach is one that determines the relationships of locations of interest in the environment. Mapping can also be classified as either world-centric or robot-centric. World-centric mapping represents the map relative to some fixed coordinate system, while robot-centric mapping represents the map relative to the robot. Robotic mapping continues to be an active research area.

The environment mapping with the Lego Mindstorms NXT discussed in this paper uses a metric, world-centric approach, in order to use the simplest possible mapping algorithm. As described earlier in this document, the robot moves forward from a known starting point until it detects an obstacle with its ultrasonic sensor which is simply avoided by rotating over itself or by driving backward if too close to the object. While doing this exploration of the robot's environment, the NXT brick uses a

80x80 matrix as the representation of the world. A matrix with this dimension covers about 4x4 meters of real terrain, allowing for a more detailed representation of small spaces, while simultaneously maintaining a reasonable size that the limited memory of the robot can support. At the same time, while doing its internal mapping, the NXT transmits its traveled distance (determined with the help of the motors tachometers) and orientation data (obtained with the compass sensor) to the client application running on a computer.

Therefore, our map is built on the robot in runtime, while exploring the world and simultaneously on the client program, since it allows for a much richer map representation than the NXT display.

For the measuring of its own position, the robot registers the value of the tachometers and the direction angle from the compass. In each movement the compass value is frequently updated in an attempt to reduce odometry errors. As the tachometers are much more precise than the compass, this allows the robot to have a reasonable precision in estimating its current location.

Our approach to environment mapping was to use the NXT sensorial data to calculate the absolute position of obstacles. The robot has knowledge of its starting positions and then uses simple trigonometry to calculate its absolute positions and orientations.

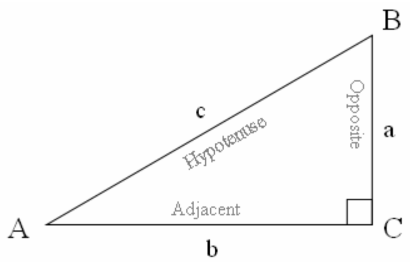


Fig. 5. Representation of the principle used for determining the absolute position of obstacles. In this right triangle: sin A = a/c; cos A = b/c.

With a bit of trigonometry, the absolute position of the detected obstacles can be calculated by using the sine and the cosine trigonometric functions, therefore resulting on the following expressions to determine the *x* (1) e *y* (2) coordinates of a certain obstacle, given the *xi* and *yi* coordinates from the previous robot position as well as the traveled distance (*distance*) and orientation (*α*) of the robot.

$$x = xi + distance \times \cos \alpha . \quad (\mathbf{1})$$

$$y = yi + distance \times \sin \alpha . \quad (\mathbf{2})$$

Thus, the program builds an environment map by calculating the absolute position of obstacles using trigonometry, robot ultrasonic sensor readings and previous calculations of absolute robot positions and orientations.

These calculations allow for sweeping an area, as the robot rotates, in each iteration marking the walls and free spaces in that direction.

Still, the distance sensor and the compass are relatively imprecise, and their results are often mistaken. To make the best of these readings, we used a system of probabilistic mapping based on the Bayes method. In blunt the calculated value is the probability of having a free space in a determined point of the matrix. A low value means high probability of there being an obstacle, while a high value represents a elevated probability of having walkable terrain. Each time a position is tested, the old value is combined with a new one to make an average probability (2). This new value conveys the probability of having a free space or an obstacle, and is relative to the distance. If a position has no valid value in it, it means that position has never been tested before, so we can't use the previous value to calculate the new one. In that case we use a formula where the probability of being occupied is assumed previously (1). We logically defined that the probability of being occupied equals the probability of not being occupied.

$$P(H|s) = (P(s|H) \times P(H)) / (P(s|H) \times P(H) + P(s|{\sim}H) \times P({\sim}H)) . \quad (\mathbf{3})$$

$$P(H|s_n) = (P(s_n|H) \times P(H|s_{n-1})) / (P(s_n|H) \times P(H|s_{n-1}) + P(s_n|{\sim}H) \times P({\sim}H|s_{n-1})) . \quad (\mathbf{4})$$

These formulas gave satisfactory results and were therefore chosen to be the final method. In practice it allows us to take into account diverse factors like distance, repetition of results, and softening of reading errors, resulting in a well balanced and flexible representation of the map.

Additionally, while the robot is designed for static environments, probabilistic mapping allows it to dynamically react to changes in the environment, like adding obstacles or opening new paths.

All these calculations are adapted to be processed on the very limited resources of the robot, which then communicates the final result to the client computer via Bluetooth, for real-time visualization of the mapping process.

6 Results

Despite its limited sensors, the robotic agent is able to create a sufficiently realistic map of its environment.

Fig. 6. Example of an environment map representation. In the left side, the real environment and in the right side the map obtained by the first approach.

To exemplify the mapping capabilities of our agent, we have some results of the MapViewer implemented on the client side, displayed in figure 6.

For the first approach, the compass sensor is easily affected by magnetic fields of nearby objects, and this can lower the quality of the mapping in some places. Another fact that limits this approach is the complexity of coordinate all the movements and all the angles that the robot (compass sensor) can read, since the robot, in this approach can walk to every side in 360 degrees. If the robot gets a wrong read of the compass, it will mark the corresponding position with a wrong value too and walk to a certain position thinking it is another.

Each of the small squares in the visualization window represent a position of the matrix, corresponding to roughly 4.3 centimeters of real terrain in its side. The red points correspond to the trajectory of the robot. The rest represent the probability of obstacles at each position. Orange means high probability of free space, decreasing into shades of yellow. White means a 50% probability of free space, while lower probabilities (probable walls), are represented in shades of gray. Increasingly darker grays, represent greater certainty in the position of the walls.

In general, orange and black spaces will dominate the areas that the robot has seen repeatedly or at a close distance, indicating the parts of the map where risk of error is lowest.

However, our second approach, while being less flexible, works without using the compass, and therefore has no problem in the aforementioned situation. This way we get none of the problems that the compass brings, but we also don't get the advantages, so if the robot rotates too much it will not know and all the following values and map representation will not take in account that error. However, if the robot keeps its navigation almost without errors, a nice and pretty realistic map will be the result of our second approach.

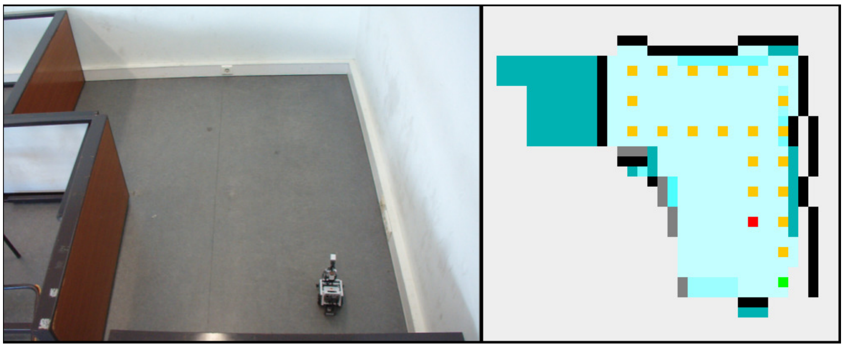


Fig. 7. Example of an environment map representation. In the left side, the real environment and in the right side the map obtained by the second approach.

In this approach, we used different measures and colors to specify the probabilities. Here, each square represent 10 units, where each unit correspond to a unit measured by the sonar sensor. The green spot is the initial position and the red one is the actual position. The brighter zones correspond to locals where the probability of being empty is high (white and brighter blues), while the darker zones correspond to the occupied places (dark cyan to black).

Fig. 8. Example of an environment map representation. In the left side, the map obtained by the first approach, in the middle, the real environment and in the right side the map obtained by the second approach.

While we are satisfied with the results, we realize that they could be vastly improved by using more powerful sonar and compass sensors. The fact of having only one sonar vastly reduces the capacity of analyzing correctly the directions that should be taken when facing an obstacle and the real distances to them or to walls. Another difficulty was the compass errors. Even though the compass often gives good results, a bad read causes a wrong interpretation and correspondingly a false representation of the map.

7 Conclusions

Environment mapping for real robots is a complicated problem, as the system resources are limited and the operation of physical parts (motors and sensors) is highly susceptible to errors and imprecision. The inherent uncertainties in discerning the robot's relative movement from its various sensors turn mapping into a very complex and challenging task. As a matter of fact, a robot can be compared to an insect, for which the environment is not interpreted as a map, and they survive only with a triggered response (e.g. reflexes). Therefore, it was our goal to provide the robot with intelligence in order to combine the information from the past and the present, thus creating a good mapping algorithm, and fortunately we achieved that.

One factor that has contributed to our success was the flexibility of the Lego Mindstorms NXT platform associated with the potential of the leJOS API, which allowed us to program the NXT in the Java programming language, thereby giving us much more control over the robot's sensors and actuators, manifesting itself by a more powerful programming than the one given by the original NXT-G [13] software that Lego provides.

The probabilistic mapping method based on the Bayes' rule proved to be a serious improvement, allowing for flexibility in the creation of the map. The variation of probabilities softens the erratic results, progressively creating a better map.

The two approaches we used allowed us to have different processes of solving a map, and each is advantageous in a certain set of conditions. The first approach is more flexible and gives more detailed results, while the second one is more stable and less sensitive to sensor errors.

As future work, since one of the biggest troubles while trying to “solve” a SLAM problem are the inaccuracies associated with the measured distance and direction travelled, then any features being added to the map will contain corresponding errors. In this manner, if not corrected, these positional errors build cumulatively, grossly distorting the map and therefore the robot's ability to know its precise location. Thus, it would be interesting, in this project’s concern to study techniques to compensate for this, such as recognizing features that it has come across previously and re-skewing recent parts of the map to make sure the two instances of that feature become one. The application in our project of other statistical techniques, such as Kalman filters [14, 19] or Monte Carlo methods [15], for instance, would definitely improve our work.

Acknowledgments

This research was supported by the Portuguese Science and Technology Foundation (FCT) under the project "ACORD – Adaptative Coordination of Robotic Teams" (PTDC/EIA/70695/2006), funded through the POCI/FEDER program.

References

1. Juan Antonio Breña Moral, “Develop leJOS Programs Step by Step”, Version 0.4, July 2008
2. Brooks, R. A., “Intelligence without Reason”, In Proceedings of the 12th International Joint Conference on Artificial Intelligence, pp. 569-595. Sydney, Australia, 1991
3. Cyrill Stachniss, Udo Frese, Giorgio Grisettil, “Simultaneous Localization and Mapping (SLAM)”, Available HTTP: http://openslam.org/
4. The LEGO Group, Lego Mindstorms NXT, Available HTTP: http://mindstorms.lego.com/
5. The LEGO Group, Lego Mindstorms RCX, Available at: http://en.wikipedia.org/wiki/Lego_Mindstorms#RCX
6. Jose Solorzano, “leJOS: Java for LEGO Mindstorms”, Available at: http://lejos.sourceforge.net/
7. Jose Solorzano and others, “leJOS NXJ: NXJ technology”, Available at: http://lejos.sourceforge.net/nxj.php
8. Sun Microsystems, Inc., Java Virtual Machine (JVM), 1994-2009, Available at: http://java.sun.com/docs/books/jvms/
9. Joe Jones, Daniel Roth, “Robot Programming: A Practical Guide to Behavior-Based Robotics”, McGraw-Hill/TAB Electronics, 1st edition, 2003 December 12
10. ComputerWorld, "QuickStudy: Application Programming Interface (API)", Available at: http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=43487
11. The Eclipse Foundation, Eclipse IDE, Available at: http://www.eclipse.org/
12. Andreas Nüchter, “3D Robotic Mapping: The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom”, Springer, 1st edition, 2009 January 22

13.The LEGO Group, NXT-G: LEGO MINDSTORMS NXT Software, Available at: http://mindstorms.lego.com/eng/Overview/NXT_Software.aspx
14.Greg Welch, Gary Bishop, The Kalman Filter: Some tutorials, references, and research, Available at: http://www.cs.unc.edu/~welch/kalman/
15.Christian P. Robert, George Casella, "Monte Carlo Statistical Methods", Springer; 2nd edition, 2005 July 26
16.Brian Bagnall, "Maximum Lego NXT: Building Robots with Java Brains", Variant Press, 2nd edition, 2007 April 30
17.Martijn Boogaarts, Jonathan A. Daudelin, Brian L. Davis, Jim Kelly, Lou Morris, Fay and Rick Rhodes, Matthias Paul Scholz, Christopher R. Smith, Rob Torok, Chris Anderson, "The LEGO MINDSTORMS NXT Idea Book: Design, Invent, and Build", No Starch Press, 2007 August 28
18.Owen Bishop, "Programming Lego Mindstorms NXT", Syngress, 2008 June 9
19.Peter Maybeck, "Stochastic Models, Estimation, and Control", Academic Pr, Volume 1, 1979 June
20.Dataport Systems, Inc, HiTechnic Products, Available HTTP: http://www.hitechnic.com/

NeatSqueak on Wheels: Neural Networks Applied to Movement Optimization

Hugo Peixoto[1,2], João Portela[1,2], Rui Teixeira[1,2], Filipe Castro[1,2], and Luís Paulo Reis[1,3]

[1] Faculdade de Engenharia da Universidade do Porto, Portugal
[2] NIFEUP - Núcleo de Informática da Faculdade de Engenharia da Universidade do Porto, Portugal
[3] LIACC - Laboratório de Inteligência Artificial e Ciência de Computadores da Universidade do Porto, Portugal
Rua Dr. Roberto Frias, s/n 4200-465 Porto PORTUGAL
`{ei03047,hugo.peixoto,ei04034,ei03095,lpreis}@fe.up.pt`
`http://www.fe.up.pt`

Abstract. In *CiberRato* competitions, deliberative agent implementations commonly use path planning algorithms. This creates the need for a path following module, which would control the mouse motors and drive him through a set of calculated waypoints. A common approach is manually defining a decision tree, where the threshold values and motor outputs are empirically determined, obtaining a sub-optimal implementation. We've addressed this problem using optimization techniques based on unsupervised learning. A neural network based solution was developed through the use of genetic algorithms. Comparative data supporting that this approach surpasses manual decision tree implementations is presented for waypoint following behaviours. The results achieved enabled us to conclude that the development platform may be a valuable tool for *CiberRato* competition participants.

Keywords: *MicroRato*, *CiberRato*, Artificial neural network, Genetic algorithm, Movement, Optimization, NEAT, Intelligent Robotics

1 Introduction

CiberRato is a set of competitions, held by the University of Aveiro, composed by a simulation of a virtual environment, in which several agents are located. The environment consists of a 2D rectangle-like arena, which is populated by obstacles and one or more targets, each one signaled by a beacon. The virtual robots are placed in a predetermined starting grid. The competitors must provide software which leads the agents to accomplish a set of predefined goals[1].

The virtual robots' morphology is simple. They're composed of a circular base, equipped with sensors, actuators and command buttons. Information is sent and received through a connection with the simulator, which adds noise to both the sensor readings and the actuator values[1].

In mainstream competitions, the main goal consists of finding one or more targets, which are located in arbitrary positions of the map. The score is usually based on the number of collisions and on the time they take to complete their tasks.

This environment poses a series of challenges regarding the algorithmic nature of autonomous robot control[2]. World mapping, path planning, waypoint following, error control and exploration behaviours can be seen as examples of such challenges.

Most documented agent architectures implemented by the participants of these competitions include a waypoint following module. In reactive architectures, it can be seen that there is a behaviour which moves the agent in the direction of a beacon[3]. In more complex architectures, there are several implementations of various path planning algorithms, which need to be complemented with a behaviour that moves the agent to a defined point[4][5][6][7].

This shows us that the waypoint following behaviour is widely used throughout several architectures, and it is present in almost every agent implementation. This being true, the optimization of this module brings a general increase in most agents' performance.

The architectures described throughout several articles[3][4][5][6][7] explain that the behaviour is usually tuned manually, without the application of an optimization technique, suggesting that the results obtained are sub-optimal solutions.

The only actuators available to the agent are the powers supplied to both left and right engines, represented as real values in the range $[-0.15, 0.15]$. We propose that the optimal values can be calculated from a set of parameters related to the set of waypoints and to the current motors' power. To determine the relationship between these two, we suggest a technique based on artificial neural networks for the optimization of the agents' movement, when following a set of waypoints.

First, the algorithmic background of the explored approach will be shortly exposed. In section 3, the architecture beneath the *CiberRato* neuroevolution platform is described, along with some choices like the fitness function that distinguishes the individuals. Next, section 4 documents the data gathered through the study, along with its analysis. Finally, in section 5, we will discuss the obtained results, detailing both the manual and the machine learning approaches.

2 State of the art

Improving the waypoint following behaviour performance can be achieved using reinforcement learning techniques. The application of these methods is attractive, as there is no need to specify how the task is to be achieved[8].

According to [9], there are two main strategies for solving reinforcement-learning problems. The first one is based on the search in the space of behaviours, trying to find one that performs well. The second approach uses statistic techniques and dynamic programming methods to estimate the utility of each action. We opted for the first strategy, in which evolutionary algorithms are commonly used techniques.

NEAT - Neuroevolution of augmenting topologies - is a neuroevolutionary technique which uses a genetic algorithm for the evolution of artificial neural networks[10], where both topology and connection weights are evolved.

With NEAT, one needs to design and implement a fitness function that evaluates the neural network according to the training goals. As previously stated[8], this technique requires no knowledge of expected network outputs, topology and connection weights.

Radi and Poli[11] showed that NEAT halts its network improvement after a smaller number of generations than other TWEANN[4] algorithms, which leads us to believe that there is no significant gain in evolving a population for a big number of generations, opting instead for recording several runs of the evolution process.

3 Approach

Through the application of the NEAT algorithm, we developed neural networks capable of driving the agent towards a specified waypoint. The agent's motors' power values are directly obtained from the neural network's outputs. This neural network will be refered to as *brain*.

The agent's path following architecture is presented in algorithm 1. The `inputs()` method used in line 7 will be discussed in section 3.2.

```
1  set_target(first_waypoint());
2  while simulation is running do
3      read_sensors();
4      if on top of target waypoint then
5          set_target(next_waypoint());
6      end
7      (leftMotor, rightMotor) = brain.activate(inputs());
8      drive_motors(leftMotor, rightMotor);
9      request_sensors();
10 end
```

Algorithm 1: Basic agent algorithm

[4] TWEANN: Topology and weight evolving artificial neural networks

The evaluation of each brain depended on the number of waypoints that the respective agent could reach within the simulation time limit. For each waypoint reached the agent was rewarded with 1 fitness point. To further distinguish between agents who have reached the same number of waypoints, they were additionally rewarded according to the distance covered towards the next waypoint. This is shown in the following formula.

$$f = N_w + 1 - \frac{D(P_a, P_{w+1})}{D(P_w, P_{w+1})}$$

- N_w - number of waypoints reached
- $D(x, y)$ - distance between point x and y
- P_a - final position of the agent
- P_w - position of the last visited waypoint
- P_{w+1} - position of the next waypoint

3.1 Evolution platform and server configuration

The evaluation of each brain is made through the use of *CiberRato* simulator. In order to reduce the evaluation time of all brains in a generation, we opted to develop a distributed platform specifically for the evolution and evaluation of *CiberRato* brains.

The platform requires the definition of a set of paremeters. First, one must specify the *CiberRato* server configuration. Second, the set of labyrinths in which the brains will be evaluated must also be specified.

Regarding the server configuration, we opted for the standard parameters, with one exception. The server usually adds gaussian noise to the sensors and actuators. In our experiments, this noise was disabled. This decision was made so that this first attempt could focus on the path following problem. An error correction module could be added in further experiments through the use of modular neural networks[12], for instance.

3.2 Test and control groups

There were several parameters which could serve as the neural network inputs. To explore the results that each set of parameters would yield, we designed a set of test groups comprising several input configurations. These groups' characteristics are illustrated in table 1.

Each waypoint is represented by two input nodes to the network, one for the distance and another for the angle. Motor information requires two additional input nodes, one for each previous motor power value. The number of inputs and outputs required by each group is also represented in table 1.

A control group was used, this group implemented a manually tuned waypoint following module — group Upsilon. This implementation is based solely on the angle between the agent's direction and the next waypoint.

	Waypoints	Motor	Input nodes	Output nodes
Alpha	One	No	2	2
Beta	One	Yes	4	2
Gamma	Two	No	4	2
Delta	Two	Yes	6	2

Table 1. Groups' characteristics

3.3 Tests Description

The agent should be capable of optimizing a series of different movements. It should be capable of following a straight path without veering, make sharp and shallow turns clock and counter-clockwise and also reverse it's orientation. A set of training maps were devised in order to evaluate each agents' efficiency performing these tasks.

To validate the generalization of the resulting networks, there was the need to create a map in which several patterns could be analysed. The map waypoints arrangement can be seen in figure 1.

Due to the deterministic nature of the neural network architecture and the absence of noise, the same simulation run more than once will always yield the same results.

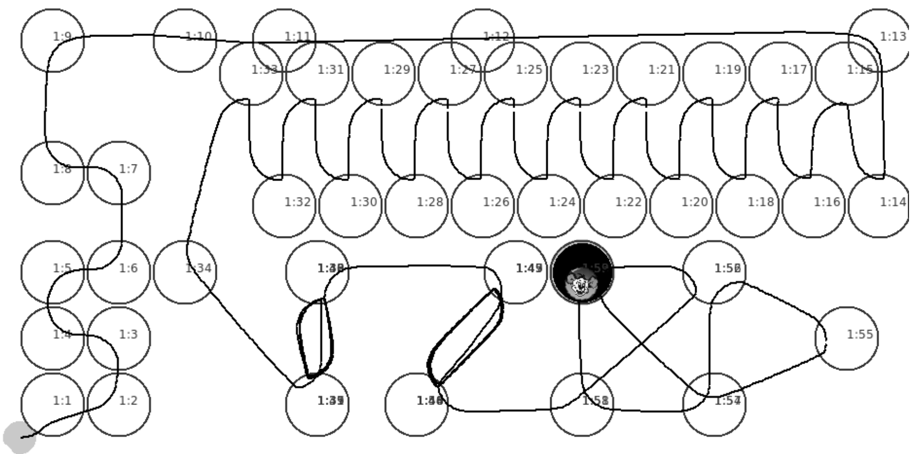


Fig. 1. Validation (Testing Grounds)

3.4 Evolution Methodology

The test groups were all evolved using the same parameters. Their population size was 104 individuals and the evolution process went on for 100 generations. The other genetic algorithm parameters were left untouched, and the ANJI platform default values were used. There are some settings worthy of notice. ANJI's capability to generate recurrent connections in its ANN was disabled, so that the network complexity wouldn't be heightened. We propose that the chosen input values are enough to determine the next simulation state. This being true, the neural network should need no extra information to provide an optimal mapping function between the inputs and the outputs.

The initial topology was set as a fully connected network, which is the simplest network in which all the outputs are dependent on every input.

4 Results

When the neuroevolution runs finished, all the champions were placed on the validation course. For each group, the group champion was defined as the run champion which had the best performance on this map. Their fitness was recorded on table 2. As described by the fitness function, these values represent, approximately, the medium number of waypoints that each brain reached per training map.

	Fitness
Alpha	7.891
Beta	8.197
Gamma	7.973
Delta	7.895

Table 2. Group champions' fitness

Group champions were then evaluated on all training maps and their performance recorded on table 3. The control group, Upsilon, was also evaluated.

	Testing grounds	Map 1	Map 2	Map 3	Map 4
Upsilon	2433	275	273	283	468
Alpha	**1653**	197	200	200	**320**
Beta	1714	**183**	196	197	324
Gamma	1674	187	**189**	**195**	**320**
Delta	1676	189	194	197	325

Table 3. Number of cycles that each group champion took to finish several maps

The fitness evolution was also recorded, as shown in figures 2, 3, 4 and 5. There are three lines in each graph, representing the minimum, average and maximum fitness per generation.

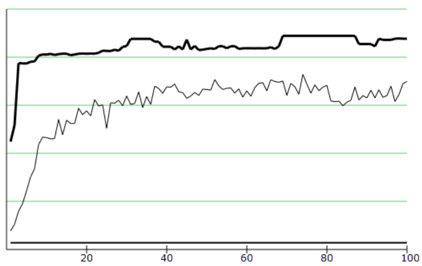


Fig. 2. Alpha fitness evolution graph

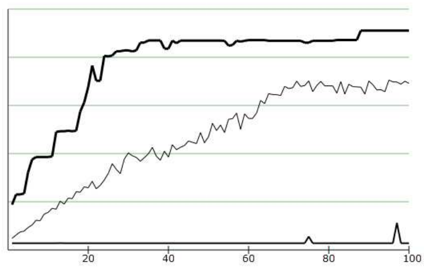


Fig. 3. Beta fitness evolution graph

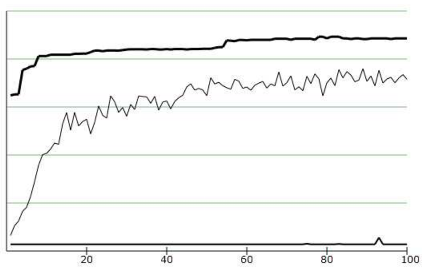


Fig. 4. Gamma fitness evolution graph

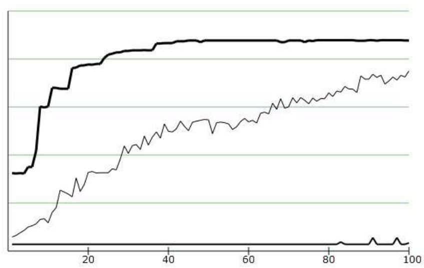


Fig. 5. Delta fitness evolution graph

4.1 Analysis

Observing table 3, it is possible to calculate the average improvement of the test groups upon the control group. The average improvement is $\mu = 30.47\%$, with a variance of $\sigma^2 = 2.5\%^2$ Although the number of samples is considerably small, the improvement is significative.

From figures 2 and 4, one can observe that both alpha and gamma groups have a high fitness improvement around the first 50 generations, and little gain in the following 50. This could be an indication that there is little room for further optimization. It should also be noted that the gamma group, which uses motor speed information, has slightly higher average fitness values. This improvement is possible due to the server's inertia simulation.

Groups with two waypoints, whose fitness evolution is represented in figures 3 and 5, have a significantly slower progress. While beta group seems to stabilize around generation 80, the delta group doesn't appear to achieve a stable position within the 100 generations.

Improvement gained from the addition of the second waypoint information is not evident when alpha and beta groups' average values are compared. Nevertheless, it can be seen that beta groups yielded higher champion values.

As expected, as the neural network base complexity grows, the neuroevolution development time increases accordingly.

5 Conclusions and further study

Optimization in *CiberRato* waypoint following behaviours using artificial neural networks offers significant improvement over manually tuned approaches. Our results support that there is room for approximately 30% gain in the time spent following waypoints.

Considering how little the difference between the first and the second place can be (regarding the *CiberRato* competition), the improvement provided by our proposed module may prove decisive, specially when the agent's architecture is heavily based on waypoint following.

Although some useful observations can already be extracted from this experiment, further statistical data gathering would allow us to develop a correlation between the groups' performance and their network inputs. The number of generations for which the test groups ran didn't prove to be sufficient in the delta (two waypoints with motor information) group. Further study with extra generations could help determine the stabilization point.

The *CiberRato* neuroevolution platform, which integrates the *CiberRato* simulation system and the ANJI platform, can prove to be a valuable asset in further neural network optimization tasks. This will ease the development of further projects, allowing developers to focus on the design of fitness formulas, training courses and other neuroevolution parameters.

The presented study can be extended to integrate obstacle detection and avoidance. This would prove valuable in *CiberRato* competitions, where mapping is usually probabilistic — adding extra flexibility to the proposed behaviour.

Acknowledgements

This work was partially supported by FCT Project PTDC/EIA/70695/2006 "ACORD - Adaptative Coordination of Robotic Teams".

References

1. Nuno Lau and Artur Pereira. Manual do ciberrato. Technical report, Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro, 2008.
2. Nuno Lau, Artur Pereira, Andreia Melo, António Neves, and João Figueiredo. Ciber-rato: Um ambiente de simulação de robots móveis e autónomos. *Revista do DETUA*, 3(7):647–650, 2002.
3. Emmanuel Lomba. R_zero: Um agente virtual "reactivo ma non troppo". *Revista do DETUA*, 3(7):666–669, 2002.
4. Pedro Ribeiro. Yam (yet another mouse) - um robot virtual com planeamento de caminho a longo prazo. *Revista do DETUA*, 3(7):672–674, 2002.
5. Luís Paulo Reis. Ciber-feup - um agente para utilizar o simulador ciber-rato no ensino da inteligência artificial e robótica inteligente. *Revista do DETUA*, 3(7):655–658, 2002.
6. Robin Lilja, Johan Hägg, Batu Akan, Fredrik Ekstrand, Hüseyin Aysan, Jörgen Lidholm, and Moris Benham. Technical description of the (l)ost agent. *CiberMouse@RTSS 2008 - The 29th IEEE Real-Time Systems Symposium*, pages 17–20, 2008.
7. Álvaro Monteiro, Fábio Aguiar, and Sara Carvalho. Speedygonzalez: A path planning and plan execution agent. *CiberMouse@RTSS 2008 - The 29th IEEE Real-Time Systems Symposium*, pages 21–23, 2008.
8. Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Pearson Education, 2003.
9. Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.
10. Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evol. Comput.*, 10(1063-6560):99–127, 2002.
11. Amr Radi and Riccardo Poli. Discovering efficient learning rules for feedforward neural networks using genetic programming. pages 133–159, 2003.
12. Farooq Azam and Farooq Azam. Biologically inspired modular neural networks. Technical report, 2000.

Using Accelerometers to Command a Cleaning Service Robot

Armando Sousa[1,2], Luís Paulo Reis[1,3]

[1] FEUP – Faculty of Engineering, University of Porto, Portugal
[2] INESC-P – Instituto de Engenharia de Sistemas e Computadores do Porto, Portugal
[3] LIACC – Artificial Intelligence and Computer Science laboratory, Porto, Portugal
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
{asousa, lpreis}@fe.up.pt

Abstract. This work studies the effectiveness of using accelerometers to control a service robot. Two modes are proposed, a steering wheel and a movement identification mode. The validation platform is an autonomous cleaning service robot that still needs a local Human Robot Interface. For convenience, accelerometer readings are obtained from the remote command of the Nintendo Wii console. The implementation of the Steering Wheel is based on a Fuzzy Logic controller and the implementation of the movement identifier uses Case Based Reasoning approach upon identified characteristics. Results from experimental tests and user surveys evaluate the effectiveness and usefulness of the presented approach on low skilled users.

Keywords: Human-Robot-Interaction; Service Robot; Robotics; Accelerometers

1 Introduction

With the price of modern day robotics dropping and its usefulness rising, the use of service robotics as human-robot working partnerships will become more common. The search for robotic autonomy is one of the current trends in order to make the robots intelligent and useful to the human society. The notion of usefulness of a service robot includes robustness, mission efficiency and the need for efficient interfaces to command the robot and treat it as a partner in a working team of robots and humans.

The interfacing device to the service robot should be robust, easy to use and preferably inexpensive. As service robotics become more sophisticated, the users of the apparatus are likely to become less skilled thus imposing additional importance on the easiness of usage of the interfacing device.

1.1 Goal of present study

This paper explores accelerometer readings to command a robot, specifically a partially autonomous cleaning service robot. For easiness of data collection, these accelerometer readings were obtained by the Nintendo Wii's [9] remote controller, short named "Wiimote" but many other consumer "gadgets" include accelerometers such as cellular phones among others (a somewhat long list of consumer gadgets with accelerometers is available at [25])

1.2 Structure

After the current introduction, Section 2 deals with related context and research. After that, chapter 3 shows details for the actual implementation. Results are presented in Chapter 4 that led to the conclusions and future work presented in the last chapter.

2. Context and related work

The current chapter will lead to the need to develop the presented work while presenting other interesting research. The work presented here was initially published in [26] and further information can be retrieved at [2].

2.1 Classical Robotic User Interfaces

Classical solutions for Human Machine Interface use a Graphical User Interface (GUI) where the user views commands and often introduces commands through touch screen, joystick, mouse and/or keyboards of several kinds [5]. Frequently these devices were local and wired to the robot.

2.2 Modern interfaces and local command and supervision

Present day autonomous service robots frequently feature some type of wireless communication strategy for development, configuration and eventually remote operation. These operations are typically done in a desktop, laptop or tablet PC frequently using standard Wi-Fi networks that connect to the robot. This type of interface is interesting and is typically done at a distance but has its limitations as some pragmatic commands benefit from locality with the robot, for instance, in a service cleaning robot, a frequent command would be "start cleaning here".

As the service robot is put into present day (or near future) exploration, it is likely that the human worker that teams-up with the robot (or team of robots) tends to be a low skilled person and wants a pragmatic interface to the robot where a few repetitive commands can be issued. The robot or team of robots will probably be cost limited and have a reasonably high level of autonomy but there is still the need for a human supervisor that, besides other tasks, validates the quality of the work done and solves unforeseeable situations like failures, inaccessibilities, etc. Such human worker probably does not want to carry a laptop nor other complex, sensible, expensive and perhaps heavy device.

As mentioned before, it is interesting to have a Local Human Robot Interface (LHRI) to issue a small number of commands that benefit from locality. It thus appears interesting to study the effectiveness of using inexpensive COTS hardware and innovative types of interfaces, even if some limitations are recognized.

2.3 Robotic autonomy and service robots

As autonomy rises, some service robots may pass on extensive Graphical User Interface as Human-Robot communication is less frequent. For an autonomous robot, communication needs include changing mission or operation parameters: redefining mission goals, operation modes or other internal parameters (and such operations are not done frequently).

Changing a small number of parameters may not require a full featured GUI if other interesting interface methods are found. An autonomous Service Robot should be aware of the presence of a human and should accept commands coming from that person if it is a valid supervisor. Present day service robots make up a team of human and robot (or more likely, in future, a team of robots and humans).

2.4 Target Platform – Service Cleaning Robot

The presented work envisages a real world application target that is the Service Cleaning Robot [1]. This robot is built with great flexibility at interesting costs as permanent concerns. It also uses mostly commercial off the shelf (COTS) hardware that is intended to produce a suitable performance at a very interesting costs. At present time, only one prototype is built but the idea of the project includes studying team work of several robots. The prototype is functional and uses wheel chair drive. It includes a fully passive cleaning apparatus purchased at a discount store and adapted to become an integral part of the robot. The processing platform is a common PC and only a limited number of pure distance sensors are available for operation. The project is connected to the structured Wi-Fi network and the robot's state can be viewed and commands issued remotely.

2.5 Needs for the LHRI

Aside from remote operation, the need was identified to have a LHRI for a limited number of commands. Such commands would be given by a human supervisor near the robot. It appears it would be would be interesting to “wave” commands to the robot. The registered needs include easily relocating the robot, starting and stopping missions (for example special cleaning of current area), error recovery, etc [1][26][14]). These needs benefit from a supervisor close to the robot and this means that implementing an effective LHRI is important [21][22].

2.6 Accelerometer background

An adequate solution is “waving” commands with a “magic wand” that leads to registering accelerometer readings over time. With the development of micro electro-mechanical systems (MEMS), accelerometers are becoming more frequent in user grade apparatus (cellular phones, recreational devices, etc). The performance of such devices is increasing to deliver devices of higher and higher sensitivity at very interesting costs [25][11].

Such user “gadgets” offer interesting hardware with very reasonable features and some such devices are very easy to work with because little or no hardware development is necessary. Of course it is necessary to be able to access to the raw accelerometer data easily.

Returning to the notion of “waving” commands at the robot, it would be interesting to have a small, rugged, inexpensive “command stick” with 3D accelerometers, preferably with wireless communications and some means of feedback.

2.7 Wiimote

The Nintendo Wii console has a revolutionary remote control device (abbreviated as “Wiimote”) that is a very interesting solution because it includes a 3D accelerometer device and BlueTooth (BT) communications. It can be considered Commercial off-the-shelf (COTS) as it is widely available at European consumer stores at approximately 40€ (2009 pricing). The Wiimote and some of its interesting features are shown in Figure 1.

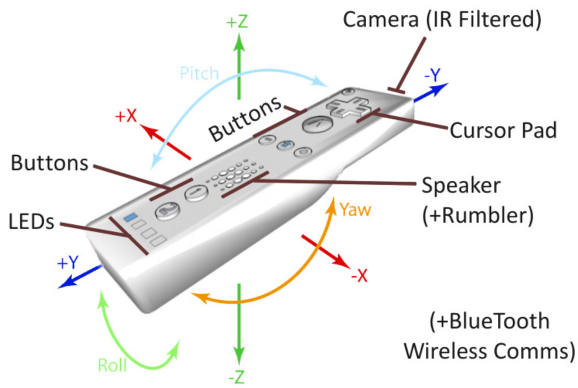


Fig. 1. Wiimote internals and defined axes

Another reason for the ease of use of the Wiimote is that although Nintendo does not release communications protocols, the user community has been successful in reverse engineering portions of the protocols and free and open source interface libraries exist. The used library for the presented work is the libwiimote (see references section).

2.8 The ADXL330 inside the Wiimote

The particular accelerometer chip used inside the Wiimote is the Analog Devices ADXL330 3D accelerometer [3] that has a full range measuring of ±3g, approximately equivalent to $\pm 30 m.s^{-2}$. This device is sampled at 8 bits and samples sent over BT link at approximately 100 reports per second.

The ADXL330 inside the wiimote is able to measure motion. The wimote and associated accelerometer ADXL330 axes are shown in figure 1.

The internal workings of the ADXL330 chip may be very briefly introduced as a set of micro machined springs that suspend a mass of the same kind; accelerations deflect the moving mass and unbalances a differential capacitor that produces measurements resulting in an output whose amplitude is proportional to acceleration. Negligible aging problems and superior temperature immunity are claimed by the maker (see the references section for a link to the datasheet).

The mentioned workings explain that accelerations applied will be measured but if no force is applied, then gravitational force may be measured. Gravitational forces will be measurable as about one sixth of the full range of the sensor.

2.9 State Of The Art

Service robots are becoming common, even in commercial grades (example: “Roomba”) but the configuration and navigation options are very limited and many times fully custom built interfaces [8].

The goal of this paper is to study the usefulness of hand and arm movement detection with the Wiimote’s accelerometer; latter on, the accelerometer is also use as a wireless steering wheel.

The presented ideas may, of course, be adapted for any other types of interfaces such as the ones mentioned in [23]. Adapting movement analysis to a given application or a given service robotic system is generally a matter of imagination to explore additional, not commonly used sensory data that is easily available from the Wiimote as general users are not used to using wireless “command sticks”. Some authors have previously researched gesture recognition for example using tracks on sensitive pads [6], vision systems [20][24] and time of flight cameras [15]. These authors explore creation of interfaces on sensible and expensive systems, less adequate for a working human to use in the command of a robot.

[5] and [16] present interesting work where wireless sensors are read and some conclusions drawn but the current work extends the mentioned works and presents ideas for the platform at hand and additional unskilled user tests. The current work also uses different implementation based on Case Based Reasoning and Fuzzy Logic for the steering wheel application.

3. Implementation

This work explores the potential of accelerometer readings (taken from the Wiimote) as a wireless control structure, exploring a single button and the accelerometers as a consumer grade, widely available device featuring robust mechanics, interesting hardware, inexpensive, lightweight and easy to interface.

3.1 Limitations of the Wiimote

The reader should take into consideration that admittedly using the Wiimote has some limitations. The BlueTooth (BT) standard starts by "pairing" two devices together that must be done at the start of the communication. Only paired devices can communicate. This must be done once whenever a given Wiimote is to control a robot (and this must be done with each robot and each Wiimote). The BT standard plays well alongside with Wi-Fi networks but the implemented protocol does not ensure delivery of all packets, that is, no guarantees about delivering all the reports from the wiimote. The wiimote interfacing libraries are having some trouble using the loudspeaker included in the wiimote when it concerns to playing more than simple beeps. Playing audio samples for user feedback would probably be interesting but is presently not possible. The current implementation takes advantage of the features of the wiimote thus uses only beeps and LEDs for user feedback.

Even in the presence of these limitations, the Wiimote is still interesting to use as Human Robot interface device, specifically in the part of the accelerometer's readings.

3.2 Mode Selection

Current implementation has two modes: wireless steering wheel and movement identifier.

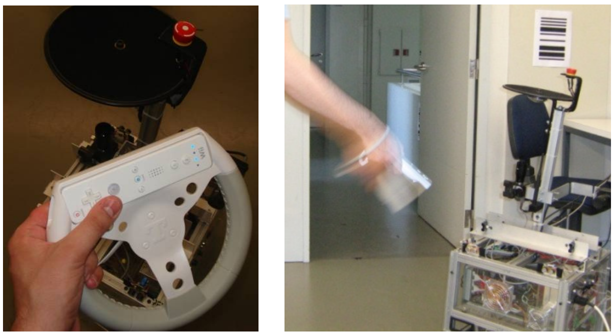


Fig. 2: Left: Wiimote as a Steering Wheel;
Right: User controlling the robot with movements of the Wiimote

In the first place, the controller of the robot was changed in order to integrate communications from the Bluetooth communications stack via the libwiimote library.

If a Wiimote is detected and the “home” key is detected, then the robot suspends previous mission and halts. The user should press buttons “1” and ”2” to awaken the Wiimote and then press the “1” button on Wiimote for steering wheel mode and the “2” button for movement identifier mode.

To inform the user of the selections and actions, each selection is followed by a single medium pitched short sound. Final mode selection that will lead to action is confirmed via a long beep. Erroneous selections activate the rumbler and sound a triple short beep.

3.3 Steering Wheel Mode

This operating mode allows the user to use the Wiimote like if it was an automobile steering wheel and its set of brake and acceleration pedals, in order to control the robot’s linear and angular velocities. The control is done by tilting the Wiimote forward or backwards and/or leftward or rightwards to increase or decrease the linear velocity (υ) and/or increase or decrease the angular velocity (ω), respectively.

As explained before, if the Wiimote is held still, the accelerometer readings determine the direction of the gravitational vector. Using these readings, a wireless steering wheel is produced. This usage of the Wiimote is natural and several games within the console world use this idea; this idea is so common that the market has produced a plastic steering wheel shaped adapter, shown in figure 2 (left).

To start the steering wheel command mode, a centre position is determined to detect each user’s normal way of handling the steering wheel. In this centre position, the robot is stopped. This calibration is done once only, at the start of the usage of this mode.

The steering wheel routines implement a MIMO non linear controller that converts the readings of the 3D gravitational vector into reference values for the robot’s linear velocity υ and angular velocity ω. The conversion process is based on a non-linear Fuzzy Logic inference system [7] based on the Mamdani controller architecture [19]. This approach is interesting due to the non-linearity associated with the analysis of the accelerations on each axis and the difficulty on establishing limits for intuitive operation.

Table 1 presents the rule base for the controller: inputs are the projected components of the vector difference from the gravitational vector to the centre position and the outputs are the robot’s velocities (υ, ω). NHT represents No Horizontal Tilt (horizontal centre) and NVT represents No Vertical Tilt (vertical centre). The implemented rules are very simple and easy to tune and create the notion that horizontal tilt makes the robot turn (controls ω) and vertical yaw alters the velocity of robot (controls υ). All movements are accounted for and it is even possible to make the robot turn over himself.

Table 1. Rules for the steering wheel’s Fuzzy Logic controller;

IF Forward THEN υ_Positive;	IF Leftward THEN ω_Positive;
IF NVT THEN υ_Zero;	IF NHT THEN ω_Zero;
IF Backward THEN υ_Negative;	IF Rightward THEN ω_Negative;

3.4 Movement Identification mode

The other working mode implemented in the LHMI was a movement identifier. In this mode the tridimensional accelerometer is used to read and save the accelerations on each axis in vectors, in order to detect and calculate the characteristics of those accelerations that would make it possible to identify which movement has been done, from a set of thirteen possible movements – an illustration of this is shown in Figure 2 (right).

The movements to be detected are:

- Linear, single stroke, over a single axis (left, right, up, down, front and back)
- Repetitive over a single axis (left and right, up and down and front and back)
- Circular on the xz plane, clock and counter clockwise.
- Square on the xz plane, clock and counter clockwise.

The acceleration's graphic of a linear movement over an axis will be similar to a one-period sine wave, due to the inversion of the acceleration at deceleration instant (braking instant). All movements have recognizable characteristics. Some movements will have more than one instants of acceleration and/or braking, thus the linear movements can be differentiated from the others through the number of maximums and minimums of their acceleration. The direction of a linear movement over an axis can be found analyzing the amplitude of the accelerations maximums and minimums. The orientation of that movement can be found through the analysis of the order in which the maximum and minimum value appear.

Samples readings for the registered accelerometers measurements for the 3 axis are shown in figure 3 for a clockwise square along the XZ plane.

Other movements are identifiable by analyzing the mentioned characteristics of the signals.

The Implementation used Case Based Reasoning (CBR) as presented by [1]. This method is interesting because it is easy to implement, light weight to execute and easy to configure. The motivation behind using this particular method is its simplicity and the expectation that with simple one time training it is possible to achieve good classification results.

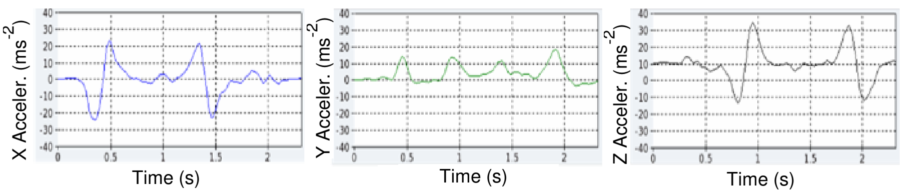


Fig. 3. Sample of accelerometer readings for a clockwise square over the XZ plane.

To distinguish all the movements, a number of very simple and interesting attributes were considered to describe all movements and that are likely to allow distinction of all movements: (i) The number of maximums and minimums of the acceleration in each axis (components); (ii) The global amplitude of maximums and minimums of the amplitude of acceleration 3D vector; (iii) The number of maximums and minimums of the acceleration components compared to the absolute value of the 3D acceleration vector; (iv) The sequence of axis where the components of the acceleration are maximum (over time); (v) The time difference between the samples tagged as maximums and the samples tagged as minimums;

These angles work as pointers to the tridimensional point in which the Wiimote's acceleration is located, and by analyzing the order in which they appear it is possible to know what the Wiimote's trajectory was.

This mode is an improvement of the steering wheel since the robot will be prepared to do a set of complex movements and/or actions that are associated with the movement the user does, thus making the mid/high level actions control simpler. A detected movement would be able to start high level task such as "follow wall to next corner" but implementation is not yet completed.

4. Results

The current chapter presents user feedback and numeric analysis of the previously presented work, improved and further tested than the work presented in [26]. Please note that the users are not the same for all modes. The selected users for the tests are all persons without special computer knowledge and all have never handled the Wiimote before. They would be comparable to the low skilled workers of a service robot.

4.1 User feedback for the Mode Selection operation

The mode selection routine is very simple. Initially it was briefly explained to 14 users. Then the users were asked to rate how hard it was to use: hard, medium or simple. 8 users rated it "Easy to use" and the other 6 "medium easy to use". User opinion's found the rumbler "interesting" and some users wanted more positive feedback for the final mode selection. Understandably, common users that have never used the Wiimote are not used to hearing feedback as beeps and LEDs from a "command stick".

4.2 User feedback for the Steering Wheel mode

The Steering Wheel allowed the user an easy way to control the robot's movement, although it was a little bit hard to stop it due to constant movement, voluntary or involuntary, of the user's hands. To make stopping the robot an easier task it was created a threshold around the υ and ω zero values. To ensure that the control is not triggered involuntarily, its start and stop is done by pressing one of the Wiimote's buttons: the "A" button.

Users were asked to rate this control method depending on the difficulty of it use: hard to use, medium or easy to use. The results were very interesting since all 14 users answered "Easy to Use".

4.3 Results for movement identification

To test the movement identification algorithm it was necessary to compile the case base with examples of all the thirteen movements that we want to identify. In order to be able to perform a complete test of this algorithm, the movements sent into the case base were from only one person.

The commands were thoroughly explained but no rehearsing of the testers was done. As stated before, all users never used a Wiimote before. The results shown in the figure 4 prove that the CBR algorithm and the attributes mentioned before can be used to correctly identify movements made by the testers. Proof of this is the algorithm's capacity to distinguish the simple movement's orientation, the capacity to distinguish squares from circles and in which way they were done.

One of the limitations of working with accelerometer readings is that concerned users use low velocities that generate low readings that, in turn, may result in bad attributes that will make the system answer incorrectly.

The conclusion from this test is that only up and down and left and right is similar for almost all untrained users (over 90% recognition success). This means that this input method for untrained user is of limited success except for the one axis, repetitive stroke movements.

Fig. 4. Correct identification percentage of the CBR algorithm.

5 Conclusions and Future Work

This paper studied accelerometer readings as a candidate device for implementing a Local Human Robot Interface intended for use with a service cleaning robot. The used data was retrieved from the "Wiimote", a promising COTS device. The Wiimote is a feature rich device, containing, among others accelerometers and buttons. Other devices could have been used.

The Steering Wheel uses accelerometers to measure the gravitational vector. This mode allows the user to control the robot like if it was an automobile with separate speed and turning controls in the wireless controller. The steering wheel has a very short learning curve to drive the robot in a demanding environment, like between people working or passing by. Even in this mode, the robot's sensors are still used to prevent bumping into things. This mode was considered easy to use by the users. Fuzzy Logic was used in the implementation of the controller.

The Movement Identifier mode in which the user's movements are identified would allow for a large variety of orders. To identify the movements, a Case Based Reasoning (CBR) algorithm was used. Except for very simple up and down and left and right repetitive movements, this method has limited success ratings in recognizing the movements of untrained users that have never used a Wiimote. Globally, the movement identifier has a 67.7% average success rate on user movements. This success rate is for a case base composed of movements from a single person only.

The global results for using the Wiimote are very satisfactory but additional work is needed to conclude some open issues like providing feedback. To improve the results, the case base should be expanded and it should be added movements made by other people, in order to add greater variety and increase classification rations.

Possible causes for misclassifications is that some users move the elbow a lot more than others and that produces very different measures (example: yawing the command left and right is different from translating the command left and right). Also, some users are afraid and use low speeds that are difficult to read and classify.

The overall conclusion is that accelerometers are interesting devices because they can easily become part of inexpensive, widely available user gadgets but interpreting commands from unskilled, untrained users is a complex task and only the simplest commands are useful.

Future work includes testing new types of movements and adding more both valid and invalid training movements to the case base.

The usage of the wiimote is interesting despite mentioned limitations. Interface improvement is probably possible by using multiple Wiimotes or controlling multiple robots using one Wiimote. The security of the system can be increased by creating a PIN code, composed by a sequence of buttons or a sequence of movements, which unlocks the interface. It would also be very interesting to allow the possibility of using the Wiimote's speaker to give feedback to the user about what the robot is doing with recorded audio.

Acknowledgements

This work has been partially funded by FCT under grant ACORD - Adaptative Coordination of Robotic Teams (PTDC/EIA/70695/2006). The authors wish to thank the Portuguese FCT for the financial support.

References

1. "CleanRob" Autonomous Cleaning Service Robot – http://www.fe.up.pt/cleanrob
2. Aamodt A. and Plaza E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches – In Artificial Intelligence Communications 7, no. 1, pp. 39-52, 1994.

3. ADXL330 http://www.analog.com/static/imported-files/data_sheets/ADXL330.pdf (visited Feb 2009)
5. Cheng Guo and Ehud Sharlin. Exploring the Use of Tangible User Interfaces for Human-Robot Interaction: A Comparative Study. Conference on Human Factors in Computing Systems - Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, Florence, Italy, Pages 121-130 2008
6. Dean Rubine. Specifying gestures by example. Computer Graphics, 25(4):329-337, July 1991.
7. Driankov, D.: An Introduction to Fuzzy Control – Second Edition, Berlin, Springer, 1996.
8. Forlizzi, J and DiSalvo, C. 2006: Interface Design and analysis Service Robots in the Domestic Environment: A Study of theRoomba Vacuum in the Home – In HRI '06, Utah, USA, March 2-4, 2006, ACM Press, New York, USA, 2006, 258-265.
9. Generic Information about the Nintendo Wii http://en.wikipedia.org/wiki/Wii [visited Feb 2009]
11. H. Luo, G. Fedder, and R. Carley 2000, A 1 mG Lateral CMOS-MEMS Accelerometer, In Proceedings of The 13th IEEE International Conference on Micro Electro Mechanical Systems (MEMS '00), January, 2000, pp. 502 - 507.
12. Helge Huttenrauch 2007. From HCI to HRI: Designing Interaction for a Service Robot. Doctoral Thesis, University of Stockholm, Sweeden, 2007
13. J. Schlenzig, E. Hunter, and R. Jain, 1994. Recursive identification of gesture inputs using Hidden Markov Models. Proceedings of the Second Annual Conference on Applications of Computer Vision, December 1994.
14. Kawanura, K., Pack, R.T., Iskarous, M. 1995: Design philosophy for service robots. Proceedings of the 1995 IEEE Conference on Systems, Man, Cybernetics. IEEE Press, 3736-3741
15. Kollorz, E., Penne, J., Hornegger, J., and Barke, A. 2008. Gesture recognition with a Time-Of-Flight camera. Int. Journal on Intelligent. System Technology Appl. 5, 3/4 (Nov. 2008), 334-343.
16. Lapides, P., Sharlin, E., and Costa Sousa, M. 2008. Three dimensional tangible user interface for controlling a robotic team. In Proceedings of the 3rd ACM/IEEE international Conference on Human Robot interaction, Amsterdam, The Netherlands, March 12 - 15, 2008. HRI '08. ACM, New York, NY, pp. 343-350.
17. LibWiimote - Simple Wiimote Library for Linux - http://libwiimote.sourceforge.net/ [visited on January 2009]
19. Mamdani, E.H. and Assilian, S. 1975: An experiment in linguistic synthesis with a fuzzy logic controller - International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.
20. Masaki Shimizu, Takeharu Yoshizuka and Hiroyuki Miyamoto 2007. A gesture recognition system using stereo vision and arm model fitting, International Congress Series, Volume 1301, July 2007, pp. 89-92
21. O. Mayora-Ibarra 2003. From HCI to HRI - Usability Inspection in Multimodal Human-Robotics Interactions, Proceedings of the 2003 IEEE International Workshop on Robot and Human Interactive Communication, 2003.
22. Perzanowski, D., Schultz, A.C., Adams, W., Marsh, E.; Bugajska, M. 2001: Building a Multimodal Human-Robot Interface, Intelligent Systems, IEEE, vol. 16, Issue 1, Jan- pp. 16 – 21, February 2001.
23. Schraft, R. D., Schmierer, G. 2000: Service Robots – A. K. Peters, 2000.
24. Timo Pylvänäinen,. Accelerometer Based Gesture Recognition Using Continuous HMMs. Lecture Notes in Computer Science, Volume 3522/2005 Pattern Recognition and Image Analysis, pp. 639-646
25. Wikipedia Accelerometer - http://en.wikipedia.org/w/index.php?title=Accelerometer&oldid=272179926 (visited Fev 2009, permanent link given)
26. Daniel Coutinho, Armando Sousa, Luís Paulo Reis, Creating a Local Human-Machine Interface for a Cleaning Robot Using the Nintendo Wii's Remote Controller, IROBOT 2008 – 3rd International Workshop on Intelligent Robotics, October 14th, 2008, Lisbon University Institute – ISCTE, Lisbon, Portugal pp. 61-70

Using Barcodes for Robotic Landmarks

Armando Sousa[1,2], Catarina Santiago[1], Paulo Malheiros[1], Paulo Costa[1,2], António Paulo Moreira[1,2]

[1] FEUP – Faculty of Engineering, University of Porto, Portugal
Rua Dr. Roberto Frias, s/n 4200-465 Porto, Portugal
[2] INESC-P – Instituto de Engenharia de Sistemas e Computadores do Porto, Portugal
asousa@fe.up.pt, catarinasantiago@gmail.com, {paulo.malheiros, paco, amoreira}@fe.up.pt

Abstract. This paper presents an innovative vision "sensor". The proposed method uses a vision system for recognizing distant barcodes. It is ideal for Real-Time Self Localization since the proposed algorithm uses light-weight processing of the image obtained by a multi-purpose vision system. The choice of the marker is discussed and the proposed solution uses the industrial barcode standard of Interleaved 2 of 5 to retrieve a measurement of relative position (x, y, z). The angle formed between the camera and the marker is obtained from the "perspective" of the seen image. The results section discusses the measurement data quality for the actual set of experiments. A maximum error of 0.2m was achieved for a camera of 1024x768 with a lens viewing angle of 60º and markers printed on a 14.8cm x 21cm plain paper for localization on a 4x4m world. Typical processing times are between 18 and 25 ms.

Keywords: Localization, Vision, Real Time Vision, BarCode.

1 Introduction

Autonomous mobile robotics is an important research area. True autonomy needs self localization, that is, a mobile autonomous system must find its own localization. Many real world robotic applications feature a semi-structured environment that is mostly unpredictable, possibly dynamic, but where some markers can be placed explicitly to allow robustness and accuracy in the complex task of robotic self localization. Frequent applications demand that the environment must be adequate for both humans and robots as they will share the same environment. This fact places constraints on what kind of markers are admissible for localization. Many other constraints exist for current day robotic applications like limited on-board computing power and Real Time constraints.

Most current robotic platforms feature vision systems because such systems are inexpensive and possess rich sensors from where a number of features can be extracted in the same measurement (a single frame from the image stream). Vision systems are also intrinsically safe to humans and allow for passive, easy to build markers. The task of Real Time image processing creates the need for careful design and implementation of the system hardware and the processing application.

1.1 Test Platform

The presented work is part of an experimental platform for a service cleaning robot, made of effective but inexpensive parts. The mentioned robot is shown in Fig. 1 and features a real consumer grade cleaning apparatus (pushed in front of the robot, yellow and black colours); the cleaning apparatus is passive, that is, the brushes spin upon the unit being pushed forward. Odometry for such a platform is even less reliable than in other applications because of variable friction forces to the ground - the spinning of the brushes on the floor causes the robot bump and twist occasionally. Self recovery from error situations makes clear the need of uniquely identifiable markers for sporadic global localization.

Fig. 1 shows the "CleanRob" platform. The robot features a differential drive, odometry and sonars sensors and a PC motherboard that uses firewire to link to the camera. The camera is meant to serve other purposes besides the localization method described. The robot is to perform autonomous cleaning missions in the faculty corridors that has significant dimensions (200 meters) and many hard-to-distinguish corridors (for additional information, visit the project's web site [17]).

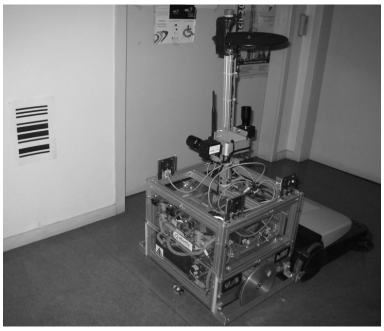


Fig. 1 Robot with front cleaning apparatus and a barcode sample of A4 size.

1.2 Objective

This work presents and characterizes an interesting, inexpensive, robust, and easy to implement global localization technique that uses a Real Time (RT) vision system. The proposed approach produces localization information adequate for RT self-localization on real mobile robotic platforms. The proposed methodology identifies markers and retrieves their relative (x, y, z) position. If the marker position on the world is known, then global localization is achieved.

The system is to be used by the "CleanRob" test platform presented in the previous section. The intent is to have a prototype for a real world autonomous service robot adequate for semi-structured environments of large complexity and size.

1.3 Structure

After the current introductory section that describes the test platform and the objectives, the second section addresses several works that are interesting and comparable to the one presented. The third section builds on the previously presented ideas to list some requirements that, in turn, justify the choices made for the marker design process and then the actual algorithm is presented. Section four presents actual experimental results that lead to the conclusions presented in the following section. Finally, some hints for future work are presented in the sixth and last section.

2 Related Work

The current section focuses on previous research that is related to (artificial) marker self-localization using vision based systems. The general overall goal is to gather as much information as possible from the seen image of the marker. The image is supposed to allow convenient (fast and real time) processing, robustness to false positives and provide accurate data.

Yoon & Kweon (2001) [15], [16] make use of colour image processing to find coloured markers. Scarce data is available about the precision of the resulting measurements but the method is tailored for small environments. This methodology is interesting because it tries to eliminate false positives by using colour combinations and seeks a number of different, identifiable markers.

A localization technique based on a pair of coloured rectangles is used by Jang et al (2002) [10]. The system uses colour processing and controlled pan and tilt to target the marker and minimize camera and lens distortions which is said to improve the accuracy of the measurements. Some results are shown on a 2.4m x 3m world with errors less than 20cm.

A feasible approach is to use industrial barcode readers to detect presence of markers. Distance reading is possible with industrial grade sensors but this technique has very limited usage for self-localization purposes as angles are unavailable.

Jörg *et al* (1999) [11] does, in fact, make use of barcodes to guide a robot based on the recognition but the authors don't present many results regarding the quality of the devised perception system but rather study the methodology of the readings.

Other approaches for visual markers include using coloured poles (Sousa et al. 2005) [13], balls, lines, corner lines, etc (Iocchi & Nardi 1999 [8]; Göhring & Burkhard 2006 [4]).

3 System Design

The proposed goal is to use a vision system to detect and identify markers and retrieve relative distance and relative angle of vision to the seen marker. If the camera is on-board the robotic system and marker's world position is known, then global localization information is available. Localization information has soft Real Time characteristics, that is, the usefulness of the information decays over time.

3.1 Vision System Hardware

The vision system is made by an Imaging Source Inc camera that is connected by firewire to the main board of the PC that consists in the brain of the robot. These cameras are adequate for real time measurements because they use very little internal buffering. The actual camera used is a DMK31AF03, with a resolution of 1024 x 768 pixels black and white camera where each pixel can have 256 grey levels.

3.2 Marker Design

Taking into consideration the application mentioned in the first section, the robot is to self-localize in a world of relatively large dimensions with a large series of hard to distinguish corridors and hallways. The idea of adding localization markers to the environment seems an interesting solution for current day real world autonomous robotic applications. The design of the marker structure itself must be carefully selected in order to solve the problem of localization adequately (Yoon & Kweon, 2002 [15], [16]).

Based on the experience taken from the robotic platform presented earlier, the marker design should be based on the following identified requirements: passive, inexpensive, unobtrusive (thin, etc.), human safe (eye safe, etc.), provide quality information, adequate for RT detection using the vision system and for reading with low computational cost, easy implementation (high repeatability) and deployment, standardized shape, provide robust measurements and allow for “many” different, identifiable, markers.

As mentioned earlier, colour image processing is prone to having problems with lighting changes and colour decaying through aging. Real world applications have to deal with a variety of lighting conditions ranging from artificial lights to different colours of the sun (or sun passing through coloured shades, etc.).

A large number of markers are essential to make up for possible partial or total occlusion on a dynamical world of large dimensions.

The mentioned issues lead to the conclusion that the marker should simply be a printed sheet of paper with adequate Black and White coding. Industrial barcodes may be easily shaped in such a way. Such systems use ingenious and proven methods to encode simple numeric information that points to a very interesting solution. Ad-Hoc barcodes for localization have previously been used by Costa *et al.* (2001) [3] where a radial “bar” code is used in order to produce external localization (position and angle) of a number of identifiable robots.

3.3 Select Barcode Technology

There are a number of industrial barcode standards available. For simplicity and robustness, a well known, unidimensional encoding technique was used as a base for the proposed system. The encoding technique is known as interleaved 2 of 5 code (for general knowledge see Wikipedia_i2of5 [14]) as defined by ISO/IEC 16390:2007 [9], [7] and ANSI / AIM [1], [2] in the Uniform Symbology Specification (for additional barcode knowledge refer to Pavlidis *et al.* 1990 [12]). The mentioned standard encodes a decimal number into bars that are easy to read with a linear optical scanner, even in the presence of slight misalignments. The proposed technique achieves dense information encoding yet retaining robustness and flexibility. The name of the code derives from the fact that each encoded digit uses 5 bars of alternating colour (black/white), each of which can be either narrow or wide. The standard also stipulates quiet zones, start sequence and stop sequence that allow calibration of the size of the bars, thresholds on how to identify narrow or wide bars and still provide a high level of immunity to noise and partial occlusion. An example of such encoding is shown on Fig. 2 (Fig. 1 also showed a barcode marker placed on the wall).

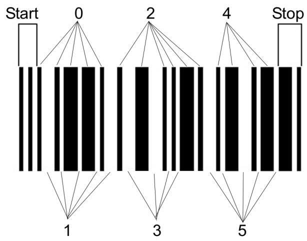


Fig. 2 Example of barcode showing individual digits for “012345” content

3.4 Application Strategy

Some elementary ground rules for Real Time image processing were obeyed in developing the measurement application such as zero copy and one pass processing. Only after all the processing is done, if time is still available, data is fed back to the user screen.

The goal of the image processing algorithm is to measure accurately and efficiently the relative position and angle to the top left of the barcode, as shown in Fig. 3.

3.5 Image Processing Algorithm

Vision systems produce a stream of images. In order to map bi-dimensional (u,v) image coordinates into real 3D world coordinates, the camera model has to be determined. The well known pin-hole camera model (Hecht, 2001 [6]; Gonzalez and Woods, 2007 [5]) is used to that purpose. There is also the non linear distortion commonly called barrel distortion that has to be modelled. Vision sensing systems include intrinsic and extrinsic parameters.

Intrinsic parameters to the vision system are: resolution, pixel size, focal distance, lens centre and barrel distortion. Extrinsic parameters are the position of the camera in the 3D world and the associated orientation angles. For simplicity, the camera is assumed to be in the origin, viewing axis aligned with x axis as shown in Fig. 3.

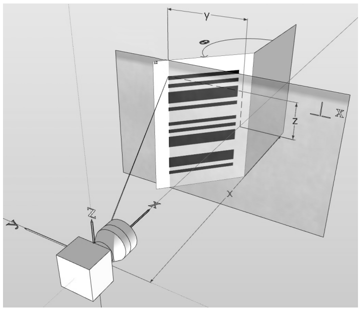


Fig. 3 Camera axes

The vision system model parameters were estimated using common calibration techniques that involve mapping real points in space and matching points on the screen and then optimizing model parameters.

For real world applications, it is interesting to use (inexpensive) wide angle lens that carry important distortion along the image. That distortion is called barrel distortion and a sample image is shown in Fig. 4.

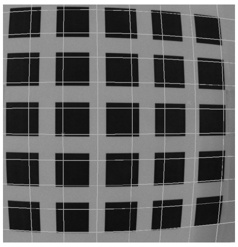


Fig. 4 Test image showing barrel distortion (for the actual camera and lens used).

A few assumptions were made to speed up the barcode search: the barcodes are placed in a vertical wall with horizontal stripes, the camera horizontal optical axis is horizontal and the camera vertical axis is aligned with world vertical.

The full algorithm can be summed up as:

1. Cycle all the configured vertical *scan lines* to find *candidate regions* – a quiet zone followed by two elementary digits (n_1,n_2); if found a *candidate region*, then produce temporary additional *scan lines* for accuracy; for each *candidate region* with marker digits (n_1,n_2), keep start pixel and end pixel coordinates;
2. Group viable *candidate regions* into *markers* – for each of the found viable marker numbers (n_1,n_2), keep only the coordinates of the 4 outer corners;
3. For all *markers*, undistort the coordinates of the corners;
4. Calculate angles using pin-hole model and relative world distance and position using known marker dimensions;
5. Estimate angle information;

Additional information for these steps appears later on, in the current section.

The left image of Fig. 5 shows a barcode and configured scan lines (note that scan lines are more frequent in the interesting region) and the right image the grey levels for a single scan line, including the upper and lower bounds to allow for black/white threshold calculations. The actual threshold value is dynamically calculated to allow for shadows, etc..

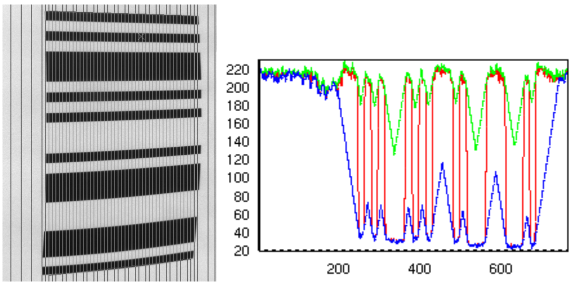


Fig. 5 Barcode maker and scan lines (left) and grey levels along one scan line (right)

The used code has a standard minimum quiet zone and a fixed start sequence that eases the recognition of a valid sequence of digits. The presented algorithm can be extended to accommodate for additional marker digits – to allow for a larger number of different marker numbers.

Once a *candidate region* is found, the coordinates of the barcode top left pixel (u_i, v_i) and bottom left pixel (u_i, v_f) are found, the world position, relative to the camera frame, of the barcode left point can be found. New coordinates (u_{tl}, v_{tl}) and (u_{bl}, v_{bl}) can be found applying the barrel un-distortion formula to eliminate the source of error.

$$\begin{aligned} \Delta_u &= u - u_c \\ \Delta_v &= v - v_c \\ d^2 &= \Delta_u^2 + \Delta_v^2 \\ u_{nd} &= \Delta_u * \left(1 + K_b * d^2\right) + u_c \\ v_{nd} &= \Delta_v * \left(1 + K_c * d^2\right) + v_c \end{aligned} \quad . \tag{1}$$

, where K_b is the barrel undistortion parameter; (u_c, v_c) is the centre of the lens that generates the barrel distortion; pixels of coordinates (u_{nd},v_{nd}) make up an undistorted image from the original image.

Using equations (2) the world angles (α, β) of the half lines that start at the camera centre and pass over the points C_{tl} (u_{tl}, v_{tl}) and C_{bl} (u_{bl}, v_{bl}) can be found (see left image of Fig. 6).

$$\begin{aligned} \alpha &= atan2\left(I_w/2 - u, d_{eq}\right) \\ \beta &= atan2\left(I_h/2 - v, \sqrt{\left(d_{eq}\right)^2 + \left(I_w/2 - v\right)^2}\right) \end{aligned} \quad . \tag{2}$$

, where I_w and I_h are the image width and height; d_{eq} is the distance from the origin to the virtual image plane – these three parameters are intrinsic to the vision system.

The world position for point C_{tl} can now be found by

$$\begin{gathered} \Delta = \tan(\rho_{tl}) - \tan(\rho_{bl}) \\ \Delta_\rho = C_s / \Delta \\ \Delta_{xw} = \Delta_\rho * \cos(\theta_{tl}) \\ \Delta_{yw} = \Delta_\rho * \sin(\theta_{tl}) \\ \Delta_{zw} = \Delta_\rho * \tan(\rho_{tl}) \end{gathered} \quad . \tag{3}$$

, where C_s is the marker vertical size (an input to this algorithm).

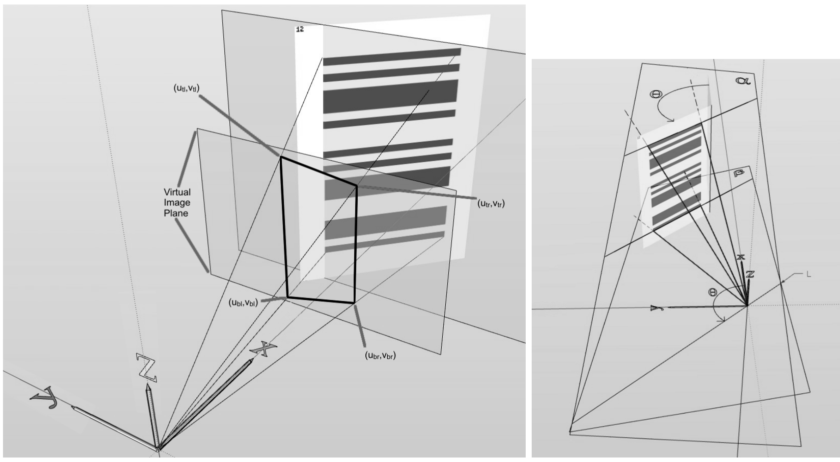


Fig. 6 Perspective of seen marker on virtual image plane (left image) and planes for angle estimation (right image)

To find the relative angle of vision between camera and barcode, "perspective" information is used. The planes α and β are defined by three points each: C_{tl}, C_{tr} and the origin for α and C_{bl}, C_{br} and the origin for β (see right image Fig. 6).

The intersection of these two planes is a line parallel to the line given by the intersection of the barcode plane and the horizontal plane. The angle between this line and the x axis gives us the barcode orientation. A vector v_l with the same orientation as this line can fount by

$$v_l = v_\alpha \times v_\beta \ . \tag{4}$$

, where v_α and v_β are vectors orthogonal to the planes α and β.

Finding the estimate for the angle of the marker concludes the discussion of the algorithm.

4 Results

This section presents the experiments made and characterizes results.

4.1 Experimental Setup

The used setup is based on an Imaging Source DMK31AF03 camera with a resolution of 1024 by 768 with 256 grey scale levels. The camera with the fitted lens has a viewing angle of about 60 degrees with significant barrel distortion. The markers tested are simply sheets of paper with the barcode printed by a common consumer grade laser printer. The size of the marker (including mandatory quiet areas) is A5 (14.8cm by 21cm).

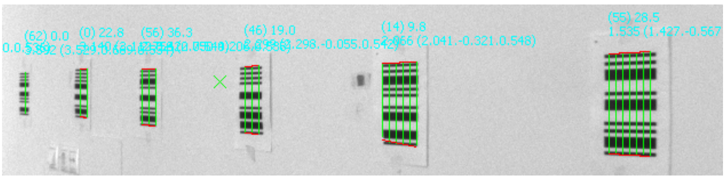


Fig. 7 Application showing 6 barcode markers with A5 size on the test wall

A partial screen shot from the localization application showing the image seen by the camera is on Fig. 7.

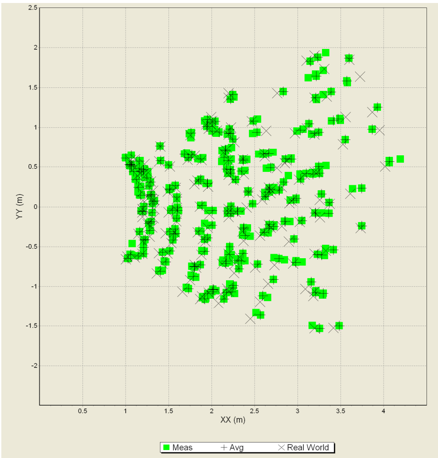


Fig. 8 Measurement dispersion for experiments: smallest and lighter arrows are individual measurements and intermediate sized blue arrows are the average of the readings; red arrows are real world poses.

Several barcodes are identified, from right to left, code numbers 55, 14, 46 and so on. Fig. 1 showed the robot with an A4 sized marker on the wall. To allow for efficient data gathering the test landscape has 6 codes and the robot with the camera moves to several manually selected poses to gather data form the presented algorithm.

The robot is carefully hand placed with the help of measuring tools, such as civil constructor's laser line generators and a large size angle meter (a custom made protractor). Experimental data is obtained by moving the robot whilst the barcodes are always in the same position - the wall (as seen in Fig. 7). Each experiment has a different known camera position and angle such that barcodes are read at different relative distances and angles. For additional ease of interpretation of the data, the results are shown in the form of a camera situated at the origin and calculating the errors of the several seen barcodes.

Fig. 8 shows the global distribution of the entire set of results, also room space presenting limitations and consequences of the used methodology for data gathering.

Each experiment has its own robot pose and is 256 measurements were taken for each experiment. Each frame can have any number of seen markers but faraway markers may become unreadable typically due to the smallest bar of the marker becoming invisible. A total in the order of 22000 frames were analyzed and more than 80000 markers were found and measurements logged.

4.2 Analysis

Fig. 8 hints that accuracy for the proposed measurement is interesting. This fact can be further ascertained by Fig. 9 that presents the accuracy of the measurements taken as a function of the number of detected lines in a given marker measurement. For the tests at hand, the z position of the marker is known and is not useful for self localization. Some angle information is also available from the perspective of the seen image of the barcode.

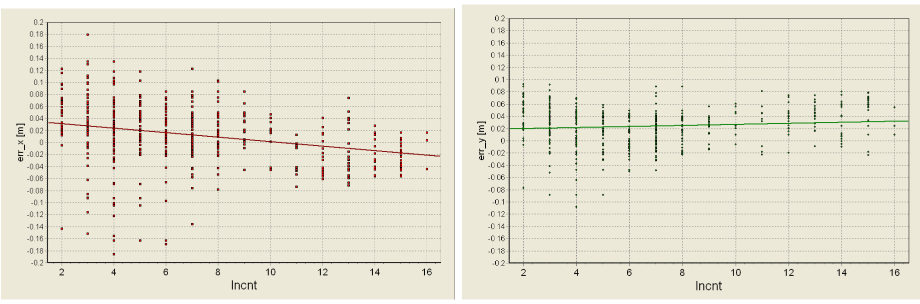


Fig. 9 Error in X over scan line counts (left) and error in Y over scan line counts (right)

For all measurements taken, the global accuracy of the proposed methodology is very interesting: all measurements have position errors in xx less than 0.2m (over 4 meters), in yy less then 0.11m (over 3.5 meters). Angle information is not as reliable and only 90% of the measurements have errors of less than 20º.

5 Conclusions

The presented work shows an innovative approach for landmark localization. The requirements that led to the development of the presented algorithm are discussed in detail because they are inspired in the operation of a real robotic testing platform.

The proposed methodology features inexpensive, unobtrusive, and simple to produce markers that carry numeric information. The presented system can handle 100 unique markers but the ideas presented can be easily extended to larger ranges.

Two applications were implemented: one produces the markers according to the industrial standard of Interleaved 2 of 5 barcode; the other runs in the embedded PC. Both applications can run in any common Linux distribution.

The system is naturally portable to other vision hardware but the used setup is based on an Imaging Source DMK31AF03 camera with 1024 by 768 resolution (256 grey scale levels) that connects via firewire link to a common PC motherboard with an AMD Athlon 3000+ MHz CPU. The camera with the fitted lens has a viewing angle of about 60 degrees with significant barrel distortion. The markers used for the experiments are simply sheets of white paper printed on a regular laser printer. The size of the marker (including mandatory quiet areas) is A5 (14.8cm by 21cm). The approach taken is to use Black and White image processing that is expected to handle well significant differences in lighting situations.

The test setup was a wall with 6 barcodes and the experiments were made moving the test robotic platform. During the experiments more than 80000 measurements were logged in an area of about 4x4m and the accuracy was always better than ±0.2meters. Most measurements (90 % of the times) have accuracy of ±0.1meter and ±20 degrees. Any number of markers can be identified in a single image.

Processing times are around 18 ms and rise up to 25 ms when there are many barcodes candidates within the 40 scan lines. It is expected that the overall time lag from reality to result to be at most 0.07 second which is adequate for a normal paced service robot to be usable in real time self localization.

The proposed approach also detects the number encoded on each marker and by knowing the marker's world position, then, the presented system is able to do Global Real Time Self Localization as was stated initially.

6 Future Work

The quality of the measurements can be further improved with additional searching of the image, of course, at the expense of additional computing power/processing time.

Several techniques are to be studied further like implementing scan line adaptive density, get a better accuracy of the marker envelope using full sub pixel resolution and use a lens with a motorized zoom for additional precision.

Another issue is to study the effect of the rotation of the marker and other angular effects on the final accuracy of the measurements as well as the effect of shadows and alterations in ambient lighting in the accuracy of the measurements.

Taking into consideration that the final goal of the proposed approach is self-localization, it then appears that changing the marker size and shape will probably impact heavily on the results so it would be interesting to test other marker sizes and derive a model for this kind of measurement. Additional work will also be dedicated to using data fusion techniques to a set of measurements that include various measurements of the approach proposed here.

Acknowledgments. This work has been partially funded by the Department of Electrical and Computer Engineering of the Faculty of Engineering, University of Porto (FEUP), under contract DEEC-2004 and partially by FCT under grant ACORD - Adaptative Coordination of Robotic Teams (PTDC/EIA/70695/2006). The authors wish to thank DEEC and Portuguese FCT for the financial support.

References

1. AIM – Association for Automatic Identification and Mobility, http://www.aimglobal.org
2. ANSI – American National Standard Institute, http://ansi.org
3. Costa P., Moreira A., Marques P., Costa P., Sousa A., Gaio S., 2001. 5dpo Team Description. In RoboCup games and conferences Andreas Birk, Silvia Coradeschi, Stoshi Tadokoro, RoboCup 2001: Robot Soccer World Cup V, pp.-, 2001
4. Göhring, D. & Burkhard, H., 2006. Multi Robot Object Tracking and Self Localization Using Visual Percept Relations. In: Proceedings of IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS) 2006, pp 31-36
5. Gonzalez R. & Woods R., 2007. Digital Image Processing, (3rd Edition). Prentice Hall. ISBN 978-0131687288
6. Hecht, E., 2001. Optics (4th Edition), Addison Wesley, 2001, ISBN:0805385665
7. IEC – International Engineering Consortium, http://www.iec.org
8. Iocchi L. & Nardi D., 1999. Self-localization in the RoboCup environment. In Third International Workshop on RoboCup (Robot World Cup Soccer Games and Conferences)
9. ISO – International Organization for Standardization - http://www.iso.org
10. Jang, G., Kim, S., Lee, Lee, W. & Kweon, I., 2002. Colour Landmark Based Self-Localization for Indoor Mobile Robots. In the proceedings of the ICRA 2002, IEEE International Conference on Robotics and Automation vol. 1, pp. 1037 – 1042
11. Jörg, K., Torsten, G. & Weber, J., 1999. Supporting Mobile Robot Localization by Visual Bar Code Recognition. IASTED Conference on Robotics and Applications (RA'99) Santa Barbara, California, USA
12. Pavlidis, T., Swartz, J. & Wang, Y., 1990. Fundamentals of Bar Code Information Theory, IEEE Computer Journal, vol. 23, no. 4, pp. 74-86, Apr., 1990
13. Sousa, A., Costa, P., Moreira, A., Carvalho, A., 2005. Self Localization of an Autonomous Robot: Using an EKF to merge Odometry and Vision based Landmarks. In the proceedings of the 10th IEEE International Conference on Emerging Technologies and Factory Automation, pp.227-234
14. Wikipedia_i2of5 - Wikipedia BarCode Interleaved 2 of 5, http://en.wikipedia.org/w/index.php?title=Interleaved_2_of_5&oldid=200950623
15. Yoon, K. & Kweon, I, 2002. Landmark design and real-time landmark tracking for mobile robot localization. Mobile Robots XVI, Douglas W. Gage, Howie M. Choset, Editors, Proceedings of SPIE Vol. 4573 (2002)
16. Yoon, K. & Kweon, I., 2001. Artificial Landmark Tracking Based on the Colour Histogram. In the proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), vol. 4, pp. 1918-1923
17. CleanRob Project – CleanRob Project web site , http://www.fe.up.pt/cleanrob

Chapter 8

KDBI - Knowledge Discovery and Business Intelligence

Exploiting Generalized Association Rules*

Marcos Aurélio Domingues[1] and Solange Oliveira Rezende[2]

[1] LIAAD-INESC Porto L.A. – University of Porto
[2] ICMC – University of Sao Paulo
marcos@liaad.up.pt, solange@icmc.usp.br

Abstract. Association Rules is a Data Mining technique frequently used for decision support in market basket analysis, marketing, retail, and so forth. However, the mining of association rules may generate large quantities of patterns, complicating the patterns analysis. An approach that can help the analysis of the association rules is the use of taxonomies. In this paper we propose a system that uses taxonomies to generalize association rules and to analyze the generalized rules.

1 Introduction

The problem of mining association rules was introduced in [2]. Given a set of transactions, where each transaction is a set of literals, called items, an association rule is represented like an expression $LHS \Rightarrow RHS$. The LHS and RHS are, respectively, the *Left Hand Side* and the *Right Hand Side* of the rule, defined by distinct sets of items. The intuitive meaning of such a rule is that transactions in the database which contain the items in LHS tend to also contain the items in RHS. So, the association rules are used to find out the tendency that allows the user to understand and exploit the behavior patterns of the data. An example of such a rule might be that 80% of the customers who purchase the Q product also buy the W product. Here 80% is called the confidence of the rule.

The association rules technique has caught the attention of companies and research centers. Several researches have been carried out with this technique and the results have been used by companies to improve their businesses (marketing, insurance policy, demographics). However the use of association rules may generate large quantities of patterns, complicating the patterns analysis.

An approach to solve the problem of large quantities of patterns, extracted by the association rules technique, is the use of taxonomies [1]. The taxonomies can be used to prune uninteresting and/or redundant rules (patterns). In this paper we propose a system that uses taxonomies to generalize association rules and to analyze the generalized rules.

2 A System for Exploiting Generalized Association Rules

The problem of mining association rules is to find all rules that satisfy a user-specified minimum support and minimum confidence [2]. In most cases, tax-

* Thanks to FCT, Portugal (SFRH/BD/22516/2005) and CNPq, Brazil.

onomies (*is-a* hierarchies) over the items of association rules are available. An example of a taxonomy is presented in Fig. 1. This taxonomy says that t-shirt *is-a* light cloth, short *is-a* light cloth, light cloth *is-a* sport cloth, etc. Generalized association rules generate rules that span different levels of the taxonomy. For example, we can infer a rule that people who buy light cloth tend to buy tennis (light cloth ⇒ tennis) from the fact that people bought t-shirt with tennis (t-shirt ⇒ tennis) and short with tennis (short ⇒ tennis).

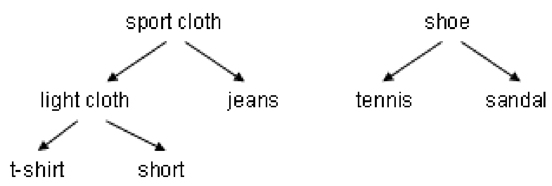


Fig. 1. Example of a taxonomy.

To make use of taxonomies to generalize association rules and to analyze the generalized rules, we propose the system $\mathcal{ENGAR}$ (Environment for Generalization and Analysis of Association Rules). This system is a desktop version of the Web module for generalized association rules, *RulEE-GAR*, proposed in [3]. It was developed due to the low performance of the Web module in processing big data sets. The system (Fig. 2) is composed for 3 modules: Data Entry, Generalization and Analysis.

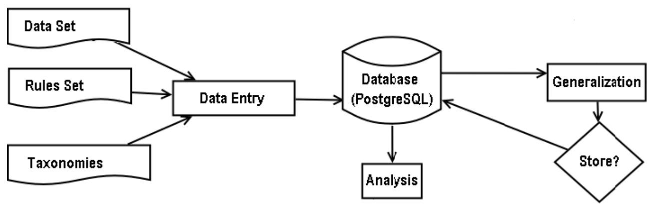


Fig. 2. Architecture of the system $\mathcal{ENGAR}$.

The Data Entry module is responsible for loading the text files containing the data set, rules set and taxonomies, to be used by the other modules. The Generalization module is the part of the system responsible for executing the algorithms of generalization of association rules. The current release of the system uses the $\mathcal{GART}$ algorithm proposed in [3]. Finally, the third module contemplates the functionality of analysis of the system, where a set of rules, previously generalized and stored in the system, can be evaluated through some kinds of analysis, for example, exploring methods [3] (to visualize the generalized rules expanded, the regular rules that were generalized, and so forth) and quality measures [4] (confidence, support, correlation, lift, laplace, etc). In Fig. 3 we show a screen of the system $\mathcal{ENGAR}$, where we can generalize and exploit the association rules.

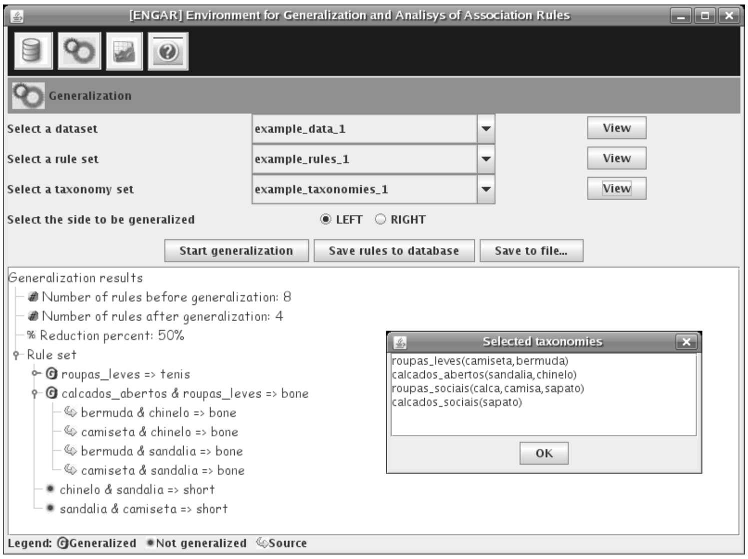


Fig. 3. A screen of the system $\mathcal{ENGAR}$.

3 Case Study

We applied our system in a sales data set of a small supermarket. The database contained sales data of 3 month. We made 5 partitions of the data set to carry out this analysis. The partitions contained sales data along of 1 day (32668 rules generated), 7 days (19166 rules generated), 14 days (16053 rules generated), 1 month (21505 rules generated) and 3 months (19936 rules generated). The rules sets were generated using the *Apriori* algorithm with minimum support value equal 0.5, minimum confidence equal 0.5 and a maximum number of 5 items by rule. We also asked to an expert to make 18 different sets of taxonomies.

We ran the $\mathcal{GART}$ algorithm combining each set of taxonomies with each set of rules. In Fig. 4, a chart shows the reduction rates of the 5 rules sets after running the $\mathcal{GART}$ algorithm, using the 18 sets of taxonomies, to generalize each rules set. In Fig. 4, the sets of taxonomies are called "T" followed by an identification number, as for example: T01. The reduction rates go from 14,61% to 50,11%.

Now the generalized rules can be analyzed using the system $\mathcal{ENGAR}$ and the results, for example, can be used to change the layout of the supermarket. For the sake of confidentiality, we can not show other results.

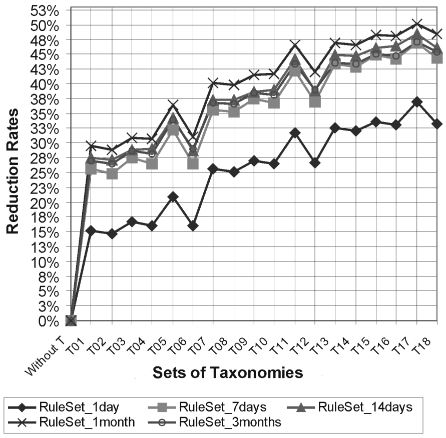


Fig. 4. Reduction rates using taxonomies to generalize association rules.

4 Conclusions and Future Work

In this paper we proposed a system that uses taxonomies to generalize association rules and to analyze the generalized rules. The taxonomies can be used to prune uninteresting and/or redundant rules, facilitating the analysis of large rules sets.

In [5], Li Yang uses parallel coordinates to visualize regular and generalized association rules. Our proposal is not so sophisticated as the one proposed in [5]. However our system provides other functionalities to exploit and to analyze the quality of generalized rules, making it one more option to analyze such rules.

As future work, we plan to make the system $\mathcal{ENGAR}$ an extensible plug-in for the Weka Data Mining Software [3]. We also plan to validate the system carrying out other experiments using artificial and real data sets.

References

1. Jean-Marc Adamo. *Data Mining for Association Rules and Sequential Patterns.* Springer-Verlag, New York, NY, 2001.
2. Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.
3. Marcos Aurélio Domingues and Solange Oliveira Rezende. Post-processing of association rules using taxonomies. In *Proceedings of the 12th Portuguese Conference on Artificial Intelligence (EPIA 2005)*, pages 192–197, Covilha, Portugal, 2005.
4. Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition).* Addison-Wesley Longman Publishing Co., Inc., USA, 2005.
5. Li Yang. Pruning and visualizing generalized association rules in parallel coordinates. *IEEE Transactions on Knowledge and Data Engineering*, 17(1):60–70, 2005.

[3] http://www.cs.waikato.ac.nz/ml/weka.

An Extension of the Core Method for Continuous Values: Learning with Probabilities

Nuno C. Marques

CENTRIA/Departamento de Informática,
Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa
Quinta da Torre, 2829-516 Caparica, Portugal
nmm@di.fct.unl.pt

Abstract. This paper proposes an extension to the neuro-symbolic core method useful when observations are expressed by continuous values. Some theoretical results are presented regarding the learning process over these observations. An illustrative example is reported, demonstrating the problems of the original approach and justifying how this extension can overcome them. Results of the extended approach on irregular continuous values (simulating probabilistic data) show similar results to the original core method on clean symbolic data and point to the validity of the approach.

1 Introduction

The core method [1], allows the encoding of any logic program into a feed-forward one hidden layer neural network. This encoding expresses the immediate consequence operator T_p of a logic program. So, a direct recursive connection of the output layer to the input layer allows the computation of the result of any logic program by a neural-network with a feed-forward core. Neuro-symbolic (NeSy) integration for first order logic was recently achieved [2] with this method.

Knowledge encoding into neural networks was addressed in many works, including initial work by McCulloch and Pitts [3], knowledge based neural networks [4] and work with the core method (e.g. [1], [2], [5]). Today, we believe that more knowledge based approaches are one of the best ways to improve the applicability of standard back-propagation learning methods [6] to data mining and machine learning in general. One of the major advantages of these methods is their deep knowledge and possible formal description of what can be computed and learned from observations. From a machine learning perspective, the core method has already very good results reported (e.g. [7], [8]), namely by achieving faster convergence and better precision results than standard non-symbolic neural networks.

Unfortunately, results while applying the core method neural networks to natural language processing (e.g. [9]) were not so clear: improvements were detected and reported, but more general use of the method revealed difficult. Namely, it was too difficult to express the proper knowledge for this problem: backpropagation learns very fast how to encode easy rules in data. However too specific rules where easily forgotten during the training process.

Regarding neural-symbolic learning capabilities, the original core method proposal just focused the logic representation, and a stepwise activation function [1]. So several studies have been made regarding neural-symbolic learning capabilities, namely by replacing the stepwise function either by the standard *sigmoid* or *atan* activation functions and using gradient methods (such as error backpropagation, e.g. [8], [5]).

Recent results by [5], successfully propose a new way to encode symbolic incomplete knowledge in NeSy networks and improve them with experimental data. Also, until now, most results have always assumed binary input encodings: as it was shown in [7], the core method requires specific input encoding intervals, i.e. encoded rule output may be incorrect when inputs with intermediate values between $TRUE$ and $FALSE$ are used. In this paper we will elaborate on the results reported in [5], and conjoin them with the discussion of the problems that occur when specific probabilistic values outside core method validity intervals are used. Namely results in the learning speed and capabilities of backpropagation learning will be discussed. A specific focus will take into account the use of a probabilistic encoding, and an extension to the original core method will be proposed.

An illustrative problem will be addressed: learning AND with simulated probabilistic encoding.

2 The Core Method

Let us assume a sigmoid unit, where[1]:

$$net = w_0 + w_1 x_1 + \cdots + w_n x_n,$$

and

$$o = \sigma(net).$$

In this unit, o is the output, w_0 the bias and w_i the connection weights for each input x_i belonging to train data observation $\boldsymbol{X}$. Finally $\sigma(x)$ is the sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

Generically core method rules are embedded into the network according to the specification in [1], [8] or [5]. A one hidden layer feed-forward neural network with a predefined constant ω is used. Each hidden layer neuron can encode an AND rule. The unit connections are set to ω for positive literals and to $-\omega$

[1] This presentation of MLP neural-network learning for the core method will follow Tom Mitchell's notation and derivation of the backpropagation learning process in neural networks [10], that can be consulted for further details.

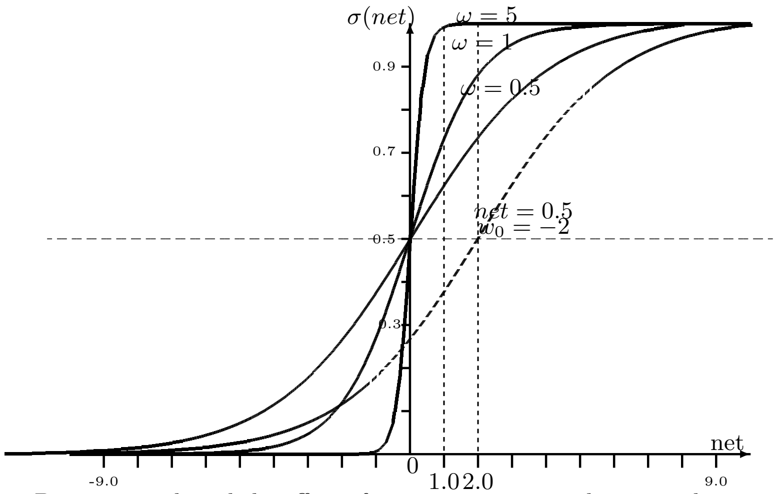


Fig. 1. Basic sigmoid, and the effect of some constants on the sigmoid activation function.

for negative literals. The bias (w_0) is set up such that this unit becomes active if and only if the input of the network coincides with the rules precondition (typically, $\omega \times (l - 0.5)$, where l is the number of literals of the rule). If no other constraints exist, the output layer can simply make an OR of hidden layer units by setting connecting weights (w_i) to 1.0 and bias to 0.0. Parameter ω is used as a multiplying factor over the predefined weights, to give additionally stability during neural network training. Bigger values of ω will make the sigmoid more similar to a step-wise activation function (figure 2), this way, high values for ω assure the encoded information is kept during learning.

Core method basic encoding can be exemplified with the following program:

$$A \wedge C \mapsto_{r_1} F$$
$$A \wedge \neg B \wedge D \mapsto_{r_2} F$$

The network for this program should have as input neurons i_A, i_B, i_C and i_D. If we take, e.g. $\omega = 5$, rule r_1 could be encoded by a hidden layer unit with $h_{r_1} = \sigma(-7.5{+}5{\times}i_A{+}5{\times}i_C)$ and rule r_2 by $h_{r_2} = \sigma(-12.5{+}5{\times}i_A{+}5{\times}i_B{+}5{\times}i_D)$. Final output F will simply be encoded by $F = \sigma(h_{r1}{+}h_{r2}{-}0)$. Usually, after encoding, but before backpropagation learning, all connections are slightly disturbed by adding some small random noise (e.g. $[-0.1, 0.1]$.).

2.1 Backpropagation Learning in the Core Method Networks

Backpropagation learns the weights w_i, by minimizing the squared error:

$$E[\boldsymbol{w}] \equiv \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2,$$

where D is the set of training examples. So, the perceptron training rule can be expressed as:

$$\Delta w_i = \eta(t - o)x_i,$$

$t = c(\boldsymbol{x})$ is the target value for observation x and η is the learning rate (usually 0.1). The above training rule is convergent if training data is linearly separable and η sufficiently small [10]. So, taking into account $\nabla E[\boldsymbol{w}]$ gradient, the training rule:

$$\Delta \boldsymbol{w} = -\eta \nabla E[\boldsymbol{w}] = -\eta \frac{\partial E}{\partial w_i},$$

can be used to derive the gradient rules to train one sigmoid unit based on:

$$\frac{\partial E}{\partial w_i} = -\sum_d (t_d - o_d) \frac{\partial o_d}{\partial net_d} x_{i,d}. \tag{1}$$

Regarding learning in the core method, we should notice that by equation 1, learning depends on $\frac{\partial o_d}{\partial net_d}$. But this introduces a problem when learning with bigger values of ω. If we look at figure 2 we see that ω is indeed influencing learning: bigger values of ω make the sigmoid approximate a stepwise function and make $\frac{\partial o_d}{\partial net_d}$ go near zero, when values of net are bigger than 1.0 (this effect was observed experimentally in [5]). This is also relevant for the core method since the usual initialization of parameters is biasing the network and making harder for equation 1 to update weights (i.e. update will be very small for large values of net). This may be a good behavior, if we are certain on the rules, but not advisable if rules need revision/are wrong (e.g. [5]). Indeed, for the following discussion we should note that there is a learning zone that depends on ω and on the values of net: in practice, only small enough values of net and ω make learning possible (or, alternatively, fast enough).

3 Decisions based on distinct continuous values

Traditional error back-propagation in feed-forward neural networks deals with continuous values. However, traditional logic is based on propositions and on the truth value of those propositions. So, the usual input for the Core Method uses a 0.0/1.0 encoding for Boolean values[2]. We think that, until now, this has hidden a potential problem when generalizing the Core Method to continuous inputs on sigmoid neural networks. Let us start by showing that:

Proposition 1 *Any logic program with Boolean inputs can be encoded in a sigmoid neural network.*

[2] An alternative $-1.0/1.0$ encoding is also frequently used with the $\sigma(x) = atan(x)$ function, namely for easier encoding of negated literals. Without loss of generality, and for a more usual presentation, we have used the the equivalent sigmoid $\sigma(x)$ function.

Previous results (e.g. [7], [8]) have already shown a similar result. However, this new formulation will be useful for our discussion because it explicitly takes into account the value of ω.

Proof 1 *Previous results showed how to encode any propositional logic program inside a neural network with step-wise activation (e.g. proposition 4 in [1]). We consider a sigmoid neuron active when the sigmoid function is bigger than some appropriate value α and inactive when the sigmoid is smaller than some value $1-\alpha$ (where $\alpha \in [0.5; 1[$). Then, for any fixed α, we can make ω as big as needed, so the sigmoid function will approach the step-wise activation function up to an value ε. This way, we can make ε arbitrarily small, so that for any given problem the neurons activate in a way that follows proposition 4 in [1].* •

By conjoining this proof with preceding discussion, we should notice there is an engineering decision that must be taken when deciding the values to use for ω. As it was already pointed, backpropagation algorithm can not learn well enough for big values of ω. On the other side, if ω is too low and the encoded information is not evident from experimental data, the core method encoded information will be lost in noise and during backpropagation updates[3]. So we must choose suitable ω values to encode our knowledge inside the neural network, and, simultaneously, allow learning to be done.

Proposition 1 can now be extended for the cases when the input is not restricted to Boolean values.

Proposition 2 *Any logic program with conditions over a vector of continuous values $\boldsymbol{X}$ can be encoded in a sigmoid neural network.*

Proof 2 *Any condition over a continuous value could be given a logical value by constraints: $X_i > \mathcal{T}_{X_i}$.*

So a sigmoid neuron having an input X, and a predefined ω can implement this constraint, namely, if we consider a neuron with bias $\omega \times \mathcal{T}_{X_i}$:

$$1 - alpha < \frac{1}{1 + e^{-\omega \times (X_i - \mathcal{T}_{X_i})}} < alpha.$$

So (e.g. figure 2),

$$X_i - \mathcal{T}_{X_i} > 0,$$

will be true iff $X_i > \mathcal{T}_{X_i}$.

Let us consider a first hidden layer of such threshold units for all $X_i \in \boldsymbol{X}$, converting $\boldsymbol{X}$ to each logical value needed by program a P. Then, by proposition

[3] During experiments we have observed that -due to the distributed nature of backpropagation learning-, in some configurations, the already given information can even be harmful. This happens when backpropagation tries to re-encode (or re-learn) that information in distinct ways.

1, we can add a second hidden layer encoding this logic program P using the logic output of this first hidden layer. •

With this result, we are no longer reduced to the single threshold bias of the rule neuron and we can have a set of threshold values, distinct for each input. Additional computational power of the neurons in this first hidden unit can also be handy for more modular encodings of logical symbols (e.g. for encoding lists). Moreover, we can increase ω value for the connection to this neurons, in order to enforce some of the intended mappings (and to make it more stable during learning) or to reduce them and so concentrate learning on those areas of the network.

Probabilistic Encoding Proposition 2 is important for problems were we don't have a direct way to describe a given object in terms of its explicit features. Usually, in these cases knowledge is better expressed using probability theory. E.g. in [9], the probability that some word w being tagged as T was calculated based on:

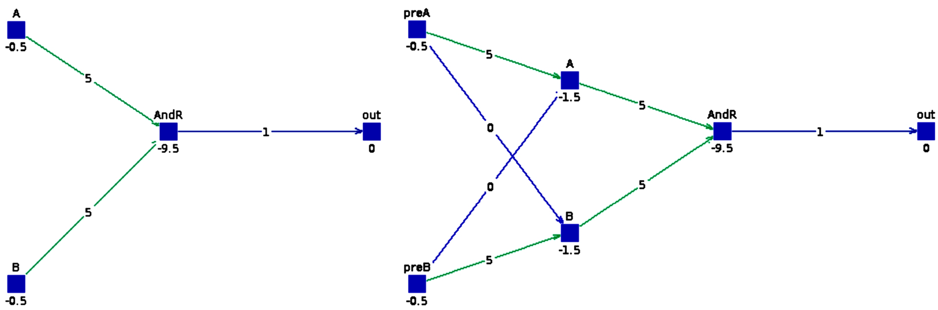
$$p_w(T|event) = \frac{freq(event, T)}{freq(event)}.$$

In this equation, $freq(event, T_i)$ denoted the number of times $event$ had mark T and $freq(event)$ denoted the total number of observations of $event$. In this encoding a probability value of, e.g. 0.1, would represent ten observations per 100 occurrences of that event (e.g. a given word). This is a meaningful value and could be modeled by some rule related to the presence of T. However there is a language problem when we talk about probabilistic events in a logical format: Assume P_b is the probability for a feature b and that P_c is the probability for an independent feature c. If we know that a is the logic consequence of both b and c, we can write:

$$a \leftarrow b(P_b), c(P_c). \tag{2}$$

According to this equation, if both b and c are probable enough to hold, i.e. $P(x) \geq \mathcal{T}_x$ then we should expect a to also hold. However, due to statistical independence, $P_a = P_b \times P_c$, i.e. the minimal value for P_a to hold is ${P_a}^{min} = \mathcal{T}_b \times \mathcal{T}_c$, a value that should be in the non-accepting region of the usual $T \geq 0.5$ core method perceptron (e.g. if $\mathcal{T}_a = \mathcal{T}_b = 0.2$, but $\mathcal{T}_c = 0.6$, then the independent event has $P_a = 0.12 < \mathcal{T}_a$). In this case, it might not be advisable to change the threshold of all the inputs by using the bias of the hidden neuron (i.e. either there is an error in considering a and b $FALSE$ or an error by considering c $TRUE$). So, there could be a problem when representing this probability knowledge for a neural network using the original core method. I.e., we may be interested in the logical consequence of two events considered true, and not on the probability of its joint occurrence. In this cases we should use the encoding proposed as a result of proposition 2, so the decision surface of the core method perceptron can be initialized according to the intended logical consequence function of individual threshold values $\mathcal{T}_b$ and $\mathcal{T}_c$.

4 An Illustrative Example while Learning an AND



1HL– Original Core Method *2HL– Extended Core Method*

Fig. 2. Initial core methods neural networks, before training and initial weight disturbance.

For better illustrating the described problem, a simple $C \leftarrow A \wedge B$ rule was encoded in a feed-forward sigmoid based neural network with two inputs (A and B) and a single output (C). According with the original core method, a neural network with a single hidden unit was defined. We set $\omega = 5.0$, so this unit had the two initial weights (one for each possible value of input) also set to 5.0. However the initial bias on all these networks was set to -9.5 (i.e. a slightly higher bias that assumes $\mathcal{T} \geq 0.9$). This hidden unit was then connected with the single output layer unit (C) with initial weight set to 1.0. The neural network is represented in figure 2 − 1HL.

For this example two datasets were built simulating different threshold values over the AND truth table: one had input $TRUE$ values encoded as 0.4 and in the other $TRUE$ was encoded as 0.9. Output $TRUE$ values were always encoded as 1.0 and all $FALSE$ values were encoded as 0.0.

A second neural network (represented in figure *2HL*) was also built according with proposition 2: a first hidden layer was added between input layer and the original hidden layer. As described, this layer implements the threshold detection for the 0.4 encoding. This layer has two hidden neurons (one representing each input neuron), connected with initial weight of 5.0 to the neuron that is being mapped and with 0.0 to the other neuron. Initial bias was set to -1.5 (i.e. $\mathcal{T}_X \geq .3$, with $\omega = 5.0$). Finally weights in *1HL* network were randomized (between -0.1 and 0.1), to provide a control case (*RND* network).

The three neural networks were trained with standard backpropagation method ($\eta = 0.2$). Learning stopped when error variation was smaller than 0.01. An initial disturbance of 0.1 was added to all networks. During training a weight disturbance of 0.05 was done every 100 iterations. The SSE values for all networks during the train process are represented in figure 3. Several other experiments were also preformed with similar results. E.g., we should point that for the more standard bias of 7.5 (i.e. $\mathcal{T} \geq 0.5$) a similar behavior was observed when $TRUE$ was encoded as 0.25. Namely the initial core method network revealed itself much more sensible to noise than the network derived from proposition 2.

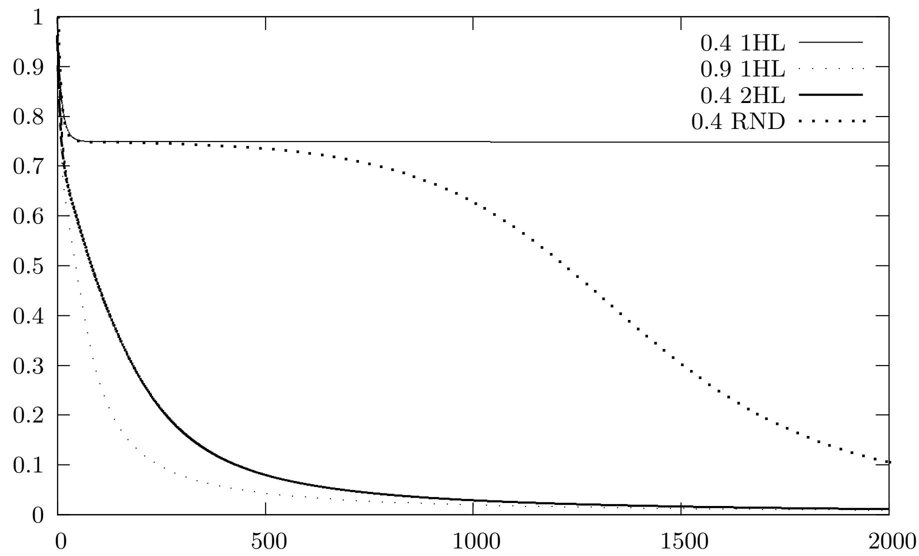


Fig. 3. Train error for learning an AND with four distinct encodings.

5 Discussion and Conclusions

The standard core method neural network on the 0.4 dataset (*0.4 1HL*) didn't converge in more than 100000 iterations[4]. We should also mention the importance of setting ω to appropriate values in *0.4 2HL* network. Indeed, the first hidden layer value for ω was first set to 1.0. However then, although the network was still faster to learn than the control (*RND*) network, a much bigger sensibility to the initial *bias* in the first hidden layer was detected, with convergence only starting around iteration 500 and 0.1 SSE achieved around iteration 1000. When $\omega = 5.0$ the behavior was more stable, and even if wrong initial bias

[4] This was due to the noise added every 100 iterations. But even if jog weighting is eliminated, *0.4 1HL* network only starts convergence around iteration 5000.

were given (e.g. 0), the backpropagation quickly found the correct bias values. Moreover, if we set ω to 10.0 the behavior of *0.4 2HL* is indistinguishable of *0.9 1HL*.

This small example shows how sensible the initial core method proposal may be, when appropriate distinct continuous input conditions exist. Indeed we should notice that $1HL$ network is much worse than the control (RND) network on this peculiar case. Even minor weight changes can condition the proper convergence of this network. From what we have observed the correct initial knowledge encoded into the network was unused and made learning almost impossible for backpropagation algorithm. From ongoing experiments we have noticed this effect to become even worst in real scenarios.

This result is clearly demonstrating the need of additional computational structures for helping the network to find the truth values when inputs are sent as continuous irregular values. It should be stressed that this was not a theoretical example. Indeed there are many cases (as, e.g., the one mentioned in [9]) where the additional information encoded in probability values is needed for the system to perform properly. The main problem was that, although this information should be provided to the neural network (so that it could learn other relations with on free neurons), it was then difficult to encode the symbolic knowledge into the network.

The solution presented by proposition 2 is quite valuable in these terms, since it allows the network and training algorithm to protect encoded knowledge from random fluctuations in data. Also the setting of ω values revealed itself as a quite powerful tool to help in optimizing and even directing learning effort into the appropriate areas of the network. Indeed, the author is now convinced that we are now approaching the point were Neuro-Symbolic integration is starting to prove its capabilities for general machine learning problems. The general perspective is that neural network programming and tuning for known knowledge is now increasingly possible while still keeping the network open to learn other information contained in training data.

6 Acknowledgments

This work would not have been possible without initial joint work done with Sebastian Bader and Steffen Hölldobler, namely during author's stay as an invited researcher in the Technical University of Dresden. I thank both of them, not only their excellent help and support, but also the very good working conditions and environment they made available in TU-Dresden.

References

1. Hölldobler, S., Kalinke, Y.: Towards a massively parallel computational model for logic programming. In: Proceedings ECAI94 Workshop on Combining Symbolic and Connectionist Processing, ECCAI (1994) 68–77

2. Bader, S., Hitzler, P., Hölldobler, S.: Connectionist model generation: A first-order approach. Neurocomputing **51** (August 2008) 2420–2432
3. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics **5** (1943) 115–133
4. Towell, G.G., Shavlik, J.W.: Knowledge-based artificial neural networks. Artificial Intelligence **70**(1–2) (1994) 119–165
5. Bader, S., Hölldobler, S., Marques, N.C.: Guiding backprop by inserting rules. In d'Avila Garcez, A.S., Hitzler, P., eds.: Proceedings of the 4th International Workshop on Neural-Symbolic Learning and Reasoning, Patras, Greece, July 2008. CEUR Workshop Proceedings, Vol. 366, 2008. ISSN 1613-0073. Volume 366. CEUR Workshop Proceedings (2008) ISSN: 1613-0073.
6. McClelland, J., Rumelhart, D.: Parallel Distributed Processing. MIT Press (1986)
7. d'Avila Garcez, A.S., Zaverucha, G.: The connectionist inductive learning and logic programming system. Applied Intelligence, Special Issue on Neural networks and Structured Knowledge **11**(1) (1999) 59–77
8. d'Avila Garcez, A.S., Broda, K.B., Gabbay, D.M.: Neural-Symbolic Learning Systems — Foundations and Applications. Perspectives in Neural Computing. Springer, Berlin (2002)
9. Marques, N.C., Bader, S., Rocio, V., Hölldobler, S.: Neuro-symbolic word tagging. In José Neves, Manuel Filipe Santos, J.M., ed.: New Trends in Artificial Intelligence, APPIA - Associação Portuguesa para a Inteligência Artificial (12 2007) 779–790
10. Mitchell, T.M.: Machine Learning. McGraw-Hill (March 1997)

Histogram Based Payload Processing for Unsupervised Anomaly Detection Systems in Network Intrusion

Iñigo Perona, Iñaki Albisua, Olatz Arbelaitz, Ibai Gurrutxaga, José I. Martín, Javier Muguerza, and Jesús M. Pérez

Dept. of Computer architecture and Technology
University of the Basque Country
M. Lardizabal, 1, 20018 Donostia, Spain
{inigo.perona,inaki.albisua,olatz.arbelaitz,i.gurrutxaga,j.martin,j.muguerza,txus.perez}@ehu.es

Abstract. The popularity of computer networks broadens the scope for network attackers and increases the damage these attacks can cause. In this context, any complete security package includes a network Intrusion Detection System (nIDS). This work focuses on nIDSs which work by scanning the network traffic. We present a service-independent payload processing approach, based on histogram representation, to increase detection rates in non-flood attacks. We implemented three different options combining histogram representation and fixed width clustering algorithm for anomaly detection, and compared them to a sysstem based on packets' header information, another system based on ad hoc payload processing and our previous general payload processing proposal. The new options outperformed the previous ones; they detected efficiently most of the attack types. Moreover, the proper integration of the knowledge of the different techniques, payload-based and packet header-based, always improved the results. This work leads us to conclude that payload analysis can be used in a general manner, with no service- or port-specific modelling, to detect attacks in network traffic.

Key words: Intrusion detection systems, unsupervised anomaly detection, payload, histogram, AUC

1 Introduction

There has been a huge increase in the use of computer networks. This fact broadens the scope for network attackers and increases the damage these attacks can cause. Network attacks affect the security of the information stored on computers connected to the network and its stability. Therefore, it is very important to build systems that are able to detect attacks before they cause damage. Any complete security package includes a network Intrusion Detection System (nIDS).

The detection of network attacks can be done by human analysis or automatically. The detection by human analysis requires memorisation, looking up description libraries or searching sample collections and it is not effective; it is too time consuming and subjective. As a consequence, in order to successfully confront the problem, the security systems require automated and robust nIDSs. In this sense, a very popular option is the use of data mining techniques, mainly trained on labelled data, to detect attacks. We can find in bibliography three main approaches for nIDSs. The first two are misuse detection approach [11] and anomaly detection approach [26]. Due to the problems the previous approaches have, a third one appeared: unsupervised anomaly detection [18]. Usually the best option is a combination of some intrusion detection systems. For example, a flood detecting firewall could first filter most flood attacks; a signature-based IDS could then be used to remove the known attacks and unsupervised anomaly detection could finally focus on detecting the unknown attacks.

The characteristics of the attacks change depending on the kind of attack and as a consequence, the suitability of a tool to detect them will also change. Most of the flood attacks can be successfully detected by scanning the TCP/IP headers of network packets but this information is not enough to detect most of the non-flood attacks. It is nearly impossible for systems to use traffic models to detect User to Root (U2R) or Remote to Local (R2L) attacks because the intruder only has to send very few packets (often, a single one is enough). R2L and U2R attacks can lead to catastrophic consequences because they are actually the only ones that allow the intruder to obtain complete control of the attacked system. In this context some authors propose the use of another source of information: the transferred information or payload. The features of the payload vary depending on the kind of network connection and service. As a consequence, most payload based IDSs we can find in bibliography are service-specific [10][25][12]. These service-specific methods are very context dependent. That is to say, as they are moved to machines offering different services or as new services appear, the system will need to be rebuilt. In this context it would be important to be able to build a system that is able to work in any environment independently of the kind of services or machines.

The aim of our work is to contribute to build efficient and context independent nIDSs. In a previous study [17] we proved that information obtained from general payload processing is useful to detect non flood network attacks in an unsupervised anomaly detection context. But we wanted to go further and the purpose of this work was to answer the following questions. Can general payload processing be more efficient than some specific payload processing for intrusion detection in an unsupervised anomaly detection context? and can this be done in a computationally efficient way?

To answer to our research questions we processed the payloads as byte histograms and used histogram comparison methods [20] to compare different payloads. We used clustering as anomaly detection technique and compared the results achieved with our system, general payload processing, to the ones achieved with specific payload processing. Based on previous experience, we combined results obtained from payload analysis with the results obtained from packet header analysis. The results showed that representing the payload as byte histograms and using histogram comparison methods as distances for clustering builds efficient anomaly detection systems for network attack detection. Furthermore, they can be used to complement techniques based on packet header analysis, since the combinations tried always improved the results.

The paper proceeds to describe in Section 2 the main automatic approaches used to build IDSs. In Section 3 we describe the approximation we used in this work for outlier detection. Section 4 is devoted to describing the histogram signature based approach we used to process payload without any context knowledge. In Section 5 we present the schema of the proposed system. The paper continues in Section 6 where we describe the data used in the experimentation and we present experimental results in Section 7. Before the conclusions Section 8 is devoted to a short discussion. Finally, we summarise in Section 9 the conclusions and further work.

2 Main approaches for IDS

We found in bibliography three main data mining based approaches for intrusion detection: misuse detection, anomaly detection and unsupervised anomaly detection. In the misuse detection approach, which is used in systems such as MADAM/ID [11] the authors use machine learning techniques on labelled data: the classifier learns from a set of labelled connections, where there is normal traffic and attacks, and in subsequent uses it recognises known attacks. These methods have two main problems. On the one hand, it is very difficult to obtain completely labelled network traffic and, on the other hand, they can not solve the zero-day problem and as a consequence, the new attacks will always succeed in damaging the system. They need to be revised each time a new kind of intrusion appears and this happens every day. Nevertheless, the primary objective will be to detect the first occurrence of intrusions and prevent it from damaging any victim.

Warrender, Forrest and Pearlmutter wrote a survey [26] of IDSs based on anomaly detection approaches. This method profiles normal network traffic behaviour and successfully detects attacks when the observed traffic deviates from the modelled behaviour. In anomaly detection approach, classifiers learn how normal traffic behaves and any anomalous connection is considered to be an attack. As a consequence, if the engineers do not model all the kinds of normal traffic, the systems will have high false positive rates. Moreover, in real environments it is not usual to have purely normal data and these approaches need it in order to model just normal traffic. If any attack is left in the hypothetical purely normal data, this attack will be learnt as normal traffic and the IDS will never produce an alert related to it.

Due to the problems the previous approaches have, many researchers are working on a third one: unsupervised anomaly detection [18]. It does not need purely normal data and it uses unlabelled data, which is easy to obtain. This option works under the assumption that the volume of normal traffic is much greater than the traffic containing attacks, and, furthermore, the intrusions' behaviour is different from normal data's' behaviour. Under these assumptions the intrusion detection problem can be confronted using outlier detection techniques. This approach can be used as a stand-alone system, or, even more effectively, it can be combined with a misuse detection or anomaly detection process.

Unsupervised anomaly detection methods are inadequate for detecting flood attacks because these kinds of attacks usually need to send a large number of packets in a short time and as a consequence they will naturally form large groups that will not be detected as anomalies. Nevertheless, flood attacks are easy to detect and some authors achieve high detection rates by simpler systems that scan network traffic or analyse headers [15]. Although it is long since the fist anomaly detection approaches appeared, it is still a successful approach being used in many systems. An example of the use of this methodology is the number of papers mentioning it in the last conference in Recent Advances in Intrusion Detection RAID 2008. Ashfaq et al [1] for example presented a comparative evaluation of 8 lately developed anomaly detectors under portscan attacks from the accuracy, scalability, complexity and detection delay point of view. The authors built two independently collected datasets for the evaluation, both of them including packet header information since all the evaluated systems are based on this information. On the other hand, Dagorn presents in [2] an anomaly-based intrusion detection system for web applications and Rehák eta al [19] present a way to improve error rate in anomaly detection by collective trust modelling.

3 Detecting outliers

As we mentioned in previous section, unsupervised anomaly detection strategies can be formulated as outlier detection problems [6]; they usually build probabilistic models of the data that will help them to decide whether or not the connections are attacks. We concretely used clustering as a tool for anomaly detection. We first performed the clustering over the points in the feature space, the connections, and assigned a score to each cluster based on its size. Then we scored the examples in each cluster based on this score, and we used this score to determine the degree of anomality of the example. We labelled the points with lower scores as anomalous. Although many clustering algorithms could be used, based on the experience of other authors [4][13], we selected the fixed-width clustering algorithm [4], also known as the leader algorithm [21]. The fixed-width scales linearly to the number of examples of the database and the number of clusters. This algorithm does not accurately fit to databases with clusters of different sizes; it over partitions the largest clusters. Nevertheless, in the unsupervised anomaly detection context we are interested just in the small clusters, so this drawback of the algorithm is not a real problem.

4 Payload processing

TCP/IP headers of the packets detected on the network traffic can be easily processed because the format of these headers is well-known. On the contrary, payload processing is a difficult task because its format in a packet depends on the application and used protocol. Moreover, many protocols have fields where any kind of data can be stored. Some authors solved this problem by performing the data processing in a specific way for each service [10][25][11]. This option has many drawbacks: it works for a reduced set of connections, the used protocol is not always known and new services are not automatically treated. The selected payload processing method needs to be helpful to detect attacks, but, it also requires having some other characteristics such as:

1. Not requiring human intervention. That is, to be automatic.
2. To be service-independent, and, as a consequence, usable in different environments and adaptable to changing situations. That is, to be general.
3. To be computationally efficient.

It is not easy to build a system with all the required skills; it seems, on the one hand, that more complex or computationally expensive systems would better model the payload. On the contrary, payload data can generally be seen as a sequence of bytes, so in a previous work [17] we already processed it regardless of the kind of service or port, based on byte frequencies and sequence comparison techniques. In this work we processed the payload in a very simple and efficient way: as byte histograms or 1-grams. We represented the ASCII characters (0-255 bins) in the x-axis and their frequencies, normalised with the payloads' length, in the y-axis.

Histogram comparison methods

Histogram comparison is an important field in pattern classification and data clustering. As a consequence there are many approaches to calculate distances or similarities between histograms. Some approaches propose distances for ordered histograms. Strelkov presented in [22] for example a distance based on the closeness of positions and shapes of peaks in the compared histograms. This peak matching measure mainly moves one histogram relatively to another in its inner compute. This kind of distance requires ordered histograms, i.e., the neighbour bins on a histogram need to contain related information. In our case, the histograms are nominal. When processing payload the bins correspond to ASCII characters which are independent from each other. We based our research mainly in the work of Serratosa and Sanfeliu [20]. They propose the use of signatures, a loss-less representation of histograms, to calculate distances between histograms in an efficient way. The signature is a vector that contains the non-zero bins of the corresponding histogram and x-axis indexes for the saved bins. Thus, the signature does not lose information. For each payload (P_l) corresponding to each of the network connetcions we built an histogram $H(P_l) = [H_1(P_l), ..., H_T(P_l)]$ where T is the amount of different discrete values, 256 in our case, and, $H_i(P_l), 1 <= i <= T$ are bins (frequency of the byte i in the payload P_l). Let $S = [S_1(P_l), ..., S_z(P_l)]$ be the signature of the set P_l. Each $S_k(P_l), 1 <= k <= z <= T$, is composed of a pair of terms, $S_k(P_l) = \{w_k, m_k\}$. The first term,w_k, shows the relation between the signature $S(P_l)$ and the histogram $H(P_l)$. Thus, if $w_k = i$ then the second term,

$m_k = H_i(P_l)$ where $m_k > 0$. To compute the distance between two signatures they need to have the same length. The authors propose the use of extended signatures: a signature with the minimum number of empty bins added so that, given a pair of signatures to be compared, the number of bins is the same in both of them. Moreover, each bin in both signatures represents the same bin in the histograms. Once we got the extended signatures we computed the distance between two histograms using Euclidean distance, or using the nominal, Manhattan, or the ordinal, Landmover, distances proposed in [20].

1. The Manhattan distance between two histograms is the number of elements that do not overlap.
$$D_{nom}(S(A'), S(B')) = \sum_{i=1}^{z'} \left| m_i^{A'} - m_i^{B'} \right|$$
2. The Landmover distance between histograms is the minimum number of unit-bin movements needed to transform one histogram to the other. This distance takes into account the sky-line of histograms and can be seen as a way of comparing smoothed histograms [23][8].
$$D_{ord}(S(A'), S(B')) = \sum_{i=1}^{z'-1} (w_{i+1}^{A'} - w_i^{A'}) \left| \sum_{j=1}^{i} (m_j^{A'} - m_j^{B'}) \right|$$

5 Schema of Intrusion detection process

For clarity, in this section we summarise the steps of the payload based intrusion detection tool we propose. Figure 5 shows a schema of the process. Once network data is collected we divide it in two main parts: the connections headers' information on the one hand, and the transferred information or payload on the other one. We processed the TCP/IP headers information to obtain a tabular representation with intrinsic variables and traffic variables and obtained byte histograms from the payload part. In next step we applied fixed width clustering algorithm to both parts: combined with Euclidean distance for the headers information and combined with the histogram based distances defined in Section 4 for payload. The output of each clustering process was a different partition. Finally, we combined the scores and obtained the final value for each connection. These scores will be the ones used to determine the degree of anomality of the connection.

6 Data generation

It is difficult to obtain labelled data or a database with purely normal data for network traffic. This makes difficult the evaluation and comparison of results of intrusion detection systems; unsupervised anomaly detection techniques do not require labelled data to work, but they need it so that the system can be evaluated. We wanted to generate comparable results and we decided to use some standard data such as Kddcup99 from the UCI repository [9]. Kddcup99 was built from the DARPA98 dataset [3], which was generated by the Information System Technology Group (IST) of the Lincoln Laboratory of the MIT

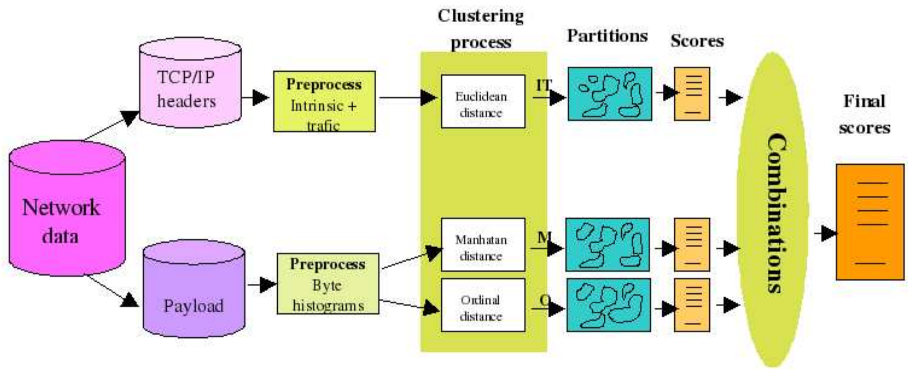


Fig. 1. Schema of the proposed Intrusion Detection System

with the collaboration of DARPA and ARFL. They built a network to simulate a real situation of network traffic containing normal traffic and attacks. They used Tcpdump [7] to sniff the network and stored all the packets belonging to network traffic in a tcpdump file. Lee generated the UCI format Kddcup99 database identifying connections and aggregating information belonging to them. Each connection has three kinds of features: intrinsic variables (those obtained by examining the packets' TCP/IP structure such as protocol, length, urgent bit); traffic variables which take into account header information of preceding connections contained in a window of some specific size; and, finally, content variables obtained by examining the payload of some particular services, such as number of failed logins, number of file creations, etc.. KddCup99 database processes a huge amount of information from DARPA98 dataset and stores it in a format suitable for most machine learning algorithms, but, it does not store the original payload information. The only payload based information it keeps is in the content variables. This is obviously not a general solution.

We reprocessed the DARPA98 database, based on Lee 1999 and using Bro [16], to add information from the original DARPA98 to Kddcup99 database. In this new database, each connection will have the intrinsic and traffic variables of Kddcup99 added to all the payload data corresponding to it. The aim of this work was to replace the information the content variables provide by automatic payload processing.

Due to the huge size of the original Kddcup99 database (about 5,000,000 connections), most authors performed their experiments using a sample of the original dataset. This sample contains about the 10% of the connections. Similarly, we extracted a stratified sample of about 10% of the size of the original one. Since our goal is to find the non-flood attacks, and the DARPA98 is overloaded with flood attacks, we filtered all the flood attacks in the dataset. Thus, we worked with a database of 178,810 examples, where 3,937 examples belong to intrusions of 27 different kinds. We show in the first two columns in Table 1 the information about the kind of attacks and their frequency.

7 Experimental results

In a previous work [17] we proved that the payload on its own could be used to detect intrusions. In this work we evaluated the performance of the histogram-based payload representation and the three distances described in Section 4. As first approximation we used just the payload information to build classifiers based on the methodology described in Section 3. We built models of the system using fixed-width algorithm for 7 different options:

1. The specific content variables defined in Kddcup99 (C).
2. Three options experimented in our previous work [17].
   - Payload treated as sequence of bytes and compared with NCD distance [14](NCD).
   - Payload represented with the 30 most frequent bytes. We treated each byte as independent variable and used Hamming distance to compare payloads. We called this option MFN (Most Frequent Nominal).
   - Payload represented with the 30 most frequent bytes and considering it a sequence where the position of each character influences the distance. We called this option MFO (Most Frequent Ordinal).
3. Three options for comparing histograms described in Section 4
   - Euclidean distance (HE).
   - Manhattan distance (HM).
   - Ordinal distance (HO).

We experimented with the whole database, that is, normal traffic data plus data from 27 different kinds of attacks. We evaluated the results based on the ROC curves and the Areas Under ROC Curves, or AUC values, obtained [5]. To compute the ROC of just a single attack type, we ignored the examples belonging to other attack types. For each option we present detection rates of each type of attack separately, minimum AUC achieved with each model and weighted average —taking into account the number of attacks of each type— of the achieved AUC. We also generated and included results obtained using just intrinsic and traffic variables (IT) as baseline.

The rows in Table 1 belong to differend attack types whereas the columns belong to different systems. The second column in Table 1 shows the number of examples of each type of attack we find in the database, the third one shows AUC values achieved using packet header information (IT) and, next seven columns show the results achieved with each methodology based only in payload information.

In general terms, the first conclusion that can be drawn from this processing is that although no context knowledge is used and simple processing is performed, the six options for modelling payload in a general way (NCD, MFN, MFN, HE, HM, HO) are able to differentiate between normal traffic and intrusions and besides, the three options we proposed in this work, HM, HE and HO, achieved the highest average AUC values, they did it even better than the model built with data obtained using context specific knowledge for processing payload (C) or the packets' header information (IT). Besides, the row with minimum AUC value for each option shows that mainly two of the options are more interesting: HM and HO. They achieved AUC values of 0.69 or bigger for all the kinds of attacks. The rest of the options have minimum values smaller than 0.5 which means that for some kind of attack they achieved worse results than a random classifier would.

Different techniques showed the ability to detect different kinds of attacks, and based on previous experience, we knew that it is possible to integrate the knowledge of the payload based techniques and the packet header based technique to improve the original results. Thus, we combined by averaging scores [17] the results obtained with Intrinsic and Traffic variables (IT), with the results obtained with Content variables on the one hand (IT+C), and, with results obtained with HM and HO (IT+HM+HO) on the other hand; the best two options of histogram processing. The combinations contributed to increase the overall AUC values in both cases. The combination of IT, HM and HO is the one that achieved the best results with average AUC of 0.955 and minimum AUC value of 0.89.

8 Discussion

The experimentation presented in previous section proves that a general payload processing methodology, histogram representation, is more efficient than the specific payload processing done in Kddcup99 [12] for intrusion detection in an unsupervised anomaly detection context. The proposed payload processsing options were able to detect more attacks in both cases: when used on their own and when combined with the information of intrinsic and traffic variables. And besides, the proposed options are not computationally expensive since they require simple mathematical operations.

The payload of different network connections can be very different. The transferred information usually depends on the kind of service, and, as a consequence there are few works where the payload is used to model network traffic and detect the possible intruders. When payload is used with this aim service-specific approaches are developed. For example Krügel Toth and Kirda [10] presented a work that focuses on R2L attacks and uses service-specific knowledge to increase the detection rate of intrusions. They implemented a prototype that can process HTTP and DNS traffic although they only presented results for DNS. Wang, and Stolfo [25] based their work on profile byte frequency distribution and they computed the standard deviation of the application-level payload flowing to a single host and port during a training phase. They used the Mahalanobis distance during the detection phase and if the distance exceeded a certain threshold the system generated an alarm. This model is also host- and port-specific and conditioned by the payload length. In a different context, Wazumi et al. in [24] also processed the payload as byte histograms for early worm detection. But they did a different work since, instead of concentrating in the reduction of false

Table 1. Attack detection rates for different models

attacks		IT	C	NCD	MFN	MFO	HM	HE	HO	IT+C	IT+HM+HO
anomaly	9	0.76	1	0.88	0.35	0.74	0.98	0.64	1	1	0.96
dict	879	0.76	0.99	0.82	0.64	0.83	0.93	0.92	1	0.95	0.95
dict_simple	1	0.65	1	0.81	0.69	0.83	0.98	0.99	1	1	0.96
eject	11	0.76	0.98	0.82	0.8	0.8	0.99	0.35	0.98	0.98	0.96
eject-fail	1	0.99	0.8	0.48	0.99	0.58	0.95	0.95	0.97	1	0.96
ffb	10	0.8	0.85	0.88	0.72	0.93	0.95	0.65	0.96	0.93	0.95
ffb_clear	1	0.65	1	0.81	0.71	0.67	0.99	0.97	0.96	1	0.95
format	6	0.79	0.75	0.93	0.81	0.95	0.93	0.82	0.96	0.89	0.93
format_clear	1	0.52	1	0.81	0.88	0.83	0.99	0.21	0.95	1	0.92
format-fail	1	0.98	1	0.81	0.8	0.67	0.99	0.75	0.95	1	1
ftp-write	8	0.88	0.73	0.88	0.76	0.56	0.8	0.87	0.9	0.87	0.89
guest	50	0.77	1	0.85	0.81	0.83	0.93	0.92	0.89	0.94	0.96
imap	7	0.9	0.8	0.97	0.97	0.68	0.97	0.86	0.89	0.92	0.94
land	35	0.92	0.8	0.48	0.99	0.58	0.95	0.95	0.88	0.94	0.94
load_clear	1	0.65	1	0.81	0.12	0.14	0.93	0.99	0.87	1	0.92
loadmodule	8	0.7	0.84	0.71	0.69	0.68	0.97	0.77	0.87	0.87	0.89
multihop	9	0.72	0.74	0.78	0.63	0.71	0.93	0.94	0.84	0.83	0.98
perl_clear	1	0.95	1	0.81	0.52	0.87	0.99	0.99	0.81	1	0.94
perlmagic	4	0.66	1	0.83	0.86	0.86	0.99	0.99	0.77	1	0.96
phf	5	0.9	0.5	0.71	0.99	0.72	0.98	0.98	0.77	0.88	0.93
rootkit	29	0.88	0.81	0.77	0.86	0.77	0.94	0.96	0.76	0.87	0.96
spy	2	0.71	0.8	0.81	0.66	0.52	0.99	1	0.76	0.86	0.98
syslog	4	0.82	0.8	0.48	0.97	0.58	1	0.9	0.75	0.85	0.94
teardrop	1085	0.82	0.65	0.48	0.76	0.58	0.89	0.48	0.69	0.85	0.91
warez	1	0.96	0.31	1	0.12	0.85	0.93	0.97	0.69	0.98	0.96
warezclient	1749	0.81	0.68	0.86	0.86	0.86	0.94	0.92	0.69	0.83	0.95
warezmaster	19	0.94	0.75	0.87	0.96	0.88	0.82	0.87	0.69	0.96	0.95
min		0.52	0.31	0.48	0.12	0.14	0.8	0.21	0.69	0.83	0.89
Average		0.845	0.80	0.746	0.631	0.765	0.929	0.918	0.854	0.91	0.955

positives, and as a consequence the AUC in a network, they proposed a payload processing methodology to detect worms in different networks; they only experimented with a worm, Beagle_AV. Another example of payload processing can be found in the content variables of Kddcup99 [12]. In this case, the author obtained some information from the payload based on the experts' experience. This kind of processing is very context dependent and it can only be done for some well known services and protocols. The processing is totally static; it has no learning capability at all. In order for it to be adapted to new situations the experts need to manually analyse the network data and adapt their knowledge to new attacks. We presented in a previous work in nIDSs [17] three different techniques for payload processing. The three options were able to efficiently detect some of the attack types. This work showed that general payload analysis can be effective but the best results were always achieved including NCD sequence comparison method [14] for payload processing. The present work improves it in two senses:

– Any of the histogram based payload processing strategies achieved greater AUC values than NCD, MFN or MFO options. They even outperformed the results achieved with specific payload processing, Content variables obtained adhoc for the Kddcup99 database based on experts' experience.
– The new option is computationally cheaper

9 Conclusions and further work

The experimentation presented in this paper proves that a general payload processing methodology, histogram representation, is more efficient than the specific payload processing done in Kddcup99 for intrusion detection in an unsupervised anomaly detection context. The proposed payload processing options were able to detect more attacks in both cases: when used on their own and when combined with the information of intrinsic and traffic variables. The best option, IT+HM+HO, achieved an average AUC of 0.955 whereas the average AUC achieved with the best option for adhoc processing, IT+C, was 0.91. Furthermore, it was able to detect any kind of attacks because the minimum AUC, taking into account all types of attacks, was 0.89. And besides, the proposed options were not computationally expensive since they require few mathematical operations.

The way in which classifiers can be combined is an area where a deeper analysis can be carried out and more sophisticated approaches tried. The possibility of using other clustering algorithms and the optimisation of their parameters is also an area where more work can be done. In the same way, we could make the system computationally more efficient by minimising the amount of information kept on the signatures. We could keep just the bins higher than concrete threshold value in the signature. But in this case we would lose information so we would need to evaluate the trade-off.

Acknowledgements. This work was funded by the University of the Basque Country, DAMISI project (EHU 08/39), the Diputacin Foral de Gipuzkoa and the E.U.

References

1. Ashfaq B., Robert Ma . J., Mumtaz A., Ali M. Q., Sajjad A., Khayam A.: A Comparative Evaluation of Anomaly detectors under portscan Attacks. Proc. of RAID 2008. (2008)
2. Dagorn N.: WebIds: A Cooperative Bayesian Anomaly Based Intrusion Detection System for Web Applications. Proc. of RAID 2008. (2008)
3. DARPA: MIT Lincoln Laboratory - DARPA Intrusion Detection Evaluation, `http://www.ll.mit.edu/IST/ideval/index.html`. Accessed 29 Sep 2008. (1998)
4. Eskin E., Arnold A., Prerau M., Portnoy L., Stolfo S.: A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabelled data. Data Mining for Security Applications. (2002)
5. Fawcett T.: ROC graphs: notes and practical considerations for researchers. Technical Report HPL-2003-4. HP Laboratories, Palo Alto, CA, USA (2004)

6. Hodge V.J., Austin J.: A Survey of Outlier Detection Methodologies. Artificial Intelligence Review 22, 85–126 (2004)
7. Jacobson V., Leres C., McCanne S.: Tcpdump. Available via anonymous ftp to ftp.ee.lbl.gov. Accessed 29 Sep 2008. (1989)
8. Kamarainen J.K., Kyrki V., Ilonen J., Kälviäinen H. : Improving similarity measures of histograms using smoothing projections. Pattern Recognition Letters 24, 2009-2019 (2003)
9. KDD99-Cup: The third international knowledge discovery and data mining tools competition dataset, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed 29 Sep 2008. (1999)
10. Krügel C., Toth T., Kirda E.: Service specific anomaly detection for network intrusion detection. Proc. ACM Symposium on Applied Computing. (2002)
11. Lee W., Stolfo S.J., Mok K.: Data mining in work flow environments. Experiences in intrusion detection. Proc. of the Conference on Knowledge Discovery and Data Mining. (1999)
12. Lee W.: A data mining framework for constructing features and models for intrusion detection systems. Ph.D. thesis, Columbia University (1999)
13. Leung K., Leckie C.: Unsupervised anomaly detection in network intrusion detection using clusters. Proc. Australian Computer Science Conference (2005)
14. Li M., Chen X., Li X., Ma B., Vitanyi P.M.B.: The similarity metric. IEEE Transactions on Information Theory 50, 3250–3264 (2004)
15. Noh S., Jung G., Choi K., Lee C.: Compiling network traffic into rules using soft computing methods for the detection of flooding attacks. Applied Soft Computing 8(3), 1200–1210 (2008)
16. Paxson V.: Bro: a system for detecting network intruders in real-time. Computer Networks 31, 23–24 (1998)
17. Perona I., Gurrutxaga I., Arbelaitz O., MartínJ.I., Muguerza J., Pérez J.M.: Service-independent payload analysis to improve intrusion detection in network traffic. Proc. of the 7th Australasian Data Mining Conference (AusDM08). (2008)
18. Portnoy L., Eskin E., Stolfo S.: Intrusion detection with unlabeled data using clustering. Proc. ACM Workshop on Data Mining Applied to Security. (2001)
19. Rehák M., Pechoucek M., Bartos K., Grill M., Celeda P., Krmícek V.: Improving Anomaly Detection Error Rate by Collective Trust Modeling. Proc. of RAID 2008. (2008)
20. Serratosa F., Sanfeliu A.. Signatures versus histograms: Definitions, distances and algorithms. Pattern Recognition 39, 921–934 (2006)
21. Spath H.: Cluster analysis algorithms. Ellis Horwood, Chichester, UK (1980)
22. Strelkov, V.V.: A new similarity measure for histogram comparison and its application in time series analysis. Pattern Recognition Letters 29, 1768–1774 (2008)
23. Sung-Hyuk Cha, Sargur N. Srihari: On measuring the distance between histograms. Pattern Recognition 35, 1355–1370 (2002)
24. Waizumi Y., Tsuji M., Tsunoda H., Ansari N., Nemoto Y.: Distributed Early Worm Detection Based on Payload Histograms. Proc of the IEEE International Conference ICC'07. (2007)
25. Wang K., Stolfo S.: Anomalous payload-based network intrusion detection. Proc. International Symposium on Recent Advances in Intrusion Detection. (2004)
26. Warrender C., Forrest S., Pearlmutter B.: Detecting intrusions using system calls: alternative data models. Proc. of IEEE Symposium on Security and Privacy. (1999)

Knowledge Discovery Methodology for Medical Reports

Vitor Pinheiro and Victor Alves

Departamento de Informática,
Universidade do Minho
vitor.amares@gmail.com, valves@di.uminho.pt

Abstract. Medical reports contain valuable information, not only for the patient that waits for the results but also the latent knowledge that is possible to extract from them. The recent introduction of standard structured formats like the Digital Imaging and Communications in Medicine Structured Report and the Clinical Document Architecture Health Level Seven provide an efficient generation, distribution, and management mechanism. Also, they provide an intuitive and effective manner of information representation, unlike the traditional plain text format. In this paper we present a knowledge discovery methodology for structured report interchange based on plain text medical reports using YALE, a leading open-source data mining tool and Open-ESB platform that provides conversion, parsing, different protocols and message formats interchange capabilities.

Keywords: SOI, SOA, Open-ESB, BPEL, Data Mining, YALE, DICOM, HL7.

1 Introduction

Medical reports are the support and contact point between the diagnosis, the knowledge transmission of the specialist physician, and the patient or other physician. In the majority of the studies, these reports are plain text information without any standardized structure or contents sharing, that could be achieved by the usage of medical images attached to the reports. A medical report is an individual case study without any type of general correlation or association with others studies or diagnosis. These cases can be considered as epidemic studies for a given population, population sample or other option.

With the establishment of open medical standards, like DICOM - Digital Imaging in Medicine [1] and HL7 - Health Level Seven standard [2] it is now possible to integrate heterogeneous systems and medical information [3]. DICOM defines standards for exchanging, storing and printing medical imaging related information that can be used by medical equipment. HL7 defines standards to interconnect and control clinical and administrative data, of the whole life cycle of a patient clinical documentation, using the HL7 Clinical Document Architecture (CDA).

In addition to the DICOM standard definition of acquisition and storage of waveforms and images, DICOM also includes Structured Reporting (SR), a great expressive and hierarchical representation of structured medical information, containing text with links to other data such as images, waveforms, and spatial or temporal coordinates [4]. However, the medical report as we know it is static, delivered on paper and sometimes without any kind of format or in accordance with any standard. In general, this kind of medical information is treated only by the physician that asked for the medical exam and the physician who diagnosed it. There's no correlation or other kind of data analysis or knowledge discovery within this medical information.

One initial goal for the SR was that the information encoded in such reports would become more readable and thus easier to extract, then an unstructured plain text or paper report [5], making it easier to index and to selective retrieve information, without having to rely on Natural Language Parsing (NLP) [6]. Also, SR encoding and structure allows queries and data mining operation such as a query for all documents where a malignant mass of a specific dimension is reported. These operations are supported because every element of information is described by a code. A code value that can be unambiguously identified enabling data mining [4] using a data mining tool like YALE (Yet Another Learning Environment), as formally called, a data mining platform (environment) that simplifies the construction of experiments and the evaluation of different approaches. Using YALE is possible to build different experiments models to discover relations and patterns within this data, in the medical report could provide new medical knowledge, applying methods that have been developed to discover this hidden medical knowledge [7].

Nevertheless, for external communications of these medical reports with other departments it is necessary to use HL7. According to [8] DICOM SR is a matter of primary interest to HL7 CDA for different reasons. In the practical level, generally, the end users of SR are referring physicians using HL7-based systems. The SR usability requires a method of exporting these results to the HL7 domain. Although, HL7 and DICOM have joined efforts to adjust CDA and SR to avoid incompatibilities, there still is no mechanism of bi-directionally trans-coding, SR to CDA, with full fidelity defined by either group, and may never be defined, nor do they exists for plain text to structured report [5]. All these problems are common in Enterprise Application Integration (EAI). EAI technology enables incompatible protocols and messages formats to be exchanged by different entities [9]. Open-ESB [10] is a platform built using open standards like JBI [19] that can be used as a platform for both Enterprise Application Integration and Service Oriented Architecture applications development. The Open-ESB architecture enables communication between different protocols and messages, synchronous and asynchronous, interoperability and scalable applications. This advantages can solve problems related with medical reports, for example, like the trans-coding bi-directionally mechanism between DICOM and HL7.

In this paper we present a methodology for knowledge discovery for structured report interchange based on plain text medical reports using YALE [11] based on the Weka [12] tool and Open-ESB [10] platform that provides conversion, parsing, different protocols and message formats interchange capabilities.

2 Medical Reports

DICOM Structured Reports can be used for different purposes with different levels of complexity. These different levels of complexity are related with the target report or with the diagnosis complexity. So we have three different types of structured report classes:

- *Basic Text* – minimal code use, for document title, subtitles and hierarchical subtitles tree
- *Enhanced* – superset of the Basic Text, numerical measurements with representative codes for units and measurements, images and waveforms references.
- *Comprehensive* – superset of the Basic Text and Enhanced, references between elements.

The information in SR is grouped in nine modules, in which the items of information are related. There is a module for information on the patient, such as date of birth and weight, a module for the general information compliance with the document, such as names of people responsible for verifying the document and flags that indicate whether the document was found complete, and so on. Although, SR documents are not necessarily on a patient, may be on a sample as a sample of human tissue for analysis. The information contained in the contents module of the document is divided in "contents items". A content item consists of a pair name-value, where the name is a code of a selected dictionary of terms like the SNOMED CT - Systematized Nomenclature of Medicine-Clinical Term [13] and the value is a type among the fourteen types of value defined by default. These types are *text* (for text), *num* (for numbers, percentages), *image*, *date* and *waveform*. The items are hierarchically organized, so that the information in the highest levels of the hierarchy contains, or is derived from, information on items below the same. The DICOM SR standard specifies eight different types of relationships, among them are *contains* (node father of the information is contained in node child), *has properties* (information to the node is a child's property information node of the father), *has obs. Context* (the information in the node is a child comment on the information of the node father). The following sentence and examples adapted from [14], in natural language, could be divided into items of information and organized in a hierarchy:

```
Personal Habits: In average the patient drank 7 beers
per day in the last 2 years. The patient is obese, and
often passes away…
```

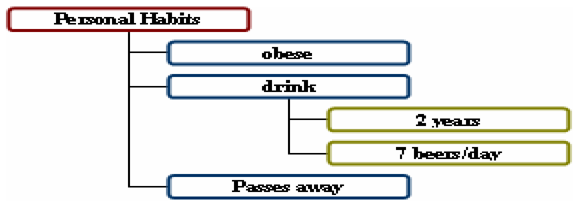


Fig. 1. Personal Habits example – the information was divided according to the most important concepts.

In DICOM SR each item should be a pair name-value, each containing a relationship with its father item and a value type, indicated above the item (Fig. 2).

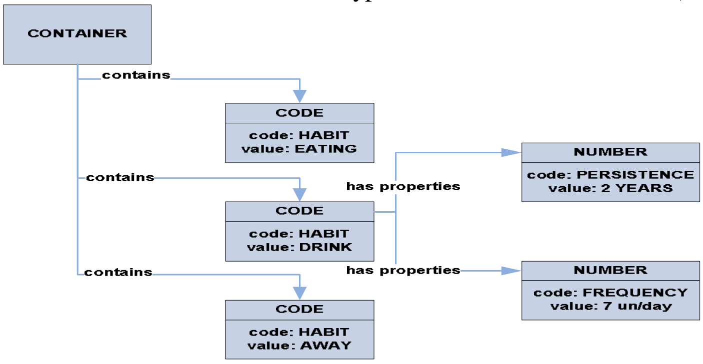


Fig. 2. Items hierarchy according to DICOM SR.

The biggest question in the structured reports is the integration and acceptance by the physician for this way representing the diagnosis. Most of them have there own templates, defined in a dot file type, and do not want to be restrained by any standard format. In our case study we use CT (Computer Tomography) reports, in text mode, that have an associated template.

3 State of Art

The introduction of open standards like HL7 and DICOM present expressive representations of hierarchical structures of medical information, capable of containing text with links to other structured data such as images and described by codes, that can be unambiguously identified enabling data mining [4]. However, the physicians continue to write their reports ignoring these standards or simply do not know of their existence. Thus, their reports can suffer from the ambiguous terms of natural language [15] and sometimes do not address the key clinical question [16], containing clinically important errors [17] despite the fact that a study showed that referring physicians strongly prefer concise well-organized radiology reports [18].
There is no well known mechanism of bi-directionally trans-coding SR to CDA with full fidelity formally defined [5]. This problem is common in Enterprise Application Integration (EAI). EAI technology enables incompatible protocols and messages formats to be exchanged by different entities [9].
In [25] we can find a proposal for a text mining system to extract and use the information in radiology reports that consists of three main modules: a medical finding extractor, a report and image retriever, and a text-assisted image feature extractor. In our approach these modules are included as the information integration from different sources, protocols, message formats or standards using service oriented integration architecture.

3.1 Service Oriented Integration

Common problems in enterprise application integration are the incompatible protocols and messages formats. In response to this kind of problems, the actual industry path is based on standards definition for business integration and standard metadata in the web services stack [9].
The JBI 1.0, JSR-208 [19], specification is an industry-wide initiative to create a standardized integration platform for Java and business applications addressing Service-Oriented Architecture (SOA). JBI employs concepts similar to J2EE to extend application packaging and deployment functionality to include JBI Components. JBI Components are an open-ended class of components that are based on JBI abstract business process metadata [19].

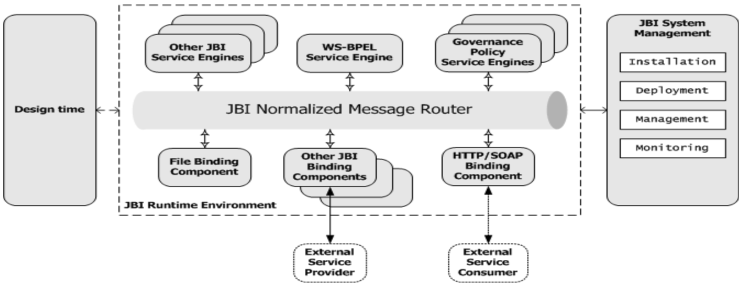


Fig. 3. The JBI Environment [20].

According to [20] the key pieces of the JBI environment (Fig. 3) are:

- *Service Engines* - enable pluggable business logic;
- *Binding Components* - enable pluggable external connectivity;
- *Normalized Message Router* – directs normalized messages from source components to destinations according to specified policies.

Java Business Integration (JBI) provides a foundation for building a SOA and is the foundation of the Open-ESB platform. It allows anyone to create JBI-compliant integration plug-in components and integrate them dynamically into the JBI infrastructure. Despite this, JBI alone does not have a single point of administration for the entire system and each operation in the system requires knowledge of the system topology. The Open-ESB solve these problems using the Java Open Enterprise Service Bus built with JBI technology enabling a set of distributed JBI instances to communicate as a single logical entity that can be centrally managed. Using the Open-ESB it's possible to integrate existing business functions as services, and decoupled interaction between service providers and consumers. It provides direct support for composite application creation through the mechanism of JBI service assemblies, which allow applications to be composed directly from the service-based interfaces of JBI service units and BPEL (Business Process Execution Language) orchestrated [20].

This direct support for composite application construction atop a service-oriented architecture and standards-based messaging infrastructure makes JBI an ideal foundation for constructing service-oriented applications and accomplishing service-oriented integration of existing systems using normalized messages in XML [9].

Integrating computing entities using only service interactions in a service-oriented architecture is defined as Service-Oriented Integration (SOI). Service-oriented integration solutions deal better with integrating legacy problems and inflexible heterogeneous systems using more often functionalities that were hidden in different applications as reusable services. The main advantages, for the traditional enterprise application integration (EAI) are the application of standards to define standards interfaces, the opaqueness of the functionality that is hidden from the service interface and the flexibility of the service in the perspective of the consumer and producer that can change except the description of the service.

4 Knowledge discovery methodology

The free plain text medical report cannot just simply be abandoned and substituted by the structured report. This does not just happen and will not happen until the physician is comfortable with standards or with tools that generate or help to write a structured report [14]. Beside this, the physician will have there own template for each possible diagnosis with his own personal mark. Also we must not forget the existent legacy plain text file medical reports that contain medical information that are valued information in future exams. Knowing that and in accordance with the presented standards for integration and medicine it's possible to suggest an information integration and knowledge extraction corresponding to the physician desires, the new structured report, all the platforms and medical systems related to this scope and obtain any extra information with the knowledge discover possibility.

We present a methodology definition for knowledge discovery supported by Service Oriented Integration architecture, the set of rules or principles that should be followed to discovery latent information in the medical reports, like patterns and association rules:

1. Use a service oriented integration architecture able to connect different services, that can use different protocols and messages;
2. Obtain a encode and decode service of a free text medical report to a normalized format, based in a model;
3. Achieve a mapping between medical terms used in the medical report and a codification non ambiguous solving problems concerning the use of language natural in medical reports (denial);
4. Apply knowledge discovery algorithms to the normalized integrated information e stored in the database.

Considering the medical reports as legacy files stored in a FTP (File Transfer Protocol) server we can orchestrate in BPEL a business process were the medical reports are encoded and the structured information saved on a database for future data mining operations using YALE.

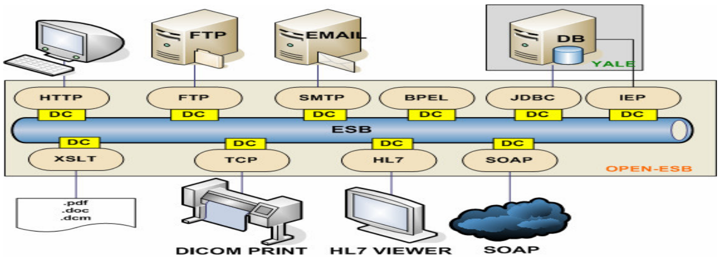


Fig. 4. Overall of the Medical Report Integration Architecture.

One possible integration process (we can have many others) is based on the architecture illustrated by Fig. 5 that is supported by the Open-ESB platform. The first step for this integration process (Fig. 6) is the oriented template parser definition. Since they all have been created based on a template we can parse the medical report using a defined XSD for each physician template and return all the information in a structured format from the FTP server. The encoded operation permits the interchange of the information between the systems and the possible conversion to a SR. Once the information is normalized it will be published to the Java Open Enterprise Service Bus (using the Direct Channel connection) and will be subscribed by different destinations (bindings) or used in other business logic (service engines). It will be possible to transform the medical reports in different file formats, send them by email or exhibit them in a HL7 viewer.

Once the information is encoded we need to describe the natural language, Portuguese, medical terms by a code. A code value that can be unambiguously identified enabling data mining [4] making easier the indexing and selective retrieval, without having to arrange to NLP - Natural Language Parsing [6].

The codification could be done using the SNOMED CT coding system but this is a proprietary medical term library. Thus we decided to code the medical terms with internal mapping defining a *CODE* service for this codification.

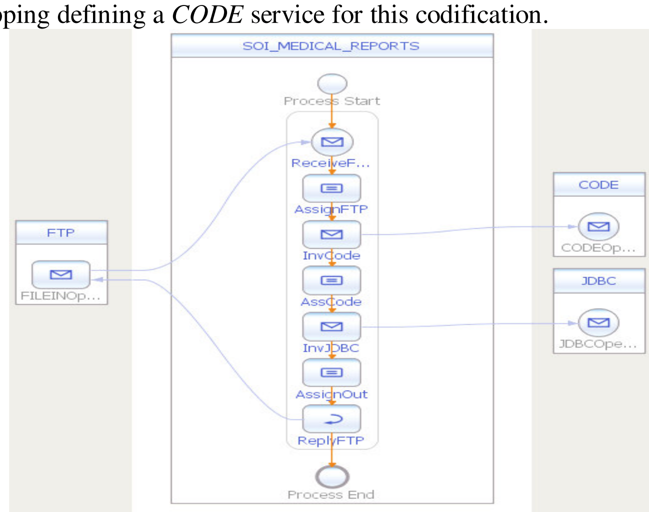


Fig. 5. BPEL Medical Reports Integration example.

The major codification difficulty is the negation terms common in natural language and in the medical reports. According to [21] the referring physicians that just say “normal” are considered as somehow inferior or incomplete. Therefore the referring physicians can also list things that are not found, even on entirely normal examinations. For example consider a report where the referring physician does not find any atrophy but found a stroke. Using our internal codification we unambiguous indentify the medical terms and encode also negation terms. For example the *CODE* service can receive the following (part) message:

```
<finding att="Not Found">atrophy</finding>
   <finding att="Found">stroke</finding>
```

The mapping that we have to these medical terms is:

Table 1. Internal map Medical Terms (part).

Medical Term	Found	Not Found
Atrophy	110	110N
Focal atrophy	1101	1101N
Stroke	120	120N
Lacunar stroke	1201	1201N

So, after receiving that message and in accordance to Table 1 the *CODE* service will reply the following (part) message:

```
<finding>110N<finding>
<finding>120</finding>
```

Therefore, the information saved on the database will facilitate the use of YALE since it is now identified by unambiguous and unique code without the usage of natural language and encoding the negation terms.

The integration process here presented is one of many possible others. For example, we can have two parallel paths for this integration process. One for taking care of the message codification before being saved in the database. And another to save the information as it is in the database without coding. Facilitating data mining and maintain history of the processed information.

After this integration process using Data Mining techniques and Machine Learning algorithms that YALE provides we can discover association rules between different medical reports. This technique of Data Mining makes it possible to identify patterns in large databases. This identification of patterns helps the gathering and interpretation of obtained results, to acquire the specific knowledge to a conclusion or assumption for the case study.

The YALE association rules discover model uses a database connector, a pre-processing operator chain, a learner unsupervised item set and association rule generator algorithm. The major pre-processing operators are the frequency discretization, filter operator nominal to binominal operator and missing value replenishment. The frequency discretization, discretizes numerical attributes by putting the values into bins of equal size. The filter operator nominal to binominal creates for each possible nominal value (YALE considers the negated terms as nominal) of a polynomial attribute a new binominal (binary) feature which is true if the example had the particular nominal value. These pre-processing operators are necessary since particular learning schemes can not handle attributes of certain value types. The next operator is the frequent item set mining operator *FPGrowth* [22]. A major advantage of *FPGrowth* compared to *Apriori* algorithm is that it uses only 2 data scans and is therefore often applicable even on large data sets [23]. This operator efficiently calculates attribute value sets often occurring together. From these so called frequent item sets the most confident rules are calculated with the association rule generator.

4.1 Case study

In our case study we have three different types of CT reports from three different physicians, available in html format. Only 100 records were kindly provided by CIT (Centro de Imagem da Trindade) for this analysis and all the records are from different patients without repetition. The information of these reports is previously extracted and normalized in the first step of the integration process to facilitate the codification process. The age of the patients were grouped by the following range:

- Baby [0;4]
- Child [4;12]
- Young [12;26]
- Adult [27;65]
- Senior [65;]

Since we are dealing with natural language, in Portuguese, there are many possible ways to represent the same word, with or without accents and lower or upper case characters we transform all characters to lower and convert the accents to the corresponding ASCII character, and e.g., the "Ç" convert in a "c". In general, these medical reports contain a patient module: patient id, sex and age. Also indicates the number of series, modalities and notes. Follow that we have the protocol(s) and the finding(s) with conclusion or not. So each item set consist of one or more patient module, series, modalities, notes, protocols, finding and conclusion codes. With this information we try to find some association rules between the followed procedures and the diagnosis. Despite the number of medical reports, each one can have many types and more than one note, protocol or finding that increase the complexity for the find association rules.

4.2 Results

Using the YALE association rules discover model for the case study presented we obtained for example some interesting frequent item sets from our case study:

- 75% of the patients are adult women
- 86% are normal
- 69,4% are normal when not found median deviation (901N) and found permeable ventriculo-cisternal system (80111).

Despite finding many association rules for the medical procedures and respective diagnosis that can be considered expected we selected two particular rules:

Table 2. Association Rules Medical Reports (part).

Rule	1	2
Premises	Notes=30	protocol= 20111211
Conclusion	protocol= 20111211	conclusion= C10
Support	61%	75%
Confidence	81,5%	96,4%

The rules presented in Table 2 show us a curious fact that the medical report notes can have an important role in the definition of the protocol and indirectly influence the diagnosis. As you can see Rule 1 (notes=30 => protocol=20111211 [61%; 81,5%]) we have 61% of our samples have low back pain and the protocol was axial plans contiguous of 10mm parallels to the orbit-meatal plan and the confidence for this is 81,5 %. For Rule 2 (protocol1=20111211 => conclusion=C10 [75%; 96,4%]) we have that the protocol axial plans contiguous of 10mm parallels to the orbit-meatal plan in 75% of cases we will have a normal exam with 96,4 % of confidence.

5 Conclusion

The knowledge discover methodology for medical reports supported by a service oriented integration architecture here presented is a response to problems raised from the legacy medical report processing to enable knowledge discovery using industry, business and medical open standards.

A solution for typical problems presented by legacy medical reports like ambiguous terms of natural language, lack of structure, hard inter-application exchange due to business logic and trans-coding of incompatible protocols and messages formats was presented.

The legacy reports encoded based on the physician templates proved to be an excellent parsing format because all physicians have there own template making easier the conversion to a structured report format. In a first approach we tried to apply XSLT transformation to the OFFIS XML Schema *dsr2xml.xsd* [24] but the legacy medical reports lacks of mandatory information required by OFFIS.

The decision to code the medical terms with internal mapping defining a service for this codification without using a coding system like SNOMED CT solves the encoding negation problem using a specific code concept name or value. This concept of negation is crucial because it's equally important to be able to say "no stroke" as it is to say "stroke".

The results obtained from information integration shows some obvious results like that bone fractures are more common in older people or the normal exams are more common in younger people. However this result also allowed concluding that the notes taken before the realization of the exam influenced the protocol and this in turn the diagnosis.

Currently we are working on:

- OFFIS *dsr2xml.xsd* - mapping the entire medical report information to this structure. For future conversion the contents of a DICOM Structured Reporting (SR) document (file format or raw data set) to XML;
- DICOM Open-ESB binding component development – at the moment the communication is only possible by TCP/IP binding;
- Intelligent Event Processor Open-ESB – at the moment the data pre-processing treatment is done with YALE. Using the IEP withdraw the YALE load;
- Geographical Data Mining - enabling epidemic studies.

Acknowledgements

We are indebted to CIT – Centro de Imagiologia da Trindade, for their help in terms of anonymous medical reports, experts, technicians and machine availability.

References

1. *Digital Imaging and Comunications in Medecine.* Retrieved August 1, 2008, from http://medical.nema.org/.
2. *Health Level Seven*. Retrieved August 1, 2008, from http://www.hl7.org/.
3. Dreyer, Keith J., Mehta, Amit, Thrall, James H. *PACS: a guide to the digital revolution*, Springer, New York (2002).
4. Noumeir, Rita. *Benefits of the DICOM Structured Report*, Journal of Digital Imaging, 295-306, Springer (2006).
5. Clunie, David A. *DICOM Structured Reporting and Cancer Clinical Trials Results*, Cancer Informatics 2007:4 33-56 (2007).
6. Langlotz, C.P. *Automatic Structuring of Radiology Reports: Harbinger of a Second Information Revolution in Radiology*, Radiology 224:5-7 (2002).
7. Prather, J. C., Lobach, D. F., Goodwin, L. K., Hales, J.W., ML Hage, Hammond, W. E. *Medical Data Mining: Knowledge Discovery in a Clinical Data Warehouse,* Proceedings of the AMIA Annual Fall Symposyum, Vol. 101, No. 5 (1997).
8. Behlen, Fred M. *DICOM Structured Reporting and the CDA Tutorial*, HL7 International CDA Conference, Berlin (2007).
9. Ten-Hove, Ron. *Using JBI for Service-Oriented Integration (SOI).* Retrieved July 1, 2008, from https://open-esb.dev.java.net/public/whitepapers/JBIforSOI.pdf.
10. Open-ESB. *The Open Enterprise Service Bus.* Retrieved July 1, 2008, from https://open-esb.dev.java.net/.
11. Mierswa et all. *YALE: Rapid Prototyping for Complex Data Mining Tasks*, in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06).
12. Witten, Ian H. and Frank, Eibe. *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco (2005).
13. SNOMED CT. S*ystematized Nomenclature of Medicine-Clinical Terms*. International Health Terminology Standards Development Organisation Retrieved August 1, 2008, from http://www.ihtsdo.org/snomed-ct/.

14. Bortoluzzi, Mariana Kessler. *Desenvolvimento e Implementação de um Editor de Documentos Estruturados no padrão DICOM Structured Report.* Universidade Federal de Santa Catarina, Florianópolis (2003).
15. Kong, A., Barnett, G., Mosteller, F., et al. *How medical professionals evaluate expressions of probability*. N Engl J Med; 315:740–744 (1986).
16. Lussier, Y. A., Shagina, L., Friedman, C. *Automating SNOMED coding using medical language understanding: a feasibility study*. Proc AMIA Symp, 418–422 (2001).
17. Holman, B., Aliabadi, P., Silverman, S., et al. *Medical impact of unedited preliminary radiology reports*. Radiology 191:519–521 (1994).
18. Naik, S., Hanbidge, A., Wilson, S. *Radiology reports: examining radiologist and clinician references regarding style and content.* AJR (American Journal of Roentgenology) 176:591-598 (2001).
19. Ten-Hove, Ron, Walker, Peter. *Java Specification Requests 208: Java™ Business Integration (JBI).* Retrieved July 1, 2008, from http://jcp.org/en/jsr/detail?id=208.
20. Raj, Gopalan Suresh, et all. *Implementing Service-Oriented Architectures (SOA) with the Java EE 5 SDK*. Retrieved July 1, 2008, from http://java.sun.com/developer/technicalArticles/WebServices/soa3/ImplementingSOA.pdf.
21. Clunie, David A. *DICOM Structured Reporting*, PixelMed Publishing, Bangor, Pensilvania. ISBN 0-9701369-0-0 (2001).
22. Han, J., Pei, J., Yin, Y. *Mining frequent patterns without candidate generation*. In Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'00), pages 1-12, Dallas, TX (2008).
23. YALE. *RapidMiner Developer Guide*, Rapid-I, Dortmund, Germany. Retrieved July 1, 2008, from http://downloads.sourceforge.net/yale/rapidminer-4.2-tutorial.pdf.
24. Kuratorium OFFIS. *dsr2xml: Convert DICOM SR file and data set to XML.* Oldenburg, Germany. Retrieved July 1, 2008, from http://support.dcmtk.org/docs/dsr2xml.html
25. Tianxia Gong, Chew Lim Tan, Tze Yun Leong, Cheng Kiang Lee, Boon Chuan Pang, C. C. Tchoyoson Lim, Qi Tian, Suisheng Tang, Zhuo Zhang, "Text Mining in Radiology Reports," icdm, pp.815-820, 2008 Eighth IEEE International Conference on Data Mining, 2008.

A Study on Change Detection Methods

Raquel Sebastião[1,2] and João Gama[1,3]

[1] LIAAD-INESC Porto L.A., University of Porto
Rua de Ceuta, 118 - 6, 4050-190 Porto, Portugal
[2] Faculty of Science, University of Porto
[3] Faculty of Economics, University of Porto
{raquel,jgama}@liaad.up.pt

Abstract. In the real word, the environment is often dynamic instead of stable. Usually the underlying data of a problem changes with time, which enhances the difficulties when learning a model from data. In this paper, different methods capable to detect changes from high-speed time changing data streams are compared. These methods are appropriated to be embedded inside learning models, allowing the adaptation to a non-stationary problem. The experimental evaluation considers different types of concept drift and data streams with different properties. Assessing measures such as: false alarm rates, number of samples until a change is detected and miss detections rates, a comparison between the algorithms' capability of consistent detection is given. The choice on the best detection algorithm relies on a trade-off between the rate of false alarms and miss detections and the delay time until detection.

Key words: change detection, data streams, machine learning, monitoring data distribution.

1 Introduction: Motivation and Challenges

The most recent developments in Science and Information Technology lead to the considerably growing of the computational capacity of small devices, which are capable to produce a large massive amount of information at a high-speed rate. Along with this, as data flows over time for large periods of time, the process generating data is not strictly stationary and evolves over time.

The motivation for studying time-changing high-speed data streams comes from the emergence of temporal applications such as signal processing, time series analysis, sensor networks, automatic control, real time-monitoring in biomedicine and industrial processes, fraud detection, user modelling, safety of complex systems and many others.

Due to the dynamic nature of data, some properties of the problem can change over time, namely the target concept on which data is obtained may shift from time to time, each time after some minimum of permanence. Learning algorithms that model time-changing underlying processes must be able to track the dynamic behaviour and adapt the decision model accordingly. For example, in a prediction task, it is reasonable to assume that a change in the underlying data distribution will increase the error rate thus making the predictions reflect characteristics that are no longer hold. Since old observations (reflecting the behaviour of nature in the past) became irrelevant to the current state, after the occurrence of a change the prediction model need to be retrained to accurately predict the underlying actual data. These dynamic processes are challenging and need to be addressed with appropriated drift detection algorithms.

The nature of change is another important and challenging issue. Changes may occur due to modifications in the context of learning (caused by changes in hidden variables) or in the intrinsic properties of the observed variables. Usually, literature considers two types of drift. The term *Concept Shift* refers to abrupt changes (for example, the patterns of costumers' buying preferences that may change with seasons), while the term *Concept Drift* is associated to gradual changes in the target concept (for example, small faults in parts of an industrial process can modify the quality of the product). *Concept Drift* is more difficult to detect, and at least in the initial phases it may be confused with noise.

The main challenge of algorithms for change detection is the combination of *robustness* to noise with *sensibility* to concept change. Noise and outliers pose difficulties and challenges to drift detection algorithms, and may increase false alarm rates. The scope of this paper is the study of different methods for drift detection, comparing their capacities under different types of concept drift, different data streams' lengths and different scenarios.

The paper is organized as follows: next section presents the related work in the field of change detection in data streams; section 3 presents the drift detection algorithms analyzed; section 4 describes the experimental evaluation and conclusions are presented in the last section.

2 The Change Detection Problem in Data Streams

When monitoring a stream it is fundamental to know if the received data comes from the distribution observed so far. Suppose a supervised learning problem, where the data consists of sequences of pairs $(\overrightarrow{x_i}, y_i)$ where $y_i \in C_1, C_2, ..., C_k$. At each time stamp t, the task of the learner is to output the class prediction $\hat{y}_t$ of that example. After checking the class y_t the error of the algorithm is computed. According with the *Probability Approximately Correct* (PAC) learning model [1] if the distribution of examples is stationary, the error rate of the learning model will decrease when the number of examples increases. A significant increase in the error rate suggests a change in the process generating data. For large periods of time, it is reasonable to assume that the process generating data will evolve. Whenever new concepts replace old ones, the old observations become irrelevant and thus the model will become inaccurate. In such case, the learning model must be adapted in accordance with the current state of the phenomena under observation.

Assuming that examples are independent and generated at random according to an unknown distribution, to assess if a concept is shifting over time, it is necessary to perform tests in order to determine if there is a change in the generating distribution. The null hypothesis is that the previously seen values and the current observed ones come from the same distribution. The alternative hypothesis is that they are generated from different distributions. Furthermore, such kind of change detection tests should detect only true changes with high probability, establishing a trade-off between false negatives and false positives. Specifically, these algorithms should:

– Be able to detect and react to drift.
– Not exhibit miss detections.
– Be resilience to false alarms (detect a change in stationary environments).
– Require few examples to detect a change after the occurrence of one.

2.1 Related Work

Since our environment is naturally dynamic, learning form high-speed time changing data streams is a considerably growing research field and several methods capable of dealing with concept drifts have been proposed [2–6, 11]. In general, approaches to cope with concept drift can be classified into two categories: (i) approaches that adapt a learner at regular intervals without considering whether changes have really occurred;(ii) approaches that first detect concept changes and afterwards the learner is adapted to these changes.

In the first approach, drifting concepts are often handled by time windows or weighted examples according to their age or utility. Weighted examples are based on the simple idea that the importance of an example should decrease with time (references about this approach can be found in [8–10]). When a time window is used, at each time step the learner is induced only from the examples that are included in the window. Here, the key difficulty is how to select the appropriate window's size: a small window can assure a fast adaptability in phases with concept changes. But in more stable phases it can affect the learner performance. On the other end, a large window would produce good and stable learning results in stable phases but can not react quickly to concept changes. In later approaches with the aim of detecting concept changes, some indicators (e.g. performance measures, data distribution, properties of the data, etc.) are monitored over time. A typical example is to monitor the evolution of a statistical function between two distributions: from past data in a reference window and in a current window of the most recent data points [2, 5]. If during the monitoring process a concept drift is detected, some actions to adapt the learner to these changes can be taken. When a time window of adaptive size is used these actions usually lead to adjusting the window's size according to the extent of concept drift [7]. As a general rule, if a concept drift is detected the window's size decreases; otherwise the window's size increases [2].

3 Change Detection Algorithms

This paper performs a comparison between four known algorithms taken from literature: the Statistical Process Control (SPC) presented in [3], the ADaptative WINDdowing (ADWIN) method introduced in [11], the Fixed Cumulative

Windows Model (FCWM), presented at [2] and a standard algorithm for change detection, the Page-Hinkley Test (PHT) [13].

3.1 Statistical Process Control (SPC)

This drift detection method controls online the trace of the probability of error for streaming observations. While monitoring the error, it defines a warning and a drift level. When the error exceeds the first (lower) threshold, the system enters in a warning mode and stores the time, t_w, of the corresponding observation. If the error drops below the threshold again, the warning mode is cancelled. However, if in a sequence of examples, the error increases reaching the second (higher) threshold at time t_d, a change in the distribution of the examples is declared. The classifier is retrained using only the examples since t_w and the warning and drift levels are reset.

Pseudo-code

Input:

labeled dataset x_1, . . . , x_t
warning threshold t_w (default t_w = 2)
detection threshold t_d (default t_d = 3)
warm-up window size w_0 (default w_0 = 30)

1. Initialize the minimum classification error $p_{min} = \infty$ and the corresponding standard deviation $s_{min} = \infty$. Set the warning zone flag, f_w, to false and w_1 =0.
2. For j = 1 to t - 1 (all the observations)
   If $j < w_0$ then
      $w_{j+1} = w_j + 1$ (warm up, only grow the window)
   Else
      i. Train a classifier on the current window of size w_j.
      ii. Classify observation w_{j+1}.
      iii. Update the error rate over the current window.
         Let $\hat{p}$ be the updated error rate and $\hat{s} = \frac{\hat{p}(1-\hat{p})}{w_j}$ be the updated standard deviation.
      iv. If $(\hat{p} + \hat{s}) < (p_{min} + s_{min})$ then update the minimum error by $p_{min} = \hat{p}$ and $s_{min} = \hat{s}$.
      v. If $(\hat{p} + \hat{s}) > (p_{min} + t_d \times s_{min})$ and f_w = true
         change has been detected, set up the detection time $t_d = j$.
         Take as the new training and detection window all the observations since t_w (size $w_{j+1} = j - t_w + 1$),
         set $p_{min} = \infty$, $s_{min} = \infty$ and $t_w = \infty$.
      ElseIf $(\hat{p} + \hat{s}) > (p_{min} + t_w \times s_{min})$
         If f_w = false
         switch the warning zone flag f_w = true and set up

```
            the warning time $t_w$ = j.
        Else
            set $f_w$ = false and update the window by adding
            $x_{j+1}$ to it (size $w_{j+1} = w_j + 1$).
3. Set $DT_{SPC} = t_w$.
```

Output: detection time DT_{SPC}.

3.2 ADaptive WINdowing (ADWIN)

The ADaptative WINDdowing method keeps a sliding window W (with length n) with the most recently received examples and compares the distribution on two sub-windows of W. Whenever two *large enough* sub-windows, W_0 and W_1, exhibit *distinct enough* averages, the older sub-window is dropped and a change in the distribution of examples is assigned. The window cut threshold is computed as follows:

$\epsilon_{cut} = \sqrt{\frac{1}{2m} ln \frac{4}{D}}$, with $m = \frac{1}{1/n_0 + 1/n_1}$, where n_0 and n_1 denote the lengths of W_0 and W_1.

A confidence value D is used within the algorithm, which establishes a bound on the false positive rate. However, as this first version was computationally expensive, the authors propose to use a data structure (a variation of exponential histograms), where the information on the number of 1's is kept as a series of buckets (in the Boolean case). It keeps at most M buckets of each size 2^i, where M is a design user-defined parameter. For each bucket, two (integer) elements are recorded: *capacity* and *content* (*size* or the number of 1s it contains). The method is detailed in [11].

Pseudo-code

Input:

```
    labeled dataset $x_1$, . . . , $x_t$
    confidence value $D \in (0,1)$
    bucket's parameter $M$

1. Initialize $W$ as an empty list of buckets
2. Initialize $\epsilon_{cut}$
3. for each $t > 0$
    do SetInput($X_t$, $W$)
        Output $u_W$ and $ChangeAlarme$

SetInput (item $e$, List $W$)
    1. InsertElement($e$, $W$)
    2. repeat DeleteElement($W$)
        until $|u_{W_0} - u_{W_1}| < \epsilon_{cut}$ holds
            for every split of $W$ into $W = W_0 \cdot W_1$

InsertElement (item $e$, List $W$)
```

```
    1. Create a new bucket b with content e and capacity 1
    2. W ← W ∪ b (add e to the head of W)
    3. Update ε_cut
    4. CompressBuckets(W)

DeleteElement(List W)
    1. Remove a bucket from the tail of list W
    2. Update ε_cut
    3. ChangeAlarm ← true

CompressBuckets(List W)
    1. Transverse the list of buckets in increasing order
       do If there are more than M buckets of the same capacity
          do merge buckets
             CompressBuckets(sublist of W not transversed)
```

3.3 Fixed Cumulative Windows Model (FCWM)

In a previous work [2] the FCWM was presented as a method to detect changes in data streams. To summarize data, it first constructs histograms using the two layer structure of the Partition Incremental Discretization (PiD) algorithm, which was designed to learn histograms from high-speed data streams [12]. The change detection problem is addressed by monitoring distributions in two different time windows: a reference window (RW), reflecting the distribution observed in the past; and a current window (CW) which receives the most recent data. In order to assess drifts, both distributions are compared using the Kullback-Leibler divergence (KLD), defining a threshold for change detection decision based on the asymmetry of this measure.
Pseudo-code

Input:

```
    labeled dataset x_1, . . . , x_t
    number of bins nBins
    number of observations in the current window InitialObs
    evaluation interval EvalInterval
    drift threshold λ

1. for t > 0
    1.1 If t > InitialObs then
          Compute the probability distribution p for the RW
    else
          Compute the probability distribution q for the CW
          From EvalInterval to EvalInterval compute
                A = |KLD(p||q) − KLD(q||p)|
          If A > λ ⇒ a drift is detected
```

```
        Return and report a change at time t_FCWM
    else
        Return to 1.1
```

Output: detection time t_{FCWM}.

3.4 Page Hinkley Test (PHT)

The Page-Hinkley test (PHT) is a sequential analysis technique typically used for monitoring change detection [13]. It allows efficient detection of changes in the normal behaviour of a process which is established by a model. The PHT was designed to detect a change in the average of a Gaussian signal [14]. This test considers a cumulative variable U_T defined as the cumulated difference between the observed values and their mean till the current moment:

$$U_T = \sum_{t=1}^{T}(x_t - \bar{x}_T - \delta)$$

where $\bar{x_T} = 1/T \sum_{t=1}^{t} x_t$ and δ corresponds to the magnitude of changes that are allowed. To detect increases, it computes the minimum value of U_t: $m_T = min(U_t, t = 1 \ldots T)$ and monitors the difference between U_T and m_T: $PH_T = U_T - m_T$. When the difference PH_T is greater than a given threshold (λ) a change in the distribution is assigned. The threshold λ depends on the admissible false alarm rate. Increasing λ will entail fewer false alarms, but might miss or delay some changes. Controlling this detection threshold parameter makes it possible to establish a trade-off between the false alarms and the miss detections.
Pseudo-code

Input:
labeled dataset $x_1, \ldots, x_t$
magnitude threshold δ
detection threshold λ

1. **for** $t > 0$
   1.1 Computes
   $\bar{x_T} = 1/T \sum_{t=1}^{t} x_t$
   $U_T = \sum_{t=1}^{T}(x_t - \bar{x}_T - \delta)$
   $m_T = min(U_t, t = 1 \ldots T)$
   If $PH_T = U_T - m_T > \lambda$
     return and report a change at time t_{PH}
   else
     return to 1.1

Output: detection time t_{PH}.

4 Experimental Evaluations

This section describes the evaluation of the mentioned methods. To assess the performance of change detection algorithms under different scenarios, different kinds of experiments with distinct purposes were implemented. The rate of false alarms, miss detections and delay time until detection were evaluated using: (i) data underlying a Bernoulli distribution, (ii) artificial datasets with distinct characteristics and (iii) a public dataset.

4.1 Artificial Data

The first set of experiments uses data streams of lengths $L = 2.000$, 5.000 and 10.000, underlying a stationary Bernoulli distribution of parameter $\mu = 0.2$ during the first $L - 1.000$ observations. During the last 1.000 samples the parameter is linearly increased using different slopes: 0(no change), 10^{-4}, 2.10^{-4}, 3.10^{-4} and 4.10^{-4}. These experiments also allow analyzing the influence (in the delay time until detections) of the length of the stationary part (the first $L - 1.000$ samples).

Table 1. Mean delay time until drift detection (DT), false alarms rates (FA) and the miss detections rates (MD), for the four methods, using the data streams with lengths 2.000, 5.000 and 10.000 and with different slopes in the Bernoulli parameter distribution. For *slope* = 0 (no change) the measurements DT and MD are not applicable.

Length	Slope	ADWIN			SPC			FCWM			PHT		
		DT	FA	MD	DT	FA	MD	DT	FA	MD	DT	FA	MD
2.000	0	(n.a.)	0.05	(n.a.)	(n.a.)	0	(n.a.)	(n.a.)	0.02	(n.a.)	(n.a.)	0.04	(n.a.)
	1.10^{-4}	581.6	0	0.03	626.6	0	0.02	853.8	0	0.37	573.0	0	0.03
	2.10^{-4}	577.6	0	0	687.2	0	0.16	894.8	0	0.59	522.9	0	0
	3.10^{-4}	428.4	0	0	536.9	0	0	647.0	0	0	397.3	0	0
	4.10^{-4}	358.6	0	0	534.4	0	0	616.3	0	0	331.3	0	0
5.000	0	(n.a.)	0.17	(n.a.)	(n.a.)	0.17	(n.a.)	(n.a.)	0.35	(n.a.)	(n.a.)	0.41	(n.a.)
	1.10^{-4}	721.9	0.16	0.30	866.4	0.21	0.77	705.5	0.27	0.64	649.6	0.23	0.13
	2.10^{-4}	512.0	0.12	0.13	732.3	0.19	0.37	674.7	0.18	0.14	462.9	0.25	0
	3.10^{-4}	382.6	0.14	0.14	667.9	0.20	0.17	572.8	0.31	0.04	337.0	0.32	0
	4.10^{-4}	320.1	0.10	0.10	587.1	0.09	0.12	501.8	0.35	0.06	279.2	0.29	0
10.000	0	(n.a.)	0.15	(n.a.)	(n.a.)	0.44	(n.a.)	(n.a.)	0.22	(n.a.)	(n.a.)	0.68	(n.a.)
	1.10^{-4}	721.6	0.19	0.35	829.3	0.39	0.94	723.7	0.27	0.69	649.6	0.60	0.10
	2.10^{-4}	505.0	0.19	0.19	842.7	0.56	0.57	687.4	0.31	0.36	466.0	0.71	0
	3.10^{-4}	401.3	0.17	0.17	719.5	0.29	0.53	597.4	0.29	0.29	343.9	0.68	0
	4.10^{-4}	327.2	0.23	0.23	642.2	0.52	0.42	481.0	0.24	0.24	280.2	0.66	0

As it is presented in table 1, rows are indexed by the value of L and corresponding slope. So, these rows present the delay time (DT) until the detection of the change that occurs at time stamp $L - 1.000$ (averaged over all runs), the miss detections rates (MD) and the false alarms rates (FA) for each algorithm studied. In general, the increase of the number of samples leads to an increase of the number of false alarms and miss detections. It can also be observed that as the number of samples increases, the percentage of changes detected decreases (with exception of the Page Hinkley Test). As it is reasonable, for all the algorithms, the increase on the slope of the Bernoulli's parameter contributes to a decrease of the time until the change is detected. As a final remark, the PHT maintains a good rate of miss detections, independent of the number of stationary past samples. This method presents smaller mean delay times, however these results are compromised with false alarms.

In the following set of experiments were used the following artificial datasets previously used in concept drift detection [15]:

- **SINE1**: Abrupt concept drift, noise-free examples.
- **SINIRREL1** Abrupt concept drift, noise-free examples, presence of irrelevant attributes (the same classification function of SINE1 but the examples have two more random attributes with no influence on the classification function).
- **CIRCLES**: Gradual concept drift, noise-free examples.
- **GAUSS**: Abrupt concept drift, noisy examples.

These artificial datasets have several different characteristics that allow assessing the methods' performance in various conditions: abrupt and gradual drift, presence and absence of noise and presence of irrelevant attributes. All the problems have two classes and each class is represented by 50% of the examples in each context. To ensure a stable learning environment within each context, the positive and negative examples in the training set are alternated. The number of examples in each concept is 1.000. To compute the error rate in these classification problems, the detection algorithms were embedded inside a decision tree classifier. Figure 1 presents the mean delay time for the four methods. In spite of the smaller delay for PHT, this method closely competes with ADWIN and SPC. On the other extreme, FCWM exhibits the highest delays. With exception of the ADWIN that presents a false alarm when using the SINE1 dataset, in all of the other cases there are no false alarms to report.

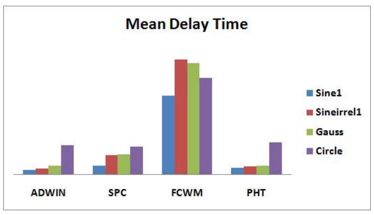


Fig. 1. Mean delay time until drift detection (DT) for the four methods, using the datasets SINE1, SINEIRREL1, GAUSS and CIRCLES.

4.2 Evaluation on a Public Dataset

In the previous experiments, the datasets did not allow checking the algorithms' performance in large problems, which is important since concept drift mostly occurs in huge amounts of data arriving in the form of stream. To overcome this drawback, an evaluation of the change detection algorithms was performed using the dataset SEA concepts [16], a benchmark problem for concept drift. Figure 2 shows the error rate (computed using a naive-Bayes classifier), which presents three drifts. The drifts and the corresponding detections are also represented by vertical lines. Table 2 presents the delay time in detection of concept drifts in this dataset (where the number of false alarms is indicated in parenthesis). One can observe that all the algorithms require too much examples to detect the first drift. However, with the ADWIN, the FCWM and PHT, the others are detected within a reasonable delay time. The resilience to false alarms and the ability to reveal changes without miss detections must be stressed out.

Table 2. Delay time in three drift scenarios using different change detection methods. The number of false alarms is indicated in parenthesis.

$\#Drift$	Delay Time			
	ADWIN	SPC	FCWM	PHT
1	826	3314	817	1404
2	115	607	273	118
3	242	489	249 (1)	256

5 Conclusions and Further Research

Regarding the delay time until detection and the miss detections, the Page Hinkley Test and the ADaptive WINdowing revealed to be the more appropriated algorithms to detect changes in the evaluated drift scenarios. However, in the first set of experiments the PHT presents a high rate of false alarms. Nevertheless, since learning algorithms run in fixed memory, time and memory consumption are also important constrains. It should be pointed out that PHT and Statistical Process Control are less time and memory consuming than the ADWIN and the Fixed Cumulative Windows Model, since they do not require any data structure to evaluate drifts.

Acknowledgements

The work of Raquel Sebastião is supported by the Portuguese Foundation for Science and Technology (FCT) under the PhD Grant SFRH/BD/41569/2007.

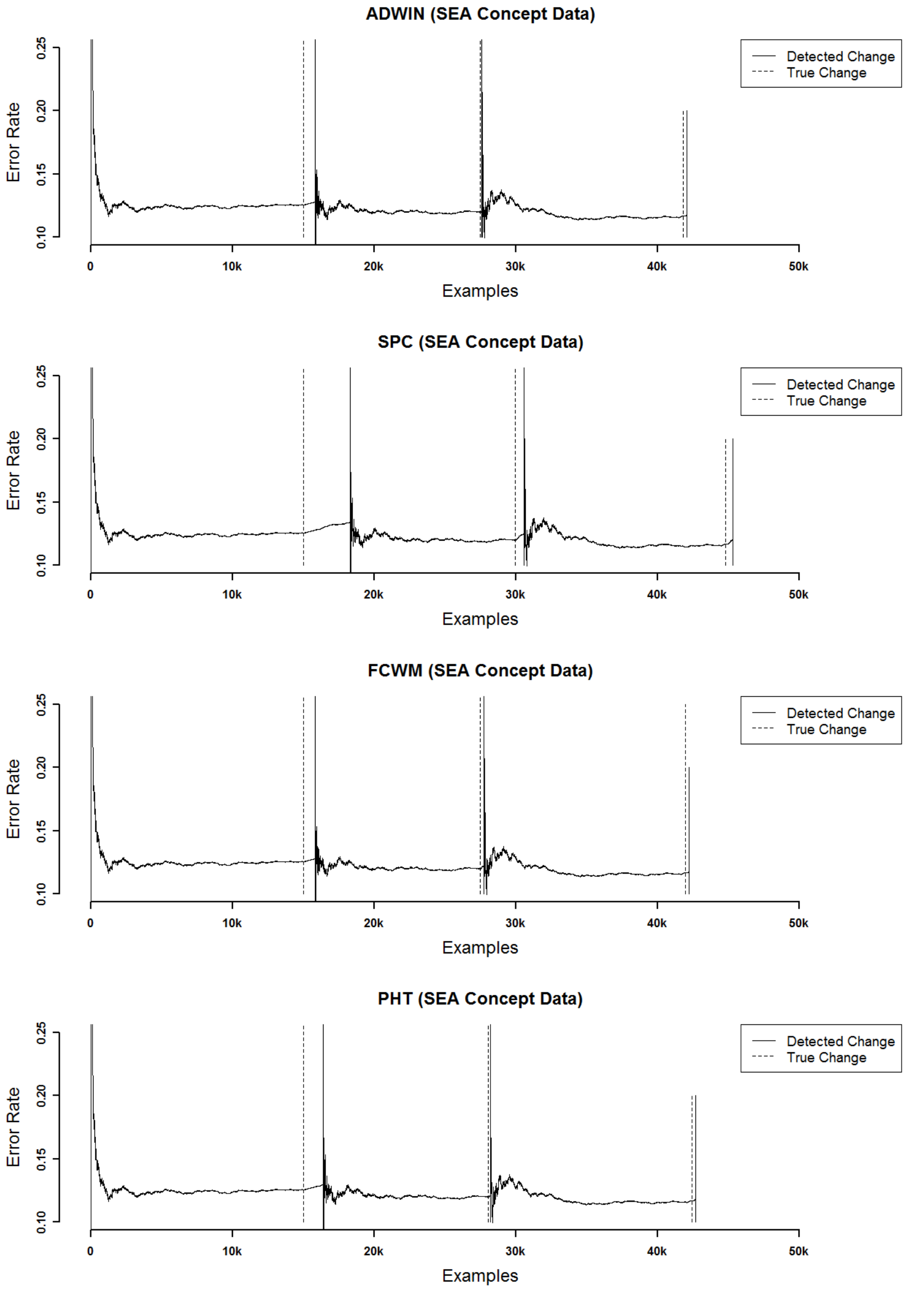


Fig. 2. Evolution of the error rate and the delay times in drift detection using the four presented methods. Vertical dashed lines indicate drift in data, and vertical lines indicate when drift was detected.

References

1. Tom Mitchell. Machine Learning. McGraw Hill, 1997.
2. R. Sebastião and J. Gama. Monitoring Incremental Histogram Distribution for Change Detection in Data Streams. In Ranga Raju Vatsavai, Olufemi Omitaomu, João Gama, Nitesh V. Chawla, Mohamed Medhat Gaber and Auroop Ganguly, editors. Lecture Notes in Computer Science (LNCS). Springer Verlag (to appear).
3. J. Gama, P. Medas, G. Castillo and P. P. Rodrigues. Learning with Drift Detection. *Advances in Artificial Intelligence - SBIA 2004*, Vol.3171 of Lecture Notes in Computer Science, pages:286-295, So Luiz, Maranhão, Brazil, October 2004. Springer Verlag.
4. G. Hulten, L. Spencer and P. Domingos. Mining Time-Changing Data Streams. ACM SIGKDD 2001. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages:97-106, 2001. ACM Press.
5. D. Kifer, S. Ben-David and J.Gehrke. Detecting change in data streams. In *VLDB 04: Proceedings of the 30th International Conference on Very Large Data Bases*, pages:180-191, 2004. Morgan Kaufmann Publishers Inc.
6. G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23:69-101, 1996.
7. R. Klinkenberg and I. Renz. Adaptive information filtering: Learning in the presence of concept drifts. In *Learning for Text Categorization*, pages:33-40, 1998. AAAI Press.
8. R. Klinkenberg and T. Joachims. Detecting concept drift with support vector machines. *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 487–494, Stanford, US, 2000. Morgan Kaufmann Publishers.
9. R. Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 8(3):281-300, 2004.
10. M. Maloof and R. Michalski. Selecting examples for partial memory learning. *Machine Learning*, 41:27–52, 2000.
11. A. Bifet and R. Gavaldà. Learning from Time-Changing Data with Adaptive Windowing. *Proceedings of the 7th SIAM International Conference on Data Mining*, Minneapolis, Minnesota, USA, 2007.
12. J. Gama and C. Pinto. Discretization from Data Streams: applications to Histograms and Data Mining. *Proceedings of the 2006 ACM Symposium on Applied Computing*, pages:662-667, 2006.
13. E. S. Page. Continuous Inspection Schemes. *Biometrika*, Vol. 41:100-115, 1954.
14. H. Mouss, D. Mouss, N. Mouss and L. Sefouhi. Test of Page-Hinkley, an Approach for Fault Detection in an Agro-Alimentary Production System. *Proceedings of the 5th Asian Control Conference*, Vol.2, pages: 815-818, 2004.
15. M. Kubat and G. Widmer. Adapting to drift in continuous domain. *Proceedings of the 8th European Conference on Machine Learning*, pages 307-310. Springer Verlag, 1995.
16. W. N. Street and Y. Kim. A streaming ensemble algorithm SEA for large-scale classification. *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages: 377-382, 2001. ACM Press.

Chapter 9

MASTA - Multi-Agent Systems: Theory and Applications

A Formal Notion of Objective Expectations in the Context of Multiagent Systems Routines

A.C.R. Costa[1], G.P. Dimuro[1], J. Dugdale[2], and Y. Demazeau[3]

[1] PPGINF, Universidade Católica de Pelotas, 96010-000 Pelotas, Brazil
[2] Laboratoire d'Informatique de Grenoble - Université Pierre-Mendes France
[3] Laboratoire d'Informatique de Grenoble - CNRS, BP 53 - 38041 Grenoble, France
{rocha,liz}@ucpel.tche.br,{Yves.Demazeau,Julie.Dugdale}@imag.fr

Abstract. This paper introduces an objective notion of *routine expectation*, to allow for the external account of expectations in the context of routine procedures, in multiagent systems (MAS). The notion of expectation as usually applied to MAS is briefly reviewed. A formalization of *routine procedure* is given, so that the formal notion of *routine expectation* can be defined with respect to actions and facts. A view previously proposed in the literature, to base expectation values on a combination of probability values and utility values, is adopted. However, it is adapted to the context of repetitive, periodic system routines, where utility values can be replaced by the *degrees of perfection* with which actions and facts are realized.

1 Introduction: motivation and related works

In general there are different ways of considering the notion of expectation in agent systems. For example, a *mentalist*, or *cognitivist* approach defines an expectation as a complex set of cognitive elements (a combination of goals, beliefs, etc.) [1]. Such kind of *subjectivist* approach may also take the way adopted in [2, 3], briefly suggested in [1], which gives expectations a combination of *probability values* and *utility values*, assigned to the *possible outcomes* of the actions (using the notion of *subjective expected utility values* [4]).

A crucial aspect of both these works is that an agent's expectation is related to a momentary situation in the system. That is, it relates to a situation that is artificially isolated from the whole behavior of the agent, and from the continuous operation of the system. The agent is seen as having a goal and performing an action at a given instant (or believing that a given event will happen that will help to achieve the goal), so that the entire past and future ways in which the agent participates in the system is not taken into account. Expectations about future agent interactions can not be adequately studied in such settings.

However, most practically important agent systems, including human ones, are important precisely because of their *continuous* functioning over time and, in particular, because of their repetitive, *periodic* functioning; that is, their *routines*. Routines have long been recognized as an essential feature of the operational dynamics of social systems (cf., e.g., Giddens in [5]), and the same idea seems to apply to multiagent systems.

Here the functioning aims at the continuous monitoring and/or control of some target system, or the continuous servicing of their users [6].

An example of the importance of periodic routines and expected patterns of behavior in managing electricity in home situations was seen in [7]. Also, modeling expectations seems very promising in the idea of developing personal assistants that will support a human user in his/her usual day-life activities [8].

We thus develop a *complementary* point of view. We formulate in set-theoretic terms an *objective*, *behavioristic* notion of expectation, centered around the *objective effects* of the actions that the agents perform, at a given moment, within the context of some *system routine*. We do not consider here, however, the possible ways in which the two approaches may be combined. Specially, we do not consider how the context of system routines can be integrated into the mentalist approach.

In this paper, we introduce two novel aspects concerning agent expectations. First, we extend the notion of expectation so that, in the context of system routines, it may be applied not only to the facts resulting from the execution of agent actions or environment events, but also to the actions and events themselves. In fact, we consider that *objective expectations of actions and events* are fundamental, and that *objective expectations of facts* should be derived from them. Accordingly, in Sect. 2 we formalize a notion of system routine where those two kinds of expectations may be defined.

Secondly, we would like to stay as close as possible to the *subjective expected utility values* approach to expectations. However, utility values, as such, can not be fruitfully considered in connection to actions and facts that happen in system routines. This is because routines are usually adopted in a system only when all the actions are clearly useful to the routines where they occur. That is, less useful actions tend to be replaced by actions more adapted, and thus more useful, to the routine and to the functioning of the system as a whole. If an action is included in a routine, or if a fact is counted upon by the routine to operate properly, it is clear that it certainly has a utility for the routine, and there is no point in trying to quantify that value. Therefore, in order that the *objective expected utility* of an action (or fact or event) could stay close to the model of *subjective expected utility*, it was necessary to replace the subjective components by corresponding objective components. Specially, the *subjective probabilities* and *utility values* that essentially constitute the subjective expected utility values were respectively replaced by *objective probabilities* and by *objective degrees of perfection* of the realization of actions, facts and events. This is explained in Sect. 3 of the paper.

A contextualized example of the approach is given in Sect. 4. Section 5 concludes the paper with a discussion.

2 Periodic Routines and their Performances

In the following, let Act be a finite set of agent actions and Ctx be the set of contexts where those actions may be performed, with the empty action denoted by ε. Let $T = t_0, t_1, \ldots$ be a linear, discrete time structure. The effects of the performance of an action $\alpha \in Act$, in a context $C \in Ctx$, at a certain time $t \in T$, may be defined in a rule-based form with the help of pre- and post-conditions, which can assume *degrees of perfection* varying in the interval $[0, 1]$, as the degrees of truth in Fuzzy Logic [11], as follows:

$$\frac{pre_1 : d_1 \ldots pre_n : d_n}{post_1 : f_1(d_1, \ldots, d_n) : min_1 \ldots post_m : f_m(d_1, \ldots, d_n) : min_m} [\alpha : C] \quad (1)$$

where $\mathrm{pre}[\alpha : C] = \{pre_1, \ldots, pre_n\}$, with $n \in \mathbb{N}$, is the set of pre-conditions to be evaluated at any time the action α is ready to be performed in the context C, and $d_1, \ldots, d_n$ are their respective degrees of perfection at that moment; $\mathrm{post}[\alpha : C] = \{post_1, \ldots, post_m\}$, with $m \in \mathbb{N}$, is the set of post-conditions that result from the performance of α in the context C, and $f_1(d_1, \ldots, d_n), \ldots, f_m(d_1, \ldots, d_n)$ are their respective *expected degrees of perfection* after that performance, determined by the functions $f_1, \ldots, f_m : [0,1]^n \to [0,1]$ that give the expected values of the post-conditions considering the degrees of perfection of all pre-conditions; $min_1, \ldots, min_n$, with $0 < min_i \leq 1$, are the minimum degrees of perfection required for the post-conditions in order that the performance of the action α be considered satisfactory. In particular, it holds that $d_i = 1$, for any $pre_i \in \mathrm{pre}[\varepsilon : C]$, and $f_j(d_1, \ldots, d_n) = min_j = 1$, for any $post_j \in \mathrm{post}[\varepsilon : C]$, in any context C.

An action α is said to be *satisfactorily performed* in the context C, at time $t \in T$, if and only if each post-condition assumes, at time $t+1$, a degree of perfection equal or greater than the minimum degree of perfection required for that post-condition. In particular, a satisfactorily performed action α is said to be *perfectly performed* in the context C, at time t, if and only if all post-conditions assume, at time $t+1$, a degree of perfection equal to 1. An action α is said to be *non-satisfactorily performed* in the context C, at time t, if and only if at least one of the post-conditions assumes, at time $t+1$, a degree of perfection less than the minimum required for that post-condition. A non-satisfactorily performed action α is said to be *non-performed* at time t, if and only if all post-conditions assume, at time $t+1$, a degree of perfection equal to 0.

Definition 1. *Let $\alpha \in Act$ be an action performed in the context C at time t. The* degree of satisfaction *of α's performance is given by the function $ds^t_{\alpha:C} : [0,1]^m \to \{s, ns\} \times [0,1]$. This evaluates the degrees of perfection $u_1, \ldots, u_m \in [0,1]$ assumed, at time $t+1$, by the post-conditions $post_1, \ldots, post_m$ of the action α. It generates a pair $ds^t_{\alpha:C}(u_1, \ldots, u_m) = (x, k) \in (\{s, ns\} \times [0,1])$, where:*

(i) *$k \in [0,1]$ is the* degree of perfection*, obtained by an operator $\tau : [0,1]^m \to [0,1]$, which is application dependent;*

(ii) *$x = s$ indicates that the action α was satisfactorily performed (so that $k > 0$; in particular, if $k = 1$ then the action α was perfectly performed);*

(iii) *$x = ns$ indicates that the action α was non-satisfactorily performed (so that $k < 1$; in particular, if $k = 0$ then α was non-performed);*

A *routine* over Act, in a context $C \in Ctx$, with period π, where $\pi \in \mathbb{N}$ and $\pi > 1$, is a permanently (i.e., infinitely) repeated, structured subset of actions of Act in C. The main repeated structured set of actions in a routine ρ is the *body* of ρ. The *set of all possible routine bodies RB_{Act}* is given by a set of regular expressions over Act:

- any action $\alpha \in Act$ is a routine body: $\alpha \in Act \Rightarrow \alpha \in RB_{Act}$
- if b and b' are routine bodies ($b, b' \in RB_{Act}$), then the following regular compositions of b and b' are also routine bodies:
  - $b; b' \in RB_{Act}$ (sequential composition of the routine bodies b and b');
  - $b + b' \in RB_{Act}$ (alternative composition of the routine bodies b and b');

Table 1. Recursive definitions of length and set of actions of routine bodies, for $b, b' \in RB_{Act}$

Length	Set of actions
$length(b) = 1$, if $b = \alpha \in Act$	$\mathrm{act}[b] = \{\alpha\}$, if $b = \alpha \in Act$
$length(b; b') = length(b) + length(b')$	$\mathrm{act}[b; b'] = \mathrm{act}[b] \cup \mathrm{act}[b']$
$length(b + b') = \max\{length(b), length(b')\}$	$\mathrm{act}[b + b'] = \mathrm{act}[b] \cup \mathrm{act}[b']$
$length(b \mid b') = length(b) + length(b')$	$\mathrm{act}[b \mid b'] = \mathrm{act}[b] \cup \mathrm{act}[b']$
$length(b^n) = n \cdot length(b)$	$\mathrm{act}[b^n] = \mathrm{act}[b]$

- $b \mid b' \in RB_{Act}$ (parallel composition of the routine bodies b and b');[4]
- $b^n \in RB_{Act}$ (n-times repetition of the routine body b, with $n \in \mathbb{N}$ and $n > 0$);

The $length$ of a routine body is defined as shown in the left-hand column of Table 1. The length of a routine's body is the *period* of that routine (note the case of the parallel composition, reserving a time slot for its eventual sequential realization, with one alternative fully performed before the other). So, any routine ρ, with period π, is such that $\rho = b^\omega$, where $b \in RB_{Act}$, $\pi = length(b)$, and ω indicates infinite repetition of the body b. The set of actions of the routine $\rho = b^\omega$, denoted by $\mathrm{act}[\rho]$, is the same as the set of actions of its body, as defined in the right-hand column of Table 1.

A *cut* of a routine ρ is any non-empty subset of actions in Act that the routine may simultaneously perform at a given time instant. The set of cuts of a routine $\rho = b^\omega$ is denoted by $\mathrm{Cuts}[\rho]$ and is the same as the set of cuts of its body $\mathrm{Cuts}[b]$, which is defined recursively as follows, for routine bodies $b, b' \in RB_{Act}$:

$$\begin{aligned}
\mathrm{Cuts}[b] &= \{\{\alpha\}\}, \text{ if } b = \alpha \in Acts \\
\mathrm{Cuts}[b; b'] &= \mathrm{Cuts}[b] \cup \mathrm{Cuts}[b'] \\
\mathrm{Cuts}[b + b'] &= \mathrm{Cuts}[b] \cup \mathrm{Cuts}[b'] \\
\mathrm{Cuts}[b \mid b'] &= \mathrm{Cuts}[b] \cup \mathrm{Cuts}[b'] \cup \{X \cup Y \mid X \in \mathrm{Cuts}[b], Y \in \mathrm{Cuts}[b']\} \cup \{\{\varepsilon\}\} \\
\mathrm{Cuts}[b^n] &= \mathrm{Cuts}[b]
\end{aligned}$$

The notion of a cut of a routine is central to our concept of routine expectation. It was drawn from the notion of cut in *occurrence nets* from Net Theory [12].

Definition 2. *Let ρ be a routine over Act and $\mathrm{Cuts}[\rho]$ be the set of cuts of ρ. A* performance *of a routine ρ in the context C is a periodic mapping*

$$\ll\rho : C\gg : T \to \wp(\mathrm{Cuts}[\rho]) \times \mathrm{Cuts}[\rho] \times \wp(Act) \times \wp(Act \times [0, 1]),$$

which gives, for each time $t \in T$, a 4-tuple where:

(i) *the first component $\ll\rho : C\gg^t[1]$ is the set of cuts* enabled *by the performance $\ll\rho : C\gg$ at time t;*
(ii) *the second component $\ll\rho : C\gg^t[2]$ is the cut* selected *to be performed at time t in $\ll\rho : C\gg$, among the enabled cuts at that time;*[5]
(iii) *the third component $\ll\rho : C\gg^t[3]$ is the* step *of the performance $\ll\rho : C\gg$ at time t, which is the subset of actions of the selected cut $\ll\rho : C\gg^t[2]$ that are put into execution by the system's agent(s) at time t;*

[4] Observe that the empty action ε is the neutral element of the parallel composition operator, that is, $\varepsilon \mid b = b \mid \varepsilon = b$, for any routine body b.

[5] A routine may enable more than one cut at a given time. However, only one cut must be selected to be performed at each time.

(iv) *the fourth component* $\ll \rho : C \gg^t [4]$ *is the* result *of the step* $\ll \rho : C \gg^t$ [3], *indicating the degree of satisfaction of the performance of each action of that step.*

The *period* of the performance $\ll\rho:C\gg$ is given by the period of ρ. The set of all possible performances $\ll\rho:C\gg$ of ρ in the context C is denoted by $\mathrm{Prfs}[\rho : C]$. The four components of a performance are accessed through the following functions:

Definition 3. *Consider a performance* $\ll\rho:C\gg \in \mathrm{Prfs}[\rho:C]$. *Then* [6]*:*

(i) *The* cuts enabled by the performance $\ll\rho:C\gg$ *in each time* t *are specified by a mapping* $\mathrm{ecuts} : \mathrm{Prfs}[\rho:C] \to T \to \wp(\mathrm{Cuts}[\rho])$*, such that the cuts enabled by a performance* $\ll\rho:C\gg$*, at a given time* $t \in T$*, is denoted by* $\mathrm{ecuts}[\ll\rho:C\gg]^t$ *and* $\mathrm{ecuts}[\ll\rho:C\gg]^t = \ll\rho:C\gg^t [1]$. *The* trace of subsets of enabled cuts *by* $\ll\rho:C\gg$ *in* T *is defined as the time-indexed sequence:* $\mathrm{trace}[\mathrm{ecuts}[\ll\rho:C\gg]]^T = \mathrm{ecuts}[\ll\rho:C\gg]^{t_0}, \mathrm{ecuts}[\ll\rho:C\gg]^{t_1}, \ldots$.

(ii) *The cuts of a routine* ρ *that are* selected *to be performed during the performance* $\ll\rho:C\gg$ *are selected among the enabled cuts in each time* t*; they are specified by a mapping* $\mathrm{scut} : \mathrm{Prfs}[\rho:C] \to T \to \mathrm{Cuts}[\rho]$*, such that, given the set of enabled cuts* $\mathrm{ecuts}[\ll\rho:C\gg]^t$*, the cut selected by* $\ll\rho:C\gg$ *to be executed at a given time* t *is denoted by* $\mathrm{scut}[\ll\rho:C\gg]^t$ *and* $\mathrm{scut}[\ll\rho:C\gg]^t = \ll\rho:C\gg^t [2] \in \mathrm{ecuts}[\ll\rho:C\gg]^t$. *The* trace of selected cuts *by* $\ll\rho:C\gg$ *in* T *is defined as the time-indexed sequence:* $\mathrm{trace}[\mathrm{scut}[\ll\rho:C\gg]]^T = \mathrm{scut}[\ll\rho:C\gg]^{t_0}, \mathrm{scut}[\ll\rho:C\gg]^{t_1}, \ldots$.

(iii) *The* steps *of the performance* $\ll\rho:C\gg$ *are specified by a mapping* $\mathrm{step} : \mathrm{Prfs}[\rho:C] \to T \to \wp(Act)$*, such that, given the selected cut* $\mathrm{scut}[\ll\rho:C\gg]^t$*, the subset of actions that are executed by the agent at the time* t *is denoted by* $\mathrm{step}[\ll\rho:C\gg]^t$ *and* $\mathrm{step}[\ll\rho:C\gg]^t = \ll\rho:C\gg^t [3] \subseteq \mathrm{scut}[\ll\rho:C\gg]^t$. *The* trace of steps *of the performance* $\ll\rho:C\gg$ *in* T *is defined as the time-indexed sequence:* $\mathrm{trace}[\mathrm{step}[\ll\rho:C\gg]]^T = \mathrm{step}[\ll\rho:C\gg]^{t_0}, \mathrm{step}[\ll\rho:C\gg]^{t_1}, \ldots$.

(iv) *The* results of the steps *of the performance* $\ll\rho:C\gg$ *at time* t *is given by a mapping* $\mathrm{result} : \mathrm{Prfs}[\rho:C] \to T \to \wp(Act \times [0,1])$ *such that, given the step* $\mathrm{step}[\ll\rho:C\gg]^t$ *of the performance* $\ll\rho:C\gg$ *at time* $t \in T$*, the set of results of that step is denoted by* $\mathrm{result}[\ll\rho:C\gg]^t$ *and* $\mathrm{result}[\ll\rho:C\gg]^t = \ll\rho:C\gg^t [4] = \{(\alpha, ds^t_{\alpha:C}(u_1,\ldots,u_m)) \mid \alpha \in \mathrm{step}[\ll\rho:C\gg]^t\}$*, where* $ds^t_{\alpha:C}(u_1,\ldots,u_m)$ *is the degree satisfaction of the performance of* α*, obtained by the evaluation of the degrees of perfection* $u_1,\ldots,u_m$ *assumed by* α*'s post-condition* $post_1,\ldots,post_m$ *at time* $t+1$ *(see Def. 1). The trace of results of steps of the performance* $\ll\rho:C\gg$ *in* T *is given by the sequence:*

$$\mathrm{trace}[\mathrm{result}[\ll\rho:C\gg]]^T = \mathrm{result}[\ll\rho:C\gg]^{t_0}, \mathrm{result}[\ll\rho:C\gg]^{t_1}, \ldots$$

The set of possible steps of a cut c is the set of its subsets $\wp(c)$. Given a family of cuts X, the set of possible steps derived from X is given by $\mathrm{psteps}[X] = \bigcup_{c\in X} \wp[c]$.

The effects of a cut $c \in \mathrm{Cuts}[\rho]$, in a context C, may be defined in a rule-based form with the help of pre- and post-conditions, analogously to rule (1), where the set of pre-conditions is the union of the sets of pre-conditions of all the actions that constitute the cut c, given by $\mathrm{pre}[c] = \bigcup\{\mathrm{pre}[\alpha] \mid \alpha \in c\}$. Similarly, the set of post-conditions of a cut $c \in \mathrm{Cuts}[\rho]$ is given by $\mathrm{post}[c] = \bigcup\{\mathrm{post}[\alpha] \mid \alpha \in c\}$.

[6] For simplicity, in this paper, given any function $f : T \to X$, where T is the time sequence, we denote $f(t)$ by f^t.

Considering a performance $\ll\rho:C\gg\in \mathrm{Prfs}[\rho:C]$ in a context $C\in Ctx$, a selected cut $c\in \mathrm{ecuts}[\ll\rho:C\gg]^t$ is said to be *satisfactorily (perfectly) performed* at time t if and only if all its actions are satisfactorily (perfectly) performed at that time. On the contrary, a selected cut $c\in \mathrm{ecuts}[\ll\rho:C\gg]^t$ is *non-satisfactorily performed* at time t if and only if at least one of its actions is non-satisfactorily performed at time t. A non-satisfactorily performed selected cut c is said to be *non-performed* at time t if and only all its actions are non-performed at time t.

Given a selected cut $c\in \mathrm{ecuts}[\ll\rho:C\gg]^t$, the set of pre-conditions of a step $s\subseteq c$ is the union of the set of pre-conditions of the actions that constitute that step, that is, $\mathrm{pre}[s]=\bigcup\{\mathrm{pre}[\alpha]\mid\alpha\in s\}\subseteq\mathrm{pre}[c]$. Analogously, the set of post-conditions of a step $s\subseteq c$ is given by $\mathrm{post}[s]=\bigcup\{\mathrm{post}[\alpha]\mid\alpha\in s\}\subseteq\mathrm{post}[c]$. A step $s\subseteq c$ can either be satisfactorily (perfectly, non-satisfactorily) performed or non-performed, in the same way as was defined for selected cuts.

Therefore, a selected cut $\mathrm{scut}[\ll\rho:C\gg]^t$ of a routine performance $\ll\rho:C\gg$ at time t is *satisfactorily* (or *perfectly*) *performed* if and only if $\mathrm{step}[\ll\rho:C\gg]^t=\mathrm{scut}[\ll\rho:C\gg]^t$ and $\mathrm{step}[\ll\rho:C\gg]^t$ is satisfactorily (or perfectly) performed. It is *non-satisfactorily performed* if either $\mathrm{step}[\ll\rho:C\gg]^t\neq\mathrm{scut}[\ll\rho:C\gg]^t$ or $\mathrm{step}[\ll\rho:C\gg]^t$ is non-satisfactorily performed. It is *non-performed* whenever either $\mathrm{step}[\ll\rho:C\gg]^t=\emptyset$ or $\mathrm{step}[\ll\rho:C\gg]^t$ is non-performed.[7]

Definition 4. *Consider a routine ρ over Act, in a given context $C\in Ctx$, with period π, whose performance $\ll\rho:C\gg$ is done in a time sequence $T=t_0,t_1,\ldots$. Then:*

- **(i)** *The performance $\ll\rho:C\gg$ is said to be* satisfactorily *(*perfectly, non-satisfactorily*) performed at time t if and only if* $\mathrm{scut}[\ll\rho:C\gg]^t$ *is satisfactorily (perfectly, non-satisfactorily) performed at time t.*
- **(ii)** *The performance $\ll\rho:C\gg$ fails at time t if* $\mathrm{scut}[\ll\rho:C\gg]^t$ *is non-performed at time t.*
- **(iii)** *The performance $\ll\rho:C\gg$ is said to be* satisfactorily (perfectly) performed in a cycle *that starts at time $t_i\in T$, if it is satisfactorily (perfectly) performed at all $t\in[t_i,t_{i+\pi-1}]$.*
- **(iv)** *The performance $\ll\rho:C\gg$ is said to be* non-satisfactorily performed in a cycle *that starts at time $t_i\in T$ if there is at least one $t\in[t_i,t_{i+\pi-1}]$ at which it fails or is non-satisfactorily performed.*
- **(v)** *The performance $\ll\rho:C\gg$ fails in a cycle that starts at time $t_i\in T$, if it fails at all $t\in[t_i,t_{i+\pi-1}]$.*

Example 1. Consider $Act=\{\alpha_1,\alpha_2,\alpha_3\}$ and let $\rho=(\alpha_1;(\alpha_2+\alpha_3))^\omega$ be a routine with period equal to 2. The possible cuts in any performances of this routine are $\mathrm{Cuts}[\rho:C]=\{c_1,c_2,c_3\}$, where $c_1=\{\alpha_1\}$, $c_2=\{\alpha_2\}$ and $c_3=\{\alpha_3\}$. The cuts enabled during a *perfectly performed* performance $\ll\rho:C\gg$ in the time interval $[t_0,t_2)$ (just one cycle) are given by: $\mathrm{ecuts}[\ll\rho:C\gg]^{t_0}=\{c_1\},\mathrm{ecuts}[\ll\rho:C\gg]^{t_1}=\{c_2,c_3\}$.

Then, there are only two possibilities of traces of a perfectly performed performance $\ll\rho:C\gg$ in that interval:

[7] Observe that the enablement of a subset of cuts of a routine ρ at time $t+1$ of a performance $\ll\rho:C\gg$ may depend on the step of the performance of the routine ρ at time t, which indicates if the cut selected at time t was or was not satisfactorily performed.

(i) either the selected cuts are $\mathrm{scut}[\ll \rho : C \gg] = t_0 \mapsto c_1, t_1 \mapsto c_3$, and the traces of steps and results of steps are, respectively:

$$\mathrm{trace}[\mathrm{step}[\ll \rho : C \gg]]^{[t_0,t_2)} = t_0 \mapsto c_1, t_1 \mapsto c_2$$
$$\mathrm{trace}[\mathrm{result}[\ll \rho : C \gg]]^{[t_0,t_2)} = t_0 \mapsto \{(\alpha_1,(s,1))\}, t_1 \mapsto \{(\alpha_2,(s,1))\}$$

(ii) or, the selected cuts are $\mathrm{scut}[\ll \rho : C \gg] = t_0 \mapsto c_1, t_1 \mapsto c_2$, and the traces of steps and results of steps are, respectively:

$$\mathrm{trace}[\mathrm{step}[\ll \rho : C \gg]]^{[t_0,t_2)} = t_0 \mapsto c_1, t_1 \mapsto c_3,$$
$$\mathrm{trace}[\mathrm{result}[\ll \rho : C \gg]]^{[t_0,t_2)} = t_0 \mapsto \{(\alpha_1,(s,1))\}, t_1 \mapsto \{(\alpha_3,(s,1))\},$$

where each step in each time, which coincides with the cut selected at that time, is perfectly performed. Consider that the pre-conditions of both actions α_2 and α_3 are the post-conditions generated by the performance of the action α_1, and the minimum degrees of truth required for the post-conditions of both α_2 and α_3 are equal to 1. Then, supposing that, in both cases (i) and (ii) above, one has that $\mathrm{result}[\ll \rho : C \gg]^{t_0} = \{(\alpha_1,(s,k_1 < 1))\}$, then, probably, one would have $\mathrm{result}[\ll \rho : C \gg]^{t_0} = \{(\alpha_2,(ns,k_2 < 1))\}$ (in case (i)) or $\mathrm{result}[\ll \rho : C \gg]^{t_0} = \{(\alpha_3,(ns,k_3 < 1))\}$ (case (ii)), characterizing examples of $\ll \rho : C \gg$ that are non-satisfactorily performed (failed performances whenever $k_1 = k_2 = k_3 = 0$).

Example 2. Consider $Act = \{\alpha_1, \alpha_2, \alpha_3\}$ and let $\rho = ((\alpha_1 + \alpha_2)^3 \mid \alpha_3)^\omega$ be a routine with period equal to 4. The possible cuts in the performance of this routine are $\mathrm{Cuts}[\rho : C] = \{c_0, c_1, c_2, c_3, c_4, c_5\}$, where: $c_0 = \{\varepsilon\}$, $c_1 = \{\alpha_1\}$, $c_2 = \{\alpha_2\}$, $c_3 = \{\alpha_3\}$, $c_4 = \{\alpha_1, \alpha_3\}$ and $c_5 = \{\alpha_2, \alpha_3\}$. Table 2 shows the cuts enabled during a performance $\ll \rho : C \gg$ in 5 cycles, the selected cuts in each time t and the trace of steps of the performance, which was perfectly performed in the first cycle, but failed in the second cycle, since the cut at time t_6 failed. The third and fourth cycles $\ll \rho : C \gg$ were just non-satisfactorily performed, since the cuts at times t_8 and t_{12} were non-satisfactorily performed. At time t_8, the action α_1 of the selected cut at that time was executed in the respective performance step, with a performance satisfactoriness degree of 0.5, but in a non-satisfactory way (since at least one of its post-conditions did not have the required minimum degree of truth). At time t_{12}, the action α_3 of the selected cut at that time was not even executed in the respective performance step. Finally, the fifth cycle failed, since the step at time t_{19} was empty.

3 Expectations

We consider two kinds of expectations: (i) *expectations of actions*, that is, that some set of actions happen in the performance of a given routine at a given time, and (ii) *expectations of facts*, which are expectations that some facts become true, with a certain degree, as a consequence of some set of actions occurring in a given routine at a given time. For example, in calendar systems [8], expectations of actions typically occur when a user requests that the personal assistants perform some actions for him or her. Alternatively, expectations of facts occur when personal assistants interact together in order to fulfil a global goal that is given by the user they support.

Table 2. Sample performances of the routine $\rho = ((\alpha_1 + \alpha_2)^3 \mid \alpha_3)^\omega$

Time	Enabled Cuts	Selected Cut	Step	Results
t_0	$\{c_0, c_1, c_2, c_3, c_4, c_5\}$	c_1	c_1	$\{(\alpha_1, (s, 1))\}$
t_1	$\{c_0, c_1, c_2, c_3, c_4, c_5\}$	c_1	c_1	$\{(\alpha_1, (s, 1))\}$
t_2	$\{c_0, c_1, c_2, c_3, c_4, c_5\}$	c_5	c_5	$\{(\alpha_2, (s, 1)), (\alpha_3, (s, 1))\}$
t_3	$\{c_0\}$	c_0	c_0	$\{(\varepsilon, (s, 1))\}$
t_4	$\{c_0, c_1, c_2, c_3, c_4, c_5\}$	c_5	c_5	$\{(\alpha_2, (s, 1)), (\alpha_3, (s, 1))\}$
t_5	$\{c_0, c_1, c_2\}$	c_2	c_2	$\{(\alpha_2, (s, 1))\}$
t_6	$\{c_0, c_1, c_2\}$	c_1	c_1	$\{(\alpha_1, (ns, 0))\}$
t_7	$\{c_0\}$	c_0	c_0	$\{(\varepsilon, (s, 1))\}$
t_8	$\{c_0, c_1, c_2, c_3, c_4, c_5\}$	c_4	c_4	$\{(\alpha_1, (ns, 0.7)), (\alpha_3, (s, 1))\}$
t_9	$\{c_0, c_1, c_2\}$	c_1	c_1	$\{(\alpha_1, (s, 1))\}$
t_{10}	$\{c_0, c_1, c_2\}$	c_2	c_2	$\{(\alpha_2, (s, 1))\}$
t_{11}	$\{c_0\}$	c_0	c_0	$\{(\varepsilon, (s, 1))\}$
t_{12}	$\{c_0, c_1, c_2, c_3, c_4, c_5\}$	c_4	$\{\alpha_1\}$	$\{(\alpha_1, (s, 1))\}$
t_{13}	$\{c_0, c_1, c_2\}$	c_1	c_1	$\{(\alpha_1, (s, 1))\}$
t_{14}	$\{c_0, c_1, c_2\}$	c_2	c_2	$\{(\alpha_2, (s, 1))\}$
t_{15}	$\{c_0\}$	c_0	c_0	$\{(\varepsilon, (s, 1))\}$
t_{16}	$\{c_0, c_1, c_2, c_3, c_4, c_5\}$	c_1	c_1	$\{(\alpha_1, (s, 1))\}$
t_{17}	$\{c_0, c_1, c_2, c_3, c_4, c_5\}$	c_1	c_1	$\{(\alpha_1, (s, 1))\}$
t_{18}	$\{c_0, c_1, c_2, c_3, c_4, c_5\}$	c_2	c_2	$\{(\alpha_2, (s, 1))\}$
t_{19}	$\{c_3\}$	c_3	$\emptyset$	$\emptyset$
...	...	...	...	

Given the set $E = \text{ecuts}[\ll \rho : C \gg]^{t_i} \subseteq \text{Cuts}[\rho]$ of the cuts enabled by a performance $\ll \rho : C \gg$ at a certain time $t_i \in T$, it may be possible to have a cut-selection probability distribution over E, given by $\chi^{t_i}_{\ll\rho:C\gg} : E \to [0, 1]$, such that, for each $c \in E$, $\chi^{t_i}_{\ll\rho:C\gg}(c)$ gives the probability of the selection of c, considering the context $C \in Ctx$, such that $\sum_{c\in E} \chi^{t_i}_{\ll\rho:C\gg}(c) = 1$. Denote by ρ_χ the routine ρ with associated cut-selection probability functions $\chi^{t_i}_{\ll\rho:C\gg}$, where $i \in \mathbb{N}$.

Definition 5. *Given a performance $\ll \rho_\chi : C \gg$ of a routine ρ_χ in a context C, the* cut selection expectation *of $\ll \rho_\chi : C \gg$ at a certain time $t_i \in T$ is the subset of cuts of the cuts enabled at time t_i whose cut-selection probabilities are maximal in* $\text{ecuts}[\ll \rho_\chi : C \gg]^t \subseteq \text{Cuts}[\rho_\chi]$*, that is,*

$$\text{csxpt}[\ll \rho_\chi : C \gg]^{t_i} =$$
$$\{c \in \text{ecuts}[\ll \rho_\chi : C \gg]^{t_i} \mid \forall c' \in \text{ecuts}[\ll \rho_\chi : C \gg]^{t_i} : \chi^{t_i}_{\ll\rho:C\gg}(c') \leq \chi^{t_i}_{\ll\rho:C\gg}(c)\}$$

Given the cut $\text{scut}[\ll \rho : C \gg]^{t_i}$, selected at time t_i in a performance $\ll \rho : C \gg$, the probability that an action $\alpha \in \text{scut}[\ll \rho : C \gg]^{t_i}$ is executed in the step $\text{step}[\ll \rho : C \gg]^{t_i}$ by an agent, in the context $C \in Ctx$ and at t_i, is given by the function $\phi^{t_i}_{\text{scut}[\ll\rho:C\gg]^{t_i}} : \text{scut}[\ll \rho : C \gg]^{t_i} \to [0, 1]$. The probability of $\alpha \in \text{scut}[\ll \rho : C \gg]^{t_i}$ not being executed in the step $\text{step}[\ll \rho : C \gg]^{t_i}$ is $1 - \phi^{t_i}_{\text{scut}[\ll\rho C\gg]^{t_i}}(\alpha)$. Then, the probability of a step being executed in $\text{scut}[\ll \rho : C \gg^{t_i}]$ is given by the function $\Phi^{t_i}_{\text{scut}[\ll\rho:C\gg]^{t_i}} : \wp[\text{scut}[\ll \rho : C \gg]^{t_i}] \to [0, 1]$, defined by

$$\Phi^{t_i}_{\text{scut}[\ll\rho:C\gg]^{t_i}}(X) = \prod_{\alpha\in\text{scut}[\ll\rho:C\gg]^{t_i}} Pr(\alpha), \text{ with } Pr(\alpha) = \begin{cases} \phi^{t_i}_{\text{scut}[\ll\rho:C\gg]^{t_i}}(\alpha) & \text{if } \alpha \in X \\ 1 - \phi^{t_i}_{\text{scut}[\ll\rho:C\gg]^{t_i}}(\alpha) & \text{otherwise.} \end{cases} \tag{2}$$

Denote by ρ_Φ the routine ρ with associated step probability functions $\Phi^{t_i}_{\text{scut}[\ll\rho:C\gg]^{t_i}}$ for each selected cut, with $i \in \mathbb{N}$.

Definition 6. *Given a performance* $\ll \rho_\Phi : C \gg$*, the* step expectation *in a cut* $\mathrm{scut}[\ll \rho_\Phi : C \gg]^{t_i}$ *selected at time* t_i*, are the steps of the cut* $\mathrm{scut}[\ll \rho_\Phi : C \gg]^{t_i}$ *whose probabilities of being executed at* t_i *are maximal in* $\wp[\mathrm{scut}[\ll \rho_\Phi : C \gg]^{t_i}]$*:*

$$\mathrm{stxpt}[\mathrm{scut}[\ll \rho_\Phi : C \gg]]^{t_i} = \{X \in \wp[\mathrm{scut}[\ll \rho_\Phi : C \gg]^{t_i}] \mid \forall X' \in \wp[\mathrm{scut}[\ll \rho_\Phi : C \gg]^{t_i}] : \Phi^{t_i}_{\mathrm{scut}[\ll \rho_\Phi : C \gg]^{t_i}}(X') \leq \Phi^{t_i}_{\mathrm{scut}[\ll \rho_\Phi : C \gg]^{t_i}}(X)\}.$$

The *overall step probability* of any possible step being executed in a performance $\ll \rho : C \gg$ at a given time $t_i \in T$ is given by a function $\sigma^{t_i}_{\ll \rho : C \gg} : \mathrm{psteps}[\mathrm{ecuts}[\ll \rho : C \gg]]^{t_1} \to [0, 1]$, defined by:

$$\sigma^{t_i}_{\ll \rho : C \gg}(X) = \sum_{c \in \mathrm{ecuts}[\ll \rho : C \gg]^{t_1}} \chi^{t_i}_{\ll \rho : C \gg}(c) \cdot Pr(X), \text{ with } Pr(X) = \begin{cases} \Phi^{t_i}_c(X) & \text{if } X \subseteq c \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where χ and Φ are the cut-selection probability functions and step probability functions, respectively. Denote by ρ_σ the routine ρ with associated *overall step probability* functions $\sigma^{t_i}_{\ll \rho : C \gg}$, with $i \in \mathbb{N}$.

Definition 7. *The* possible-step expectation *of a performance* $\ll \rho_\sigma : C \gg^{t_i}$*, at a certain time* $t_i \in T$*, is the subset of possible steps whose probabilities of being executed by the performance* $\ll \rho_\sigma : C \gg$ *of the routine* ρ_σ *at time* t_i *and context* C *are maximal in* $\mathrm{psteps}[\mathrm{ecuts}[\ll \rho_\sigma : C \gg]]^{t_1}$*, that is,*

$$\mathrm{psxpt}[\mathrm{ecuts}[\ll \rho_\sigma : C \gg]]^{t_1} = \{X \in \mathrm{psteps}[\mathrm{ecuts}[\ll \rho_\sigma : C \gg]]^{t_1} \mid \forall X' \in \mathrm{psteps}[\mathrm{ecuts}[\ll \rho_\sigma : C \gg]]^{t_1} : \sigma^{t_i}_{\ll \rho_\sigma : C \gg}(X') \leq \sigma^{t_i}_{\ll \rho_\sigma : C \gg}(X)\}.$$

Then, considering a routine with period π, the possible step expectation of a performance $\ll \rho_\sigma : C \gg$, in a period that starts at time t_i, is a sequence

$$\mathrm{psxpt}[\ll \rho_\sigma : C \gg]^\pi = \mathrm{psxpt}[\ll \rho_\sigma : C \gg]^{t_i}, \ldots, \mathrm{psxpt}[\ll \rho_\sigma : C \gg]^{t_i + \pi - 1}.$$

Definition 8. *The* expectation of the result of an action $\alpha \in Act$ *in a context* C*, at time* t_i*, is defined as*

$$\mathrm{rxpt}[\alpha : C]^{t_i} \equiv pre_1 : d_1, \ldots, pre_n : d_n \Rightarrow post_1 : f_1(d_1, \ldots, d_n), \ldots, post_m : f_m(d_1, \ldots, d_n), \tag{4}$$

where $f_1, \ldots, f_m : [0, 1]^n \to [0, 1]$ *are the functions giving the expected degrees of perfection of the post-conditions* $post_1, \ldots, post_m$ *of* α*, considering the degrees of perfection* $d_1, \ldots, d_n$ *of the pre-conditions* $pre_1, \ldots, pre_n$ *at time* t_i*.* [8]

The expectation of the result of a set of actions $X \subseteq Act$, which are performed at a certain time t_i in a context C, is defined as $\mathrm{rxpt}[X : C]^{t_i} = \{\mathrm{rxpt}[\alpha : C]^{t_i} \mid \alpha \in X\}$.

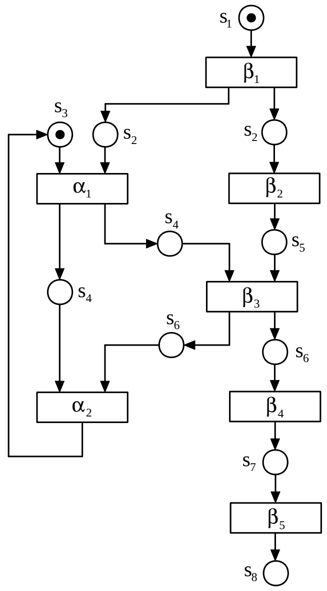
Definition 9. *Consider a possible-step expectation* $\mathrm{psxpt}[\ll \rho_\sigma : C \gg^{t_i}]$ *of a performance* $\ll \rho_\sigma : C \gg$*, at a certain time* $t_i \in T$*. The expectation of the results of a performance* $\ll \rho_\sigma : C \gg$ *at time* t_i*, is then defined as*

$$\mathrm{rxpt}[\ll \rho_\sigma : C \gg]^{t_i} = \{\mathrm{rxpt}[X : C]^{t_i} \mid X \in \mathrm{psxpt}[\ll \rho_\sigma : C \gg]^{t_i}\}.$$

The resulting expectation of $\ll p_{\rho_\sigma} : C \gg$, in a period π that starts at time t_i, is

$$\mathrm{rxpt}[\ll \rho_\sigma : C \gg]^\pi = \mathrm{rxpt}[\ll \rho_\sigma : C \gg]^{t_i}, \ldots, \mathrm{rxpt}[\ll \rho_\sigma : C \gg]^{t_i + \pi - 1}.$$

[8] The notion of expectation of the result of an action can be treated as a particular case of the notion of expectation adopted in [9, 10], if one includes the treatment of probabilities and degrees of perfection of realization of actions in periodic routines.



Table(a) Agent actions

Action	Agent: action specification
α_1	a: to open the door
α_2	a: to close the door
β_1	b: to enter into the scene
β_2	b: to approach the door
β_3	b: to pass through the open door
β_4	b: to withdraw from the door
β_5	b: to leave the scene

Table(b) Conditions

Notation	Condition
s_1	the scene started
s_2	the office employee is in the scene
s_3	the door is closed
s_4	the door is open
s_5	the office employee is near the door
s_6	the office employee is inside the building
s_7	the office employee withdrew from the door
s_8	the scene ended

Table(c) Action pre- and post-conditions

Action	Pre-conditions	Post-conditions
α_1	$\{s_2, s_3\}$	$\{s_4\}$
α_2	$\{s_4, s_6\}$	$\{s_3\}$
β_1	$\{s_1\}$	$\{s_2\}$
β_2	$\{s_2\}$	$\{s_5\}$
β_3	$\{s_4, s_5\}$	$\{s_6\}$
β_4	$\{s_6\}$	$\{s_7\}$
β_5	$\{s_7\}$	$\{s_8\}$

Fig. 1. The Petri net representation and action specifications of the routine ρ

4 A Contextualized Example

Consider the routine ρ_σ of a porter (agent a_1), who is in charge of opening and closing the front-door of a building, and a person (agent b) who needs to get inside the building to reach the office where he works everyday. Let the routine be represented in the Petri net shown in Fig. 1, where Table(a) specifies the agent actions involved in ρ_σ, Table (b) specifies the conditions that may occur in ρ_σ, and Table(c) specifies the pre- and post-conditions of those actions in terms of such situations.

The structure of the routine ρ_σ can be described as $\rho = (\beta_1; (\alpha_1 \mid \beta_2); \beta_3; (\alpha_2 \mid \beta_4; \beta_5))^\omega$. Routine ρ_σ has period $\pi = 7$ and overall step probability functions $\sigma^{t_i}_{\ll\rho:C\gg}$ (specified below). The possible cuts in any performance of ρ are $c_0 = \{\varepsilon\}$, $c_1 = \{\beta_1\}$, $c_2 = \{\alpha_1\}$, $c_3 = \{\beta_2\}$, $c_4 = \{\alpha_1, \beta_2\}$, $c_5 = \{\beta_3\}$, $c_6 = \{\alpha_2\}$, $c_7 = \{\beta_4\}$, $c_8 = \{\beta_5\}$, $c_9 = \{\alpha_2, \beta_4\}$ e $c_{10} = \{\alpha_2, \beta_5\}$.

Let $\chi^{t_i}_{\ll\rho:C\gg}$ be the cut-selection probability functions associated to ρ_σ and consider a particular performance $\ll \rho_\sigma : C \gg$ in a context C, where, at time t_1, the enabled cuts are $\mathrm{ecuts}[\ll\rho_\sigma : C\gg]^{t_1} = \{c_0, c_2, c_3, c_4\}$, and the cut-selection probabilities are $\chi^{t_1}_{\ll\rho_\sigma:C\gg}(c_0) = 0.05$, $\chi^{t_1}_{\ll\rho_\sigma:C\gg}(c_2) = 0.1$, $\chi^{t_1}_{\ll\rho_\sigma:C\gg}(c_3) = 0.15$ and $\chi^{t_1}_{\ll\rho_\sigma:C\gg}(c_4) = 0.7$. Then, the cut selection expectation of $\ll\rho_\sigma : C\gg$ is $\mathrm{csxpt}[\ll\rho_\sigma : C\gg]^{t_1} = \{c_4\}$, that is, the cut that is expected to be selected at time t_1 is $c_4 = \{\alpha_1, \beta_2\}$.

Consider that the cut that is selected at time t_1 fits that expectation, that is, $\mathrm{scut}[\ll \rho_\sigma : C \gg]^{t_1} = c_4$. Suppose that the porter (agent a) is very attentive, and usually opens the door (action α_1) when the office employee appears in the scene. However, the office employee (agent b) is easily distracted and sometimes stops before approaching the door (action β_2) in order to do an action that is not in the routine, such as to talk to someone or to answer the mobile phone. Suppose that the probabilities of the actions $\alpha_1, \beta_2 \in c_4$ being executed at the step at time t_1 are $\phi^{t_1}_{c_4}(\alpha_1) = 0.9$ and $\phi^{t_1}_{c_4}(\beta_2) = 0.7$, respectively. The possible steps of the selected cut c_4 are given by $\wp[c_4] = \{\emptyset, \{\alpha_1\}, \{\beta_2\}, \{\alpha_1, \beta_2\}\}$. Then, the probability of each possible step being executed at time t_1, according to Eq. (2), is:

$$\Phi^{t_1}_{c_4}(\emptyset) = (1 - \phi^{t_1}_{c_4}(\alpha_1)) \cdot (1 - \phi^{t_1}_{c_4}(\beta_2)) = 0.03, \Phi^{t_1}_{c_4}(\{\alpha_1\}) = \phi^{t_1}_{c_4}(\alpha_1) \cdot (1 - \phi^{t_1}_{c_4}(\beta_2)) = 0.27$$
$$\Phi^{t_1}_{c_4}(\{\beta_2\}) = (1 - \phi^{t_1}_{c_4}(\alpha_1)) \cdot \phi^{t_1}_{c_4}(\beta_2) = 0.07, \Phi^{t_1}_{c_4}(\{\alpha_1, \beta_2\}) = \phi^{t_1}_{c_4}(\alpha_1) \cdot \phi^{t_1}_{c_4}(\beta_2) = 0.63.$$

The step expectation related to the cut c_4 at time t_1 is $\mathrm{sxpt}[\ll\rho_\sigma : C\gg]^{t_1} = \{\{\alpha_1, \beta_2\}\}$.

The probability of the possible steps being executed in the performance $\ll\rho_\sigma : C\gg$ at time t_1 (Eq. (3)), given that the enabled cuts are $\mathrm{ecuts}[\ll\rho_\sigma\gg]^{t_1} = \{c_0, c_2, c_3, c_4\}$, are:

$$\sigma^{t_1}_{\ll\rho:C\gg}(\{\alpha_1, \beta_2\}) = \chi^{t_i}_{\ll\rho:C\gg}(c_4) \cdot \Phi^{t_1}_{c_4}(\{\alpha_1, \beta_2\}) = 0.7 \cdot 0.63 = 0.441$$
$$\sigma^{t_1}_{\ll\rho:C\gg}(\{\alpha_1\}) = \chi^{t_1}_{\ll\rho:C\gg}(c_2) \cdot \Phi^{t_1}_{c_2}(\{\alpha_1\}) + \chi^{t_1}_{\ll\rho:C\gg}(c_4) \cdot \Phi^{t_1}_{c_4}(\{\alpha_1\}) = 0.279$$
$$\sigma^{t_1}_{\ll\rho:C\gg}(\{\beta_2\}) = \chi^{t_1}_{\ll\rho:C\gg}(c_3) \cdot \Phi^{t_1}_{c_3}(\{\beta_2\}) + \chi^{t_1}_{\ll\rho:C\gg}(c_4) \cdot \Phi^{t_1}_{c_4}(\{\beta_2\}) = 0.154$$
$$\sigma^{t_1}_{\ll\rho:C\gg}(\emptyset) = \chi^{t_1}_{\ll\rho:C\gg}(c_0) \cdot \Phi^{t_1}_{c_0}(\emptyset) + \ldots + \chi^{t_1}_{\ll\rho:C\gg}(c_4) \cdot \Phi^{t_1}_{c_4}(\emptyset) = 0.126$$

Then, the possible-step expectation of the performance $\ll \rho_\sigma : C \gg$ at time t_1 is $\mathrm{psxpt}[\ll\rho_\sigma : C\gg]^{t_1} = \{\{\alpha_1, \beta_2\}\}$.

Now, consider that the possible step at time t_1 is indeed the step $\{\alpha_1, \beta_2\}$, and that the expectation of the performance result of the actions α_1 and β_2 in the context C are defined, respectively, as $\mathrm{rxpt}[\alpha_1 : C]^{t_1} \equiv s_2 : d_{s_2}, s_3 : d_{s_3} \Rightarrow s_4 : f_{s_4}(d_{s_2}, d_{s_3})$ and $\mathrm{rxpt}[\beta_2 : C]^{t_1} \equiv s_2 : d_{s_2} \Rightarrow s_5 : f_{s_5}(d_{s_2})$ where s_2, s_3, s_4 and s_5 are situations specified in Fig. 1(b), and $f_{s_4} : [0,1]^2 \to [0,1]$ and $f_{s_5} : [0,1] \to [0,1]$, defined by $f_{s_4}(x, y) = \min\{x, y\}$ and $f_{s_5}(x) = x$, are the expecting functions of actions α_1 and β_2, respectively, as specified in Eq. (4). Thus, the result expectation of the performance $\ll\rho : C\gg$, at time t_1, is $\mathrm{rxpt}[\ll\rho_\sigma : C\gg]^{t_1} = \{\{\mathrm{rxpt}[\alpha_1 : C]^{t_1}, \mathrm{rxpt}[\beta_1 : C]^{t_1}\}\}$.

Finally, consider the result expectation $\mathrm{rxpt}[\alpha_1 : C]^{t_1}$ and suppose that the door is half-open and that the porter is not sure that the employee is already at the scene, such that the degrees of perfection of α_1's pre-conditions are $d_{s_2} = 0.9$ and $d_{s_3} = 0.7$ at time t_1 in the context C. In such a situation, the result expectation of the performance of α_1 is that the door may not be opened correctly, since the degree of perfection of α_1's post-condition s_4 is expected to be $f_{s_4}(0.9, 0.7) = \min\{0.9, 0.7\} = 0.7$.

Notice, however, that the notion of routine is defined so that even if all pre-conditions are *totally* true (i.e., $d_{s_2} = d_{s_3} = 1$, so that $f_{s_4}(1, 1) = 1$) there is always a possibility that the door is not correctly opened after the performance of α_1. For instance, the action α_1 may not be performed correctly, or some condition (which happens only rarely and so is justifiably not included in the routine) occurs negatively affecting the result of the action.

5 Conclusion

This paper introduced an objective notion of expectation of actions and facts in the context of routine-oriented multiagent systems. Expectations were defined on behavioral terms, inspired by the notion of maximal expected utility, where utility values were substituted by degrees of perfection of actions (involving degrees of perfection of facts). The proposed notion of expectations was defined in order to ground the cognitive concept of expectation in the context of routine-oriented multiagent systems. In this way, notions that are essential for proactive agents, like surprise and disappointment, can be defined in such context. We hope to integrate these aspects in our future work. In terms of applications, such as modeling family behaviors in domestic settings [7], our proposed approach may help the cognitive modeling of expectations, since the subjective model should be compatible with the objective one. The same may happen for personal assistants in, e.g., calendar systems [8].

Acknowledgements. This work was done while ACRC was visiting lecturer at UJF, and GPD was an invited researcher at the LIG/MAGMA Group, in Grenoble, between January and February 2009. Support from CNPq and CAPES is also acknowledged. The authors thank M. Piunti for noting how the objective degrees of perfection could support a notion of utility value (related to routines, not to actions) for the system as a whole.

References

1. Castelfranchi, C.: Mind as an anticipatory device: For a theory of expectations. In Gregorio, M.D. et al., eds.: Brain, Vision, and Artificial Intelligence. Number 3704 in LNCS. Springer (2005) 258–276
2. Piunti, M., Castelfranchi, C., Falcone, R.: Expectations driven approach for situated, goal-directed agents. In Baldoni, M., et al., eds.: Proc. of 8th AI*IA/TABOO Joint Work. "From Objects to Agents": Agents and Industry, Genova, Seneca Edizioni Torino (2007) 104–111
3. Piunti, M., Goncalves, J., Martinho, C.: Modeling expectations for affective agents. In: Proceedings of the ACII-07 Doctoral Consortium. (2007)
4. Savage, L.J.: The Foundations of Statistics. Wiley, New York (1954)
5. Giddens, A.: The Constitution of Society: Outline of the Theory of Structuration. Polity Press, Cambridge (1984)
6. Costa, A.C.R., Dimuro, G.P.: A minimal dynamical organization model. In Dignum, V., ed.: Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models. IGI Global, Hershey (2009) 419–445
7. Dugdale, J., Salembier, P.: Modélisation située des activités individuelles et collectives pour la conception. Rapport de contrat de collaboration EDF-IRIT/NEB, IRIT, Toulouse (2008)
8. Crépin, L., Demazeau, Y., Boissier, O., Jacquenet, F.: Privacy preservation in a decentralized calendar system. In Demazeau, Y., et al., eds.: 7th Intl. Conf. on Practical Aspects of Agents and Multi-agent Systems (PAAMS 2009). Number 55 in Advances in Soft Computing. Springer, Berlin (2009) 525–537
9. Cranefield, S., Winikoff, M.: Verifying social expectations by model checking truncated paths. In Hübner, J.F. et al., eds.: Coordination, Organizations, Institutions and Norms in Agent Systems IV. Number 5428 in LNAI. Springer (2009) 204–219
10. Cranefield, S.: Modelling and monitoring social expectations in multi-agent systems. In Vázquez-Salceda et al., eds.: Coordination, Organizations, Institutions and Norms in Agent Systems II. Number 4386 in LNAI. Springer (2007) 308–321
11. Zadeh, L.A.: Fuzzy sets. Information and Control **8** (1965) 338–353
12. Petri, C.A.: Concurreny. In Brauer, W., ed.: Net Theory and Applications. Number 84 in LNCS. Berlin, Springer-Verlag (1980) 251–260

Towards a Learning Agent Architecture for Cross-Map Transfer

Cédric Herpson[1] and Vincent Corruble

Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie,
104 avenue du Président Kennedy 75016 Paris, France
{cedric.herpson,vincent.corruble}@lip6.fr

Abstract. The capacity to apply knowledge in a context different than the one in which it was learned is still an open research question within the area of learning agents. This paper specifically addresses the issue of transfer of knowledge acquired through online learning in a partially observable environment characterized by its 2D geographical configuration. We propose an autonomous agent architecture combining an agent-centered representation and the supervised and unsupervised learning of discriminating concepts to achieve efficient cross-map transfer. Our preliminary experiments on a grid-world environment where two agents duel each other show that the agent's performances are improved through learning, including when it is tested on a map it has not yet seen.

Keywords: Generalization, Abstraction, Agent, Transfer Learning

1 Introduction

Learning and transfer of knowledge is a cross-discipline issue for those interested in understanding or simulating intelligent behavior. Knowledge transfer opens up the possibility of improving both learning speed and global performance of a system on a problem close to a known one. It has been addressed in particular by research in cognitive psychology and neuroscience [1]. In Artificial Intelligence, the ability to generalize has been an important focus of study, but mainly in the field of classification [2], i.e. for the identification of new instances of a given concept (e.g. spam recognition). Within the agent community, though a significant amount of research addresses the exploration of unknown environments, it mostly focuses on the exploration optimality, regardless of whether it is carried out in a mono- [18] or multi-agent setting [17]. Thus, with the notable exception of Case Base Reasoning [5], until recently, little effort had been put into the transfer of learned knowledge. Indeed, the need for an agent architecture which integrates learning and transfer capacities has become crucial in recent years with the development of new

[1] Corresponding author

application domains using, or open to the use of, autonomous agents, such as video games, military simulations or general public robotics.
Work in the area of strategy games is one concrete example; it has led to techniques which let agents learn strategies through playing [3,4]. Yet, learned strategies are only relevant to the game context in which they have been learned, that is to say a scenario, and more pointedly a specific "game map". Hence, each new scenario requires a new phase of learning since previous experience on previous game maps is not put to use. Recent work try to go beyond these limitations. However, they cannot be easily applied in open and/or complex environments as they require the full description of the state-action space.
The goal of this paper is to present a robust learning agent architecture able to realize efficient cross-map transfer in open or complex environments. To this aim, its main feature is to discover from an agent-centered point of view abstract concepts which allow the transfer of knowledge learned on a given topology to a different one, yet unknown. In the following we briefly introduce relevant literature. Next, we describe the proposed architecture and a preliminary evaluation thereof in a simplified game environment where the relevance of a learned concept is evaluated from the standpoint of the agent's performances. Finally, we discuss experimental results and current limitations of our approach before concluding with some perspectives.

2 Related Work

An important part of recent work tackles the learning and transfer of knowledge problem by means of the Reinforcement Learning (RL) framework [6, 7, 8, 16]. To be able to generalize and to use RL in complex worlds, [6] assumes that states of the modeled Markovian Decision Process (MDP) are of different types (in a restricted number). This assumption simplifies the representation of the environment enough to allow the use of RL methods. However, the different types of states are determined *a priori*, which reduces the use of this technique to known and closed worlds. With a similar idea, i.e. to generalize across states, Dzeroski & Driessen [4] proposed the Relational Reinforcement Learning approach (RRL). This attractive combination of RL and Inductive Logic Programming is based upon the relational structure of the problem to abstract from the current goal. However, if RRL showed generalization abilities which are absent from traditional RL techniques, the use of first-order logic generates a sharp increase in the number of components usable for the complete description of each state. This increase led to the emergence of the same drawbacks as the tabular representation initially used in RL and does not allow its use in complex environments.
In the last few years, research on learning and transfer has found excellent test beds in the area of games. Indeed, games, and especially Real-Time Strategy (RTS) ones, provide research environments which are rich and complex, often stochastic, and favor experimental work [9]. In this context, a first approach of transfer models game states at a high level of abstraction meant to describe the state globally. Alternatively, another approach is to consider each unit or character of the game as an autonomous

agent interacting with the environment and processing only the information it can perceive locally.

Following this last approach, [10] tackles knowledge transfer within the SOAR cognitive architecture [11] and follows an agent-centered approach. The test bed is a first-person-shooter map where the agent's goal is to move from one point to another, and where some topological changes occur between the learning and test phases. It uses learning mechanisms based upon creation/modification[2] rules and on the storage of all spatial knowledge to reach this goal. However, storing all the topological information requires the use of huge amounts of memory which does not allow applying it on a larger scale.

With the same application domain, [16] addresses the problem at a high level of abstraction. It represents the structure of the problem as a relational MDP to tackle the planning problem on different maps in a RTS. However, the required full description of the state-action space does not allow the use of this approach in open or partially observable environments. More recently, [8] has proposed an architecture where an agent controls from a central point of view all the units from its side using a CBR learner defined as a combination of CBR and RL. This proposal was tested on a strategy game simulator, on a scenario where each side aims to destroy enemy forces. Their system builds a high-level description of the game state using only global information such as percentage of units "alive" on each side, percentage of territory controlled, etc. This radical abstraction makes the representation independent from any given map or game context. The architecture then selects an action solely as a function of this high-level description. Results obtained with this approach indicate its ability to reuse learned knowledge when initial positions and/or number of units vary. However, the fact that the game state description on which decisions are made is completely unrelated to the context (including its topology) constitutes a major obstacle to more ambitious transfer. Thus, a map of higher complexity, or a complete change of environment will not impact the state description and lead therefore to only one high level description for two distinct situations. As a result, only one action will be chosen where two different actions have to be selected.

In summary, the research described above make strong assumptions such as being totally independent of the context or knowing the structure of the problem in advance. In the next section, our proposal attempts to go beyond these limitations.

3 An Architecture for Cross-Map Transfer

3.1 Abstraction and Situated Representation

We consider a representation using the notion of a situated agent. It lets one change from a central, globalized point of view, often used in work on strategy games to an agent-centered perspective.

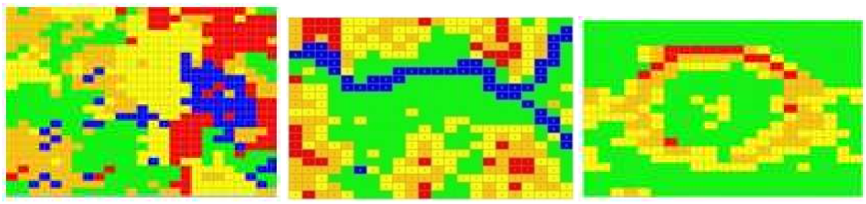
[2] In Soar, agent's knowledge is represented as rules.

Definition 1 (Situation). *A situation is the world view as perceived by the agent from its sensors at a given moment.*

A situation is the basic level of information; data obtained from sensors are expressed as a set of attribute-value couples. Our first assumption is that the elimination of information outside the sensors' field of view, considered as a "passive" abstraction [12] and usually seen as a constraint in domains such as situated robotics [13], brings significant advantages here. It offers the possibility for the agent's reasoning to be independent of map-specific (x,y) coordinates and of the environment's complexity, in terms of size as well as richness. However, by using agent-centered limited perceptions and representations, we lose some possibly relevant state information and enter the realm of partially observable environments. Thus, if the environment contains multiple agents, they become unobserved and unpredictable; which means that the environment is not stationary anymore. These properties are usually working hypotheses in most learning agent frameworks, as they are necessary to guarantee convergence with typical learning algorithms. Doing without them requires special attention.

3.2 The Duel Example

Before presenting in detail our architecture, we introduce an example that will be used both to illustrate our following explanations as well as to evaluate it in the experimental section. Consider a grid-world type of environment where each location on the grid is characterized by its altitude besides its (x,y) coordinates.



(a) (b) (c)

Fig. 1. Map (a) corresponds to the learning environment. Maps (b) and (c) respectively correspond to the Mountain and Cirque test environments. Altitude varies from blue (level 0, the lowest) to red (highest).

The effective field of view of an agent evolving on this map takes into account the obstacles present. As a consequence, the field of view of an agent located on a high position is better than the one of an agent located downhill.

The chosen scenario consists of a duel between two agents evolving within the above grid-world. Each agent's goal is to become the last survivor. They have available a range weapon and have to hit their competitor twice to win. The probability of hitting when shooting depends on the shooter-target distance as well as the angle of incidence of the shooting. A horizontal shoot has a very high hit probability, whereas when two agents are located at different altitudes, the one positioned higher will have a hit probability much higher than the lower one.

Final goal : alone(x) $\wedge$ alive(x)
State : ammo(x) $\in$ {true,false} and Status $\in$ {hurt,dead,ok}
Rules : ok(x) $\vee$ hurt(x) $\rightarrow$ alive(x) ; dead(x) $\rightarrow$ alone(y) [with x $\neq$ y] ;
hurt(x) $\wedge$ hit(x) $\rightarrow$ dead(x) ; ok(x) $\wedge$ hit(x) $\rightarrow$ hurt(x)
Actions : ammo(x) $\wedge$ see(x,y) $\rightarrow$ Shoot(x,y) ;
move and rotation towards the 4 cardinal directions.

Fig. 2. A simplified version of available rules for a duel

3.3 Architecture

Our architecture is based on a perception-action loop involving several components.

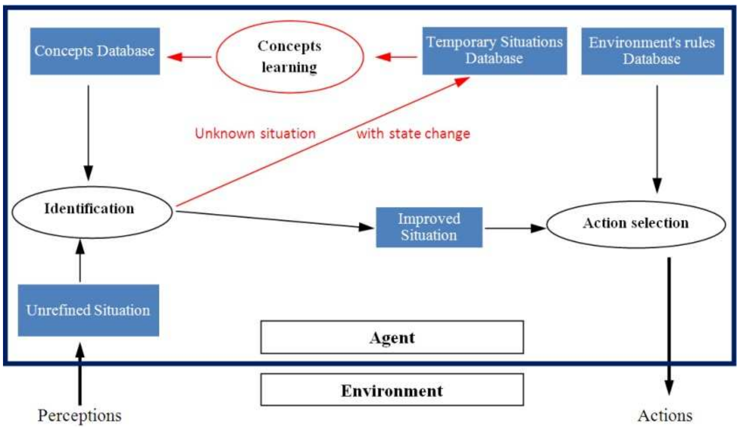


Fig. 3. General architecture of a learning agent for cross-map transfer

As shown in Figure 3, an agent has a memory to save facts, learned concepts and rules about the environment (*Concepts* and *Environment rules' Databases*).

Definition 2 (Concept) *Let* ***pre*** $\longrightarrow$ ***Action*** *be a rule.*
A concept is a couple <***pre***, ***descr***> *where* **descr** *is the agent's representation of the precondition* ***pre****, learned from its world view (the situations).*

Thus, the learning mechanism (*Concept Learning*), using both supervised and unsupervised methods, continuously extracts patterns from perceived situations and associates them to the premise of the agent's actions/decision rules (red process in Fig 3). An inference engine (*Action Selection*) uses the relations between these different elements in order to select the actions that can lead the agent toward its current goal. The automatic recognition in the current situation of a prior learned pattern (*Identification*) helps reaching better decisions. Thus, learned concepts are used as an interface between the physical sphere and the representation sphere.

3.3.1 Learning Phase

As shown in the duel's rules example, Figure 2, the agent's internal state is defined by one or more variables. An agent's change of internal state corresponds to a change in one of its attributes. A change occurs as a consequence of the agent's last action or of another agent's one. Without prior knowledge, the learning or discovery of a new concept proceeds in three stages:

1. Save in the *Temporary Situation Base* (TSB) all situations newly encountered when a change in the agent's internal state occurs. Each saved situation is associated to the pre-condition part of the last action rule used.
2. Apply a learning algorithm to infer a pattern when the number of situations associated with the same precondition is above a threshold N.
3. Add the resulting couple < pre, pattern> into the *Concept Base (*CB).

In our duel example, if we consider that the weapon cartridge clip is not empty, then the *Shoot* action can only be used when an agent perceives its enemy. After a successful shot has killed its enemy, the shooter-agent's final goal is reached. This internal state change triggers the learning algorithm. Thus, the current situation is associated with the pre-condition *see(x,y)*. After N duels, the agent discovers that being located, for example, at a high point, improves the chances of seeing the enemy and therefore of winning.

In stage (2), two types of learning mechanism are used depending on whether the target concept is related to the final goal (case A) or to a sub-goal (case B).

Definition 3 (Sub-goal*).* *A sub-goal is an intermediary objective considered by the agent as a necessary condition to reach its final goal.*

The reason for this distinction is that we hypothesized that concepts linked to win-lose situations can be defined within a single representation. Assuming that it is true, winning and losing situations can be opposed and form two classes which can be discriminated more easily using classical learning algorithms. However, if the targeted concept is a sub-goal - e.g. to find a cartridge clip at the beginning of a duel - the only situations available to store will be the ones obtained when this sub-goal is reached.

A) Concept related to the final goal, learned under supervision

Environments in which agents evolve are, just as the real world, stochastic. There is therefore a significant amount of noise in the data from which we wish to extract a relevant situation's pattern. To tackle this, our algorithm (Algorithm 1 below) firstly removes some noise from data using a majority vote process (line 1 to 9). After this step, it then infers a pattern among remaining situations using a probabilistic classification tree[3] based on the C4.5 algorithm [14].

---

[3] The classification tree was selected after having tested a number of learning algorithms including Clustering, Boosting on MLP, KNN, Kmoy and SVM.

Let **VS**: *VictorySituations*,**DS**: *DefeatSituations and* **pre**: *precondition*

$VS \leftarrow \{vs \in TSB \mid vs \in <pre,descr>\}$
$DS \leftarrow \{ds \in TSB \mid ds \in <pre,descr>\}$
$\forall\ x \in VS \bigcup DS$
  $\forall\ y \in VS \setminus \{x\}$
    If $(dist(x,y) < \eta)$ **then** $x.nbPos \leftarrow x.nbPos+1$
  $\forall\ y \in DS \setminus \{x\}$
    If $(dist(x,y) < \theta)$ **then** $x.nbNeg \leftarrow x.nbNeg+1$
  If $((\frac{x.nbPos}{|VS|} < \frac{x.nbNeg}{|DS|}) \wedge x \in DS)$ **then** $nDS \leftarrow nDS \bigcup \{x\}$
  If $((\frac{x.nbPos}{|VS|} > \frac{x.nbNeg}{|DS|}) \wedge x \in VS)$ **then** $nVS \leftarrow nVS \bigcup \{x\}$
$pattern \leftarrow C4.5(nVS,nDS)$
$Concept\ Base \leftarrow Concept\ Base \bigcup <pre,pattern>$
$TSB \leftarrow TSB \setminus (VS \bigcup DS)$

Algorithm 1. Learning a concept related to the final goal

Once these two stages have been realized, the new association between a pattern and a precondition is added to the *Concept Base* (CB). Finally, the situations associated to the precondition of the action having triggered the learning phase are removed from the *Temporary Situation Base* (TSB).

B) Concept related to a sub-goal : process of characterization

Characterizing the concept consists of identifying attributes whose values show a pattern within the subset of situations associated with one pre-condition. Let us consider that attribute L is representative of a given concept whenever more than α of its instances lie within the interval defined by a variation of γ around its mean value. The resulting situation constitutes a prototype for the concept which originated the N situations under consideration. In a manner similar to what was presented in Case A above, situations used from the TSB are deleted, and the new concept is added to the CB.

In order to distinguish the various processes at work in our architecture, we have so far considered that the CB was initially empty during the recording of <pre, new situation> couples in the TSB. However, our architecture needs to be able to use a concept already learned. As a consequence, when the CB is not empty during the recording of new situations in the TSB, a similarity measure is applied between new incoming situations and the descriptions of learned concepts. If a correspondence is detected (similarity above a threshold), the couple <pre, pattern> in the CB is updated with an additional precondition and the incoming situation is deleted. The same similarity measure will be used in the decision phase presented in the following.

3.3.2 Decision Phase

The inference engine, prolog, is interfaced with the agent's learning and action mechanisms (themselves implemented in Java). At each time step, the situation perceived by the agent updates the set of facts available to the inference engine. Prolog then uses its model of the environment and known facts so as to 'prove' the desired goal and to select the best action.

```
1 : While (!goalReached)
2 :     currentSit ← agent.updateCurrentSit();
3 :     agent.updateFactToProlog(currentSit);
4 :     action ← agent.prolog.prove();
5 :     If (action!=null) then
6 :         agent.doAction(action);
7 :     else
8 :         descr ← agent.CB.search(agent.prolog.getMissingPre());
9 :         If (descr!=null) then
10 :            location ← agent.searchInFov(descr);
11 :            x ← randomInt();
12 :            If (x > ε ∧ location!=null) then
13 :            agent.doMove(location);
14 :        else
15 :             agent.randomMove();
16 : EndWhile
```

Algorithm 2. Decision-Action loop

During the identification stage, at line 10, the agent considers virtually all the positions it can perceive in its field of view and evaluates their quality relative to the targeted concept. The identification carried out at each location works in two different manners depending on whether the description obtained from the Concept Base (line 8) is an identification function (Case A: concept related to the end goal) or a prototype (Case B: concept related to a sub-goal).

A) Identification of a concept related to the end goal.

Using the disambiguation process previously introduced in Algorithm 1, virtual situations detected within the horizon of the agent are filtered. Then, the classification tree computes for each situation its relevance level to the concept the agent is looking for (where 1 means certainty and 0 the opposite). To limit the risk of misclassification, only a virtual situation with a confidence value above a threshold β (β in [0, 1]) can be selected.

B) Identification of a concept related to a sub-goal

In this case, the element extracted from the Concept Base is a situation prototypical of the target concept. To estimate the similarity over attributes describing the various situations, a non-informed similarity measure based on the Euclidian distances is applied between the prototypical situations and perceived ones[4]. As for a concept related to the end goal, only a virtual situation with a similarity value above the threshold β is selected.

4 Experimental Evaluation

In this section, we describe an empirical evaluation of the learning and transfer capacity of our architecture. As introduced in 3.2, we chose to test our architecture in a grid-word type environment where two agents duel each other. To this aim, we set

[4] The Euclidian distances were selected after initial experiments with Manhattan, Euclidian and Mahalanobis distances.

up three maps of varying topologies, sizes and maximum altitudes. Each action's availability depends on the agent's current situation. The shoot action is automatically executed when an agent perceives its enemy in its field of view. This was decided so as to reduce the simulation time. The result of the shoot action (target hit or not) is stochastic and depends on a number of characteristics unknown to the agent.

4.1 Experimental Setup

To test the impact of learning a single concept on the performances, changes of internal states leading to the storing of situations and the learning of a concept are only to be related to end-game situations. We therefore consider that ammunitions are unlimited. N, the threshold that triggers the learning algorithm is set to 20. The β parameter is set to 0,8. Parameters α and γ are set respectively to 0,75 (overqualified majority) and 0,1. η, θ and ε, the different parameters in algorithms 1 and 2, are set to 0,1. Performances are evaluated based on three criteria : the percentage of victories obtained by each agent, the average number of time steps needed to end an episode or game (i.e. when a agent wins) and the evolution of the proportion of the environment explored by each agent over time.

The first experiment aims to evaluate the ability of an agent to transfer knowledge to a new environment. To obtain a baseline, 1000 duels between two random agents are run for the three considered environments. Next, one learning agent runs a series of episodes against a random one on the learning environment until it learns one concept related to the final-goal. Then, the learning step is deactivated and we run a series of 1000 episodes opposing a random agent and the trained agent on the three environments.

The second experiment aims to evaluate the capacity of our architecture to produce a behavior of a higher efficiency by adding a memory component. Indeed, in the first experiments, stimuli from the agent's sensors do not provide any temporal information. An agent is not able to know if it is currently in an area it just left a few steps ago. Consequently, it can move for a long time in an area where there is no enemy. If information from the new sensor, which is added to the description of each situation, is recognized as relevant during the concept learning, it may improve the results because of a better spatial exploration. Following the protocol previously introduced for the first experiments, an agent thus upgraded is therefore evaluated against a random one.

In the presentation of all results, we refer to the three types of agents as follows:

- *Random*: baseline agent, with no knowledge or learning mechanism.
- *Intel_1*: agent having discovered/learned a concept related to the end goal.
- *Intel_2*: agent with a short-term memory having learned a concept related to the end goal.

4.2 Results

The tree learned in the first experiment (fig. 4 below) shows that the agent will favor situations with a good field of view, or, if the field of view is considered average, situations where average altitude around the agent location is lower than its own. Thus the learned concept drives the agent to search and follow the map's ridge paths.

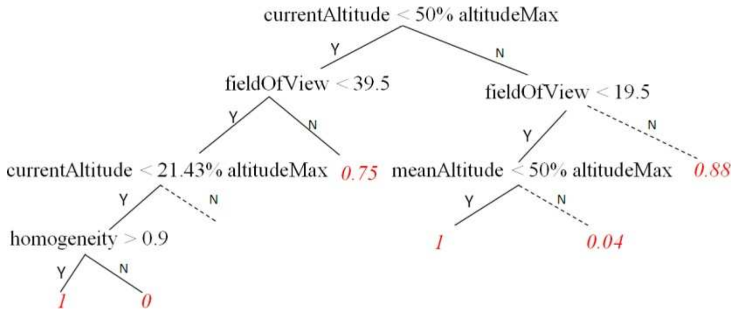


Fig. 4. Tree automatically generated and associated to the *see(x,y)* precondition. A value above 0.75 indicates a favorable situation. Dotted lines indicate sub-trees that were manually pruned from the figure for better clarity.

The results obtained for the two first evaluation criteria on the three environments are presented in fig 5. They show that the concept learned is sufficiently relevant to improve performance on the training environment. Furthermore, results obtained on the two test environments are even better than those from the training environment.

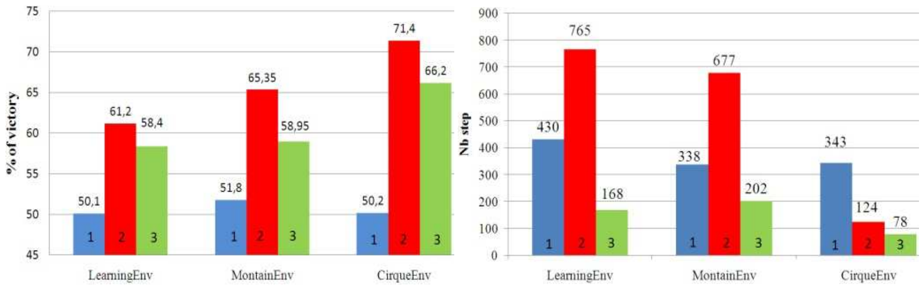


Fig. 5. *(a)* represents the percentage of wins for agents *random* (1), *Intel_1*(2) and *Intel_2* (3) respectively, all against a random agent - *(b)* shows the average duration of a duel for the three types of agents facing a random agent. In each case, results are given in the three different environments after 1000 duels.

Figure 5.a shows that *Intel_1* improves results over a random agent by 22% in the training environment and by 26 to 42% on test environments. *Intel_2* improves results by 15 to 32% depending on the environment. However, as it performs worse than *Intel_1*, it looks as if its memory is more of a hindrance than an advantage here. Results obtained in fig 5.b, with the second evaluation criterion (the length of a duel) modify this last result: *Intel_1* increases the average duel time for two of the three environments while *Intel_2* drastically reduces the time on all three of them. Results of *Intel_2* indicate that the temporal information is recognized as relevant during the concept learning and is profitable, considering the duration of a duel at least. For the third evaluation criterion, the general shape of graphs is similar for all 3 environments so we give below in figure 6 the results for the Mountain environment only.

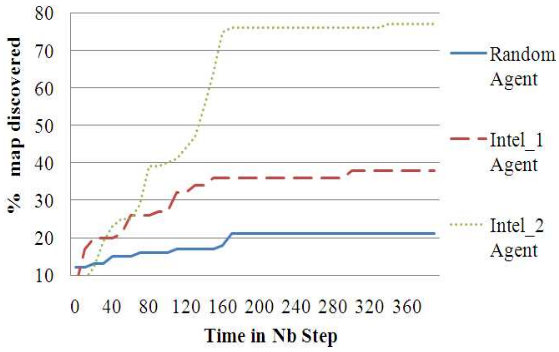


Fig. 6. Evolution over time of the proportion of the environment explored, for each agent in the Mountain environment.

Because of its memory the *Intel_2* agent goes over a larger part of the environment in a given time interval. It is therefore more likely to encounter its opponent than *Intel_1*. The latter, without short-term memory, repeatedly crosses the same zone. In this context, *Intel_1* has an advantage: as actions are simultaneous, the one moving can very well enter the field of view of its opponent and get shot immediately. Agent *Intel_1* takes "fewer risks", and therefore plays longer and wins more often.

5 Discussion

Though our results provide evidence regarding the validity of the principles behind our learning architecture for the transfer of knowledge, a lot of work remains. First of all, the unsupervised learning of concepts related to sub-goals must be evaluated. For this purpose, experiments where agents begin each duel without ammunition are in progress. Next, the topology plays a large part in determining the ability of the agents. Thus, a thorough study in more various and more complex environments has to be done to prove our architecture's robustness when scaling up. Besides these points, our architecture suffers in its current form from several limitations. Thus, our architecture lets our agents identify desirable situations within their environment by making use of the game's rules.They are,however, not able to reason on what constitutes undesirable situations for them. Using a model of the environment to model the goals of the opponent would be relevant to avoid situations which are desirable to the enemy.

Irrespective of these points, it is interesting to note that our architecture does not seem to suffer from the agent-centric drawbacks highlighted in [15]. Indeed, results show that the impact of switching from a complete world representation to a partially observable one makes the learning performances fall drastically to the point of failing to reach the targeted goal. One possible explanation is that their work relies on RL algorithms. In this framework, properties of perceived objects are independent of their functional utility. Conversely, in the line of research we follow here, the essential role of perception is to give access to information organizing the action. Thus, perceived properties of objects must be dependent on the actions capacity of the agent. Our architecture relies on this principle and lets an agent learn to choose a state with the goal of executing an action resulting from a reasoning process.

6 Conclusion and Perspectives

We have proposed a new learning agent architecture for cross-map transfer. Initial results are encouraging. They show that our architecture does lead to a relevant knowledge transfer in stochastic environments regarding topological information, which proves efficient when the environment changes. Besides some necessary improvements discussed in section 5, one short-term perspective is to evaluate the efficiency of our learning architecture facing the Reinforcement Learning approach. This experiment will allow us to compare both the adaptability and efficiency of our approach to the RL-based ones in the transfer context. With a long-term view, an important perspective for this work is to extend the architecture to address the multi-agent framework. The choice of knowledge representation proposed here is compatible with the use of a command hierarchy where low-level units with local information can interact and share their knowledge at the strategic level. We aim to tackle the issue of communication and coordination between agents at this level.

References

1. S. Thrun, L.P.: Learning to learn. Norwell edn. Kluwer Academic (1998)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
3. Madeira, C.: Adaptive Agents for Modern Strategy Games: an Approach Based on Reinforcement Learning. PhD thesis, University of Paris VI (2007)
4. M. Ponsen, H. Muoz-Avila, P.S., Aha, D.: Automatically generating game tactics through evolutionary learning. AI Magazine (2006)
5. Watson, I., Marir, F.: Case-based reasoning: a review. In: KE. (1994) 327-354
6. Bethany R. Leffler, M.L.L., Edmunds Littman, T.: Efficient reinforcement learning with relocatable action models. In: Proc of AAAI. (2007)
7. K. Driessens, R. Givan, P.T.: Rrl : An overview. In: Proc of ICML. (2004)
8. M.Sharma, M.Holmes, Santamaria, J.C., A.Irani, Lee, C., Ram, I.J.A.: Transfer learning in real-time strategy games using hybrid cbr/rl. In: IJCAI. (2007)
9. D. Aha, M.M., Ponsen, M.: Learning to win: Case-based plan selection in a real-time strategy game. In: Proc of ICCBR. (2005)
10. Gorski, N., Laird., J.: Experiments in transfer across multiple learning mechanisms. In: Proc of ICML. (2006)
11. Lehman, J. Laird, R.: A gentle introduction to soar. (2006)
12. Agre, P.: The dynamic structure of everyday life. PhD thesis, MIT (1988)
13. N. Bredeche, S.Z., Zucker, J.: Perceptual learning and abstraction in machine learning. Systems, Man and Cybernetics, Part C, IEEE (2006) 172-181
14. Breiman, L.: Classification and Regression Trees. Boca Raton, London (1993)
15. S. Finney, P. Wakker, L.K., Oates, T.: The thing that we tried didn't work very well : Deictic representation in reinforcement learning. In: Proc of UAI, Morgan Kaufmann (2002)
16. C. Guestrin, D. Koller, C. Gearhart, N. Kanodia : Generalizing Plans to New Environments in Relational MDPs In: IJCAI.(2003).
17. Burgard, W., Moors, M., Stachniss, C., Schneider, F.: Coordinated multi-robot exploration. IEEE Transactions on Robotics 21 (2005) 376-386
18. Albers, S., Kursawe, K., Schuierer, S.: Exploring unknown environments with obstacles. In Proc. 10th ACM-SIAM Sympos. Discrete Algorithms. (1998) 842-843

A Flexible Agent-Based Framework to Control Virtual Characters

Luís Moniz, Graça Gaspar, Ricardo Abreu, Ana Paula Cláudio, and Maria Beatriz Carmo

Department of Informatics, University of Lisbon, Portugal
{hal,gg}@di.fc.ul.pt ricardolafabreu@gmail.com {apc,bc}@di.fc.ul.pt

Abstract. We present a generic platform, called IViHumans, for materializing a system of multiple intelligent agents into virtual humans. It is intended to be flexible and applicable to diverse realistic simulation environments. The platform is composed of two layers: the Graphical Layer, and the Artificial Intelligence Layer, that can run autonomously. The architecture of our platform was conceived in order to allow the two layers to control the virtual agents at different detail levels.

1 Introduction

The simulation of virtual humans (VHs) is an immense challenge that requires the resolution of many problems in various areas. In order to simulate virtual humans in virtual worlds we have to combine realistic representations of humans, including the capability of expressing emotions, along with mechanisms to simulate the behaviour [1].

In 1987, Craig Reynolds was responsible for bridging the gap between artificial life and computer animation, introducing behavioural animation techniques [2]. Since then, intensive research in this area has been performed giving rise to several techniques widely used with many different purposes. Entertainment industry has been exploring and using these techniques in games like "The Sims" and films like "The Lord of the Rings". A different context of application, covered by an important number of approaches, allows the exploration of stressful social situations in the safety of the virtual world [3–5] and, fully or partially, covers a number of technical areas in Artificial Intelligence.

Research in VHs needs to use the results from Artificial Intelligence in order to achieve believable characters, while research in Multi-Agent intelligent systems can profit from virtual humans frameworks for evaluation and increasing the usability of the applications. Believable characters require that agents' behaviour should not be scripted, but should rather emerge as a result of autonomous agents' interaction, in a shared environment.

We have been developing a graphical visualization platform, called IViHumans, for multi-agent system execution with the intention of applying it to the development of realistic and compelling simulation environments for some real-world situations, such as training, education and entertainment. However it is intended to be a generic platform, not specialized in a particular problem or domain.

To pursuit this goal, the IViHumans platform is composed of two separate layers, one for Graphical Processing (GP) and one for the Artificial Intelligence (AI) computation. A special concern then is the interconnection between these two layers. They must be able to express the control of the embodied virtual agents at different detail levels. Support for relevant aspects such as sensory honesty, as described in [6], or the attention focusing needed for effective planning must be distributed among the two layers, maintaining a clear separation of concerns and responsibilities. Also, the interconnection between the two layers must accommodate their potentially different response speeds and account for delays of communication between them, without compromising the believability of the virtual human characters.

In the last decade some other authors have presented work integrating multi-agent systems with graphics components. In [7] a game-oriented multi-agent framework, built over the JADE multi-Agent System platform is presented. That approach is similar to ours, in that the MAS module and the visualization module work independently of each other. However in their system the simulation is centralized in the MAS, having the visualization module with the sole duty of exhibiting the world without allowing interaction.

Several other authors have used BDI agent platforms together with game engines or specific visualization platforms. Some of those works concentrate on particular types of applications and agents. For instance, in [8] an environment based on the Unreal Tournament game engine and the Soar AI engine is described, concentrating in the development of complex AI agents for interactive drama applications. In [9] BDI agents that model expert players of Quake 2 are developed.

In [10, 11] the BDI model is also used to control animated characters in virtual worlds. In their work, 3D articulated characters are controlled in real time by cognitive agents that are clients of the environment which, in its turn, acts as a server and is responsible for managing the information that can be perceived by the agents. However their architecture is limited in that everything an agent can perceive must be listed a priori in a list of boolean statements.

In [12] the JACK agent language is extended with a specific perceptual/motor system that allows the agents to interact with a graphical interface. Our approach is instead to make the GP layer responsible for the perception and basic motion of the agents, thus decoupling the perceptual/motor system from the cognitive part of the agents and making it more independent of the agents implementation. In [13] a military simulation tool, ROE3, is presented where the agents are also implemented using JACK. In ROE3, an Integration Layer translates messages between the agent and the synthetic environment, being able to aggregate raw percepts into higher level percepts. In this respect, our platform incorporates a similar translation mechanism, as part of the AI Layer, in the form of interface agents.

In the following section we present the general architecture of the IViHumans platform. In section 3 a brief general description of the GP Layer is presented followed by some detail of the implementation of the movement of the synthetic characters. In section 4 we present the AI Layer, describing the several types of agents that constitute the MAS system, namely the interface agents, the intelligent agents and the monitoring agents. The last section summarizes this work and discusses some future work.

2 Architecture

The IViHumans platform separates the GP layer from the AI layer, an approach that was already found in works such as the JGOMAS system and the ROE3 architecture. However our proposal differs from them by having the amount of responsibility of each layer well balanced. For instance, in what regards sensory honesty, the physical limits to what can be perceived by a character are controlled by the GP layer while the cognitive restrictions reside on the AI layer. Also, the GP layer is responsible for quickly handling low-level aspects, such as collision response, while the AI layer deals with the more complex cognitive behavior, using symbolic representations. As a side effect, this architecture also reduces considerably the communication overhead.

The GP layer is built on top of the rendering engine *OGRE* (www.ogre3d.org) and relies on the rigid body dynamics engine *ODE*(www.ode.org). The Multi-Agent system – the core of the AI layer – is built upon the JADE (jade.tilab.com) platform. The choice of all the underlying software for the platform obeyed well defined criteria, such as the quality of the provided features, the price or the existence of an active community of users.

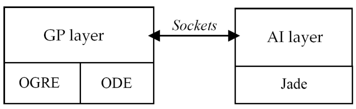


Fig. 1: The Graphical Processing and Artificial Intelligence Layers.

The communication between the two layers (figure 1) is performed by means of TCP sockets. Two reasons underlie this option: first, each layer uses a different implementation language (*C++* for the GP layer and *Java* for the AI layer); second, we intend to use the platform in a distributed environment. The GP Layer is responsible for the representation of all the elements contained in the virtual world, among which the virtual humans are the most important. Because the environments are dynamic, the GP layer must reproduce the appropriate animations that carry the flow of occurrences, consistently enacting the evolution of the world and its components.

3 Graphical Layer

The GP layer is responsible for exhibiting the evolution of the virtual world, representing all the elements contained in it (among which the VHs are the most important) constantly updating their state, and reproducing the appropriate animations.

The IViHumans platform handles VHs whose actions rely, for the time being, on three main skills: perception, movement, and emotion expression. A VH perceives its environment through a ray-casting synthetic vision algorithm [14]. Besides the effects of emotions on their behavior – something that the AI layer may take care of – the VHs can show composite facial expressions to immediately convey their internal emotional state. An explanation of the techniques used to accomplish emotional expression of the VHs can be found at [15].

The motion of the VHs is supported by the concept of *steering behavior*, according to what was introduced by Craig Reynolds at [16]. We also follow his proposal for hierarchically categorizing movement in three layers: locomotion, steering and action selection (in order of growing abstraction). The former two are under the domain of the GP layer, while the last and topmost layer can either be included in the AI layer – which happens for virtual humans – or be absent – so that human users can directly control avatars. The action selection layer operates by activating and parameterizing steering behaviors to achieve the desired goals, according to agents' planning. Locomotion corresponds to the choice of the appropriate animations on the basis of speed and according to rules that are unique for each VH model. In the remainder of this section we detail how steering behaviors are integrated in the GP layer, as a sample grounding base for the explanation of the activity of the AI layer.

3.1 Movement of Characters

To implement the steering of the virtual humans, we closely follow Reynolds' proposals, although we introduce a few ideas of our own. Because movement through steering behaviors could be applied to a myriad of entities, we decided to bring it apart from the implementation of any particular entity. This way, we created a class that models any moving entity as a point mass, as Reynolds did with his vehicle model. This *MovingCharacter* class is not associated with OGRE, except in that it uses some basic data types provided by this library (e.g. vectors).

A *MovingCharacter* is essentially characterized by a mass, a position, a velocity and a vector that specifies his facing direction. This facing vector may be automatically updated when the character moves so that it is always tangent to the path the *MovingCharacter* follows. When the character is still, it is left unchanged. However, we did not want to restrict the movement of a virtual human in this way and so the automatic update of this vector is not mandatory. It should be deactivated if one intends to have the character moving in any way that requires his local depth axis not to be collinear with the velocity vector, that is, if the character must move in one direction while facing other way.

The movement of the *MovingCharacter* is ruled by steering behaviors that specify the forces he should apply on himself. The movement produced by these forces is computed according to the basic laws of classical physics, except for the fact that it is restricted by maximum values for force and velocity. These, as well as some other values that do not vary, are loaded upon construction from a configuration file that is unique for each character.

Steering behaviours are sometimes criticised for being hard-wired into the code [17]. In an attempt to overcome this problem, we separate the character from the actual behaviors, employing polymorphism and object oriented design in general. A *MovingCharacter* may have one instance of *SteeringBehavior*, which is the common interface to all the steering behaviors that may be defined (see figure 2). Any instance of *SteeringBehavior* can be plugged in and out of the *MovingCharacter* at run-time.

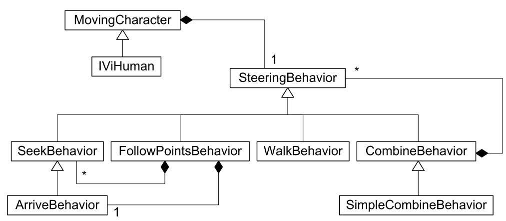


Fig. 2: Conceptual model of the main portion of the GP layer that deals with steering

The behaviors *SeekBehavior* and *ArriveBehavior* are directly inspired in the corresponding behaviors explained in [16]. The former returns a force that fully accelerates the character towards a target point, while the later does the same thing only until the character's distance to the target becomes lower than some predefined threshold. At this point, it will start to produce a force that is opposed to the movement and the character will decelerate until he eventually stops.

The behavior *WalkBehavior* does the same as one of the three rules that underlie the flocking behavior in [2]: it tries to match the velocity of the character with a given target velocity. This behavior is useful to have the AI control a virtual human at a lower level, as well as to have a user directly "piloting" an avatar. We also introduced the behavior *FollowPointsBehavior* that guides the character through a sequence of n targets by making him *seek* the first $n-1$ targets and *arrive* at the nth target. This last behavior is useful, for instance, to make the character follow a path plan. Besides these steering behaviors, many more may be implemented.

Although the *MovingCharacter* can only be associated with one *SteeringBehavior*, there is a special *SteeringBehavior* for combining more than one. It is called *CombineBehavior* and it is an abstract class that has the general functionality for coupling other *SteeringBehavior*s. This class can be extended to implement different ways of combining the forces calculated by encapsulated behaviors. Currently, we have just one such implementation – which lies in *SimpleCombineBehavior* – that simply adds the forces returned by contained behaviors, returning the net force. In the future, we intend to add other ways of combining *SteeringBehavior*s (e. g. with priorities). This design follows the *Composite Pattern*. In what concerns locomotion, the *IViHuman* objects map

Fig. 3: VHs moving independently

the abstract movement functionality of the class *MovingCharacter* into actual observable movement, by applying the correct transformations to the model that represents the corresponding VH and animating it accordingly (figure 3).

4 Artificial Intelligence Layer

While the GP layer hosts the bodies of the virtual humans, the AI layer manages their minds. Each virtual human is controlled by one or more agents that entitle him with intelligent behavior. The evolution of the world is due to the effects induced by both layers.

In our architecture the AI layer comprises two main categories of agents: interface agents and cognitive agents. The cognitive agents, together with the service agents and meta agents, are the intelligent components.

In order to give some degree of control over the agents, we include another class of agents: monitor agents. These agents act as supervisors, providing a mechanism to manipulate the cognitive agents directly, forcing their behaviour and consequently interfering in the simulation.

4.1 Interface Agents

The interface agents, one for each virtual entity, manage the communication with the virtual entities, receiving sensory information and sending commands (figure 4). Although these agents can act as a raw connection between both layers, they have two additional functions: to provide a sensing/acting cycle that further separates the communication aspects of the control of the virtual humans from the more complex, and possibly slower, cognitive aspects; and to offer a translation/filtering mechanism between crude data and symbolical representation. We can split these functions in four main components:

- **Sense**: requests sensory information from the GP layer at a defined rate, saving it into a buffer. The various requests from the cognitive agent component for sensor data are obtained from this buffer (the sensor data buffer). This isolates the sensor particulars (refresh rate and cycling) from the higher cognitive levels.
- **Act**: reads the next command from a buffer (the command buffer) and sends it to the GP layer. This feature detaches the agent cognitive level from the physical details, for instance, the number of commands that the GP layer is capable of processing in a time slot.
- **Sensor data translator/filter**: translates raw sensor information in symbolic equivalents or more abstract and constrained representations. This is achieved by splitting the information into clusters of similar data. For instance, a color name can correspond to an interval of values in the rgb gamma.
- **Command translator**: translates the higher level commands used by the cognitive components of the agent into the lower level commands used by the GP layer and saves them in the command buffer. The translation can be achieved by using predefined schemas for action decomposition. Another alternative is to incorporate a planner that produces in real time the desired action sequence.

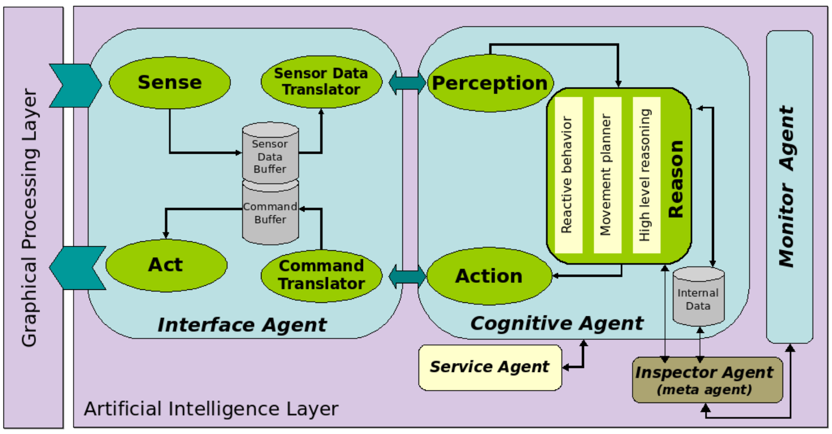


Fig. 4: Artificial Intelligence Layer detailed.

The GP layer, on the other hand, has its own interfacing component, composed by two unique objects: one that deals with the routing of messages to the adequate recipients and another one that acts as a special receiver for messages that are not directed to any particular entity, being instead targeted at a global manager, capable of operating global changes over the environment.

The concrete entities that must be able to receive orders from the AI layer, such as the virtual humans, will do so through an adapter component that manages to unravel the meaning of messages, translating them to the appropriate method calls.

4.2 Intelligent components

The objective of these components is to provide an intelligent control over the virtual entities. We can classify these components according to their main function into three categories: cognitive agents; service agents; and meta agents. As stated before, each entity in the virtual world is controlled by the intelligent component; this control is performed by a cognitive agent that acts as its mind (figure 4). Each of these agents has access to the interface agent to request sensory information and issue commands. Our architecture does not impose any restriction on how these agents are organized internally. The system designer is in charge of creating the control and coordination mechanisms to achieve the desired behaviours. Although we view a cognitive agent as a single entity, our system allows this agent to act as a representative of an agency, where a group of heterogeneous agents interact in the support of the virtual entity. Our base architectural template for these groups is a centralized society, with a central coordinator, and one external interface agent. In the example of figure 4 the agent main block corresponds to the reason structure, in this case a layered organization of progressively more complex behaviours.

Service agents, as the name suggests, provide extra services to the community. They can be shared by various cognitive agents, providing common facilities, available to all, instead of having each agent support its own version of the commodity. We group these services into three classes: core, aggregation and external services. Core services correspond to simple tasks that a single provider can solve. For instance, a path planner service, from which an agent can request a sequence of actions to reach a desired location. Aggregation services correspond to functionalities that support formation and cohesion of agencies. For instance, an internal message service that allows the agents to communicate. This feature should be carefully used; it allows building virtual communities without regarding the spacial location of the represented entities. In order to prevent undesirable use, we propose to develop this service on top of the platform messaging system incorporating adequate restriction mechanisms. Finally, external services provide agents with links to outside sources. For instance, it could be possible to create a remote web page that controls a virtual entity.

Meta agents act as invisible entities tracking, observing and controlling other agents' performance and behaviour. This kind of agent is not perceived by cognitive and service agents, it can act as an external observer that obtains information data and extracts conclusions. These agents have to be incorporated into the cognitive agency architecture in order to access private data, also they could provide a direct control mechanism over the other agents. For instance, if the user needs to guide the simulation into a specific track, he has a tool that can be used to directly modify the internal state of an agent, to add information or even to force some desired behaviour.

4.3 Monitoring Agents

We also propose the inclusion of other agents; these provide the user with custom interfaces to the MAS and give him some degree of control over the simulation. These agents use the meta agents in order to access the inner information of the cognitive agencies. An example is a monitor agent acting as an agency supervisor, allowing the user to inspect and control the behaviour of a group of agents, that control a virtual character. Another subgroup of monitoring agents corresponds to automatic triggers in the simulation; if a threshold is passed they can start an action. We could extrapolate some application scenarios: the action can have a direct effect in the simulation itself, for instance, if a new entity is created or a new scenario area is opened, several actions can be initiated; a process of data collection is started to measure some parameters; and the migration of agents to other machines if the overall performance is degraded.

We conclude this section by pointing out some key points of our architecture: independency from the graphical layer — although designed for this particular scenario, it can be used in other graphical environments; distributed characteristics (due to the JADE framework) that allow the simultaneous use different computers; and scalability by adding new agents/features/services and control facilities, tuning the system to a particular simulation.

5 Conclusions

The prevailing opinion of people working in the domain of virtual environments inhabited by virtual humans appears to be that there is still a field for subsequent research, despite the number of existing contributions. This judgment is primarily sustained by the fact that current results are still far from honestly mimicking human behavior in its interaction with the environment and with its peers.

In this paper we present a framework for construction and management of artificial characters in a virtual environment. We have focused mainly on the overall architecture of our system and we have presented, in a shallow form, a number of aspects that deserved to be treated in more depth. These include a complete description of the framework and its internal details of functioning, the methodology to design intelligent behavior and all the available modules built in our system. We also expose features that were so far included on the *IViHumans* platform as well as other possible approaches to its extension.

Our goal is to develop a general graphical visualization platform for multi-agent system execution and to apply it to the development of realistic and compelling simulation environments, and to incorporate results obtained in the area of emotion modeling. In the current stage of development the educational characteristics of the framework are not fully developed. We do believe the platform IViHumans is becoming a valuable tool for training and simulation based design purposes. It will be a working tool in post-graduation courses, used to learn basic concepts on animation and virtual environments. Moreover, it will be used as a growing kernel that is prepared to incorporate new functionalities implemented by students.

References

1. N. Magnenat-Thalmann and D. Thalmann. *Handbook of Virtual Humans.* John Wiley & Sons, 2004.
2. Craig W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, 21(4):25–34, 1987.
3. W. Swartout, J. Gratch, R. W. Hill, E. Hovy, S. Marsella, J. Rickel, and D. Traum. Toward virtual humans. *AI Mag.*, 27(2):96–108, 2006.
4. S. Marsella, L. W. Johnson, and C. LaBore. *Interactive Pedagogical Drama*, pages 301–308. ACM Press, Barcelona, Catalonia, Spain, 2000.
5. Jeff Rickel and W. Lewis Johnson. Integrating Pedagogical Capabilities in a Virtual Environment Agent. In *Proc. of the First Int. Conf. on Autonomous Agents (Agents'97)*, pages 30–38, New York, 1997. ACM Press.
6. D. Isla and B. Blumberg. New Challenges for Character-Based AI for Games. In *AAAI Spring Symposium on AI and Interactive Entertainment, 2002*, 2002.
7. A. Barella, C. Carrascosa, and V. J. Botti. JGOMAS: game-oriented multi-agent system based on JADE. In *Adv. in Comp. Entertainment Technology.* ACM, 2006.
8. Brian Magerko, John E. Laird, Mazin Assanie, Alex Kerfoot, and Devvan Stokes. AI Characters and Directors for Interactive Computer Games. In *16th Innovative Applications of Artificial Intelligence Conference*, pages 877–883, 2004.
9. Emma Norling and Liz Sonenberg. Creating Interactive Characters with BDI Agents. In *Australian Workshop on Interactive Entertainment*, 2004.
10. J. A. Torres, L. P. Nedel, and R. H. Bordini. Using the BDI Architecture to Produce Autonomous Characters in Virtual Worlds. In *IVA*, pages 197–201. Springer, 2003.
11. J. A. Torres, L. P. Nedel, and R. H. Bordini. Autonomous Agents with Multiple Foci of Attention in Virtual Environments. In *Int. Conf. on Computer Animation and Social Agents*, pages 197–201, 2004.
12. Emma Norling and Frank E. Ritter. Embodying the JACK Agent Architecture. In Brooks M., Corbett D., and Stumptner M., editors, *AI 2001: Advances in Artificial Intelligence*, volume 2256 of *LNCS*, pages 368–377, 2001.
13. R. Evertsz, F. E. Ritter, S. Russell, and D. Shepherdson. Modeling rules of engagement in computer-generated forces. In *Proc. of the 16th Conf. on Behavior Representation in Modeling and Simulation*, pages 123–134, 2007.
14. P. M. Semião, M. B. Carmo, and A. P. Cláudio. Implementing Vision in the IViHumans Platform. In *Ibero-American Symp. in Computer Graphics, SIACG 2006*, pages 56–59, 2006.
15. Ricardo Abreu, Ana Paula Cláudio, Maria Beatriz Carmo, Luís Moniz, and Graça Gaspar. Virtual Humans in the IViHumans Platform. In Dimitri Plemenos, editor, *11th International Conference 3IA 2008, the International Conference on Computer Graphics and Artificial Intelligence*, pages 157–162, 2008.

16. Craig W. Reynolds. Steering Behaviors for Autonomous Characters. In *Game Developers Conf.*, 1999.
17. John Funge, Xiaoyuan Tu, and Demetri Terzopoulos. Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters. In *Siggraph 1999, Computer Graphics Proceedings*, pages 29–38. Addison Wesley Longman, 1999.

Telecommunications Fraud: Problem Analysis - an Agent-based KDD Perspective

Eugénio Rosas and Cesar Analide

Department of Informatics
University of Minho
Braga, Portugal
eugeniorosas@gmail.com, analide@di.uminho.pt

Abstract. Telecommunications fraud is a problem that affects operators all around the world. Operators know that fraud cannot be completely eradicated. The solution to deal with this problem is to minimize the damages and cut down losses by detecting fraud situations as early as possible. Computer systems were developed or acquired, and experts were trained to detect these situations. Still, the operators have the need to evolve this process, in order to detect fraud earlier and also get a better understanding of the fraud attacks they suffer. In this paper the fraud problem is analyzed and a new approach to the problem is designed. This new approach, based on the profiling and KDD (Knowledge Discovery in Data) techniques, supported in a MAS (Multiagent System), does not replace the existing fraud detection systems; it uses them and their results to provide operators new fraud detection methods and new knowledge.

1 Introduction

1.1 Motivation

Following the exponential growth in the telecommunications sector in the end of the past century, the telecommunications operators face a new challenge: fraud. It is not only a risk, but a highly organized global business, that affects operators all over the world. In order to realize the severity of this problem, CFCA - Communications Fraud Control Association [1] (the Premier International Association for revenue assurance, loss prevention and fraud control) published some statistics stating that the annual global fraud losses in the telecoms sector are now between US$54 billion and US$60 billion, an increase of 52% since 2003.

Operators know that this is a problem that cannot be extinguished, so the approach to this problem is to minimize the losses. In order to do so, the operators have developed or acquired fraud detection systems. However, these systems were designed specifically to detect a fraud situation, this is, after it has happened. By detecting a fraud situation the operator prevents further losses, but since the fraud has already occurred, the operator has to support the fraud inherent losses. This is an important issue to the operators, because since they cannot totally eradicate fraud, it is very important to detect the fraud as soon as possible in order to minimize the losses.

1.2 Objectives

This paper has the following main objectives:

- Expose the fraud problem within the scope of telecommunications, enumerating the main causes for telecommunication fraud and the impact on the operators;
- Analyze the actual fraud detection solution, explaining the methods used by the fraud solution to detect fraud and analyzing how the solution can evolve;
- Define a new approach in order to evolve the actual fraud detection solution, defining new methods;
- Propose a solution model, which supports the methods previously defined. We will show that a MAS is a suitable approach to build a model to the solution of the problems addressed here.

1.3 Structure

The following chapter has a more detailed analysis to the fraud problem. After this analysis, an approach to solve the problem is suggested, presenting the data structure the solution is based on and the techniques used. The paper ends with a conclusion and an analysis to the work done.

2 Problem Analysis

2.1 Fraud causes

It is not possible to enumerate completely and exhaustively all the existing fraud types. Due to the constant evolution of technology existing fraud types are adapted and new fraud types are developed all the time. However, there is a set of the major fraud causes that are the ones of most concern to the telecommunications operators at the time. These are:

- **Subscription fraud** - one of the most common fraud types along with the SIM cloning. The fraudster obtains the service from the operator with no intention to pay for it, using a false identity. The damage this fraud type causes depends on the intention of the fraudster: using the service for personal use until he is detected; on a more sophisticated level, the fraudster can use the service in order to profit from the use of it.
- **Bypass fraud** - deprives the terminating operator of interconnect termination fees for incoming international calls. This is usually done using VoIP technology to bypass international calls.
- **SIM cloning** - a fraudster clones an existing normal SIM card. The software to clone SIM cards is available on the internet, so if a fraudster has physical access to a SIM card all he needs to clone it is a PC and a card reader. This is considered the most common fraud cause of all [10].
- **Internal fraud** - implies action of internal staff of the operators. Typically, operator employees with knowledge and access to the information systems, handle information in order to benefit a third party, for instance: giving free minutes, changing account settings.

2.2 Fraud detection

Over the years, telecommunications operators have developed or acquired technology in order to indentify fraud situations. This technology is based in a set of methods specifically designed to detect fraud. Some of the most common methods are:

– **High Usage** - measure the amount of traffic generated by each SIM card; detect SIM cards that generate high amounts of traffic in the operators' network.
– **Calls collision** - monitor the traffic in a time dimension for each SIM card; detect overlapped events generated by the same SIM card.
– **IMEI/IMSI stuffing** - mapping the SIM cards (IMSI) to the devices, cell phones for instance, they are used in (IMEI); detect devices that use several SIM cards.
– **Call velocity** - monitoring the traffic in a time and geographic dimensions for each SIM card; unlike the calls collision method, this method's gold is not to detect overlapped events, but physically impossible. For instance, the same SIM card has a call that ends at 2:00am in Braga and another one that starts at 2:10am in Faro (more than 600Km away). Despite the events are not overlapped, it is physically impossible to travel the distance in 10 minutes.
– **Ratio** - monitor the services used by each SIM card; detect SIM cards that use services (for instance: Voice, SMS, MMS, data) disproportionally.

These methods are implemented and thresholds are set by the operators' analysts. The methods are continually monitoring the network traffic and keep statistics for a given time period. If the threshold is reached, a possible fraud situation is detected. Then the analysts decide on each situation if this is a real fraud situation and what actions to take.

Still, the operators are struggling against a huge problem: even if a fraud is detected, all the damage has already been done. The methods to detect fraud that currently exist are reaction-based; they can only detect a fraud after it has already happen, avoiding further damage, but still supporting all the inherent damage to the situation.

Another major issue in the fraud detection systems is the lack of a knowledge base. Because of the complexity of the systems it is very complex to find a structure that stores all the fraud situations previously detected. This knowledge base could be very useful in order to the operators better understand the fraud attacks they suffer and retrieve valuable information.

2.3 Defining an approach

The new approach defined is complementary to actual fraud solution: it does not replace it, but uses the data processed by the fraud solution and the fraudsters detected by it in order to evolve the solution to a new level. The data processed will be used to build a profile for each subscriber, containing individual and behavior features. When a new fraudster is detected by the fraud solution, the profile will be used for: the identity features will be used to detect an attempt from the fraudster to re-enter the operator network; the behavior features will be used to detect other subscribers that have a similar behavior and therefore are likely to commit fraud.

In order to solve the problem previously detected that affects the telecommunications operators, the following goals were defined:

1. Define a method that monitors the operator network traffic and builds profiles based on the SIM cards actual usage. These profiles are composed by identity and behavior attributes.
2. Define a method that monitors the operator network traffic and detects fraud suspects based on a behavior comparison using the behavior attributes of the profiles of fraudsters previously detected by the fraud solution.
3. Define a method that monitors the operator network traffic and detects previously blocked fraudsters attempting to re-enter the network, based on an identity comparison using the identity attributes of the profiles of fraudsters previously detected by the fraud solution.
4. Define a knowledge base structure, capable to support information retrieved from all the known fraud cases.

Notice that, as it was mentioned before, this new approach is complementary to actual fraud solution, it does not replace it, but works cooperative with it in order to detect fraud suspects as soon as possible.

3 Solution proposal

The solution proposed to solve this problem is based in two main techniques: profiling and KDD (Knowledge Discovery in Data). These techniques will be implemented in a MAS (Multiagent System). After an initial section where the data structure that will be used as input for the solution is explained, the MAS proposal is presented, followed by a section where each agent of the MAS is detailed.

3.1 Data structure

In a telecommunications company the operator keeps record of every event processed by the system. These events are recorded in CDRs (Call Detail Records), generated automatically and are used for billing purposes. Each CDR has information regarding a set of events, voice calls or SMSs, for example. Typically, the CDR is a text file containing information structured by a pre-defined set of ordered fields separated by a pre-defined character. Each line of the CDR file is an event processed in the operators system. The structure of the CDR (the number of fields, the order of the fields and the separator character) is defined by the telecommunications company, so the CDR structure varies from operator to operator. However, there is a set of fields that, due to their importance, for billing and rating purposes, are usually common to all CDR structures:

- **A Number** - identifies the originator of the event;
- **B Number** - identifies the receiver of the event;
- **Event Date** - the date the event started;
- **Event Type** - identifies the type of the event, for example: 1 (Voice), 2 (SMS), 3 (MMS), 4 (Data);
- **Event Amount** - measure of the event, for example, in a voice call the event amount is 124 seconds, in a SMS the event amount is 45 characters;
- **Cell ID** - identifies the network cell that processed the event.

The information contained in the CDRs will be the input for all the future work. The study of the contents of CDRs is not a novelty. They were first created with billing purpose, but know they are used with different purposes of great importance to the operators, like discovering user communities [14], for instance.

3.2 Multiagent System

Distributed Artificial Intelligence (DAI) was a subfield of Artificial intelligence research dedicated to the development of distributed solutions for complex problems [2]. These days, DAI has been largely supplanted by the field of Multiagent Systems (MAS). The main purpose of the MAS is the study, construction and application of Multiagent Systems, that is, systems in which several interacting, intelligent agents pursue some set of goals and/or perform some set of tasks [15]. Some motivations that led to the development of the MAS: complexity and dimension of the problems; problems geographically and/or functionally distributed; information and knowledge disperse; multiple systems interconnection; parallelism, robustness and scalability.

The MAS implemented to support the solution proposal is a **closed MAS** [16][12], where the architecture design is static, with all the agents and functionalities pre-defined. In this closed MAS the agents communicate using a common language, each agent is developed as an expert in his functionality and they all work and cooperate together in order to achieve a main goal.

The coordination of the MAS is **cooperative** [11]: the agents do not compete; they cooperate in order to achieve a main objective. The organization is **flat** [11]: each agent is an expert in an area, there is no agent commanding other agents and all agents have the same importance. The communication between the agents is **direct** [11]: there is no agent or middleware between two agents communicating with each other, they communicate directly.

The following image illustrates the MAS architecture and the process flow between the MAS and the fraud solution.

As for the architecture, the MAS is composed by three agents:

- the **Profiling Agent** is responsible for integrating the CDR files and building the profiles (with identity and behavior features) for the subscribers;
- the **Detection Agent** is responsible for detecting new fraud suspects;
- the **KDD Agent** is responsible for processing the profiles of known fraudsters and enrich the knowledge database.

As for the process flow between the MAS and the fraud solution, the flow of information is represented by numerated arrows:

- 1 and 2 - input for MAS from the fraud solution;
- 3 and 4 - internal MAS information exchange;
- 5 and 6 - output of the MAS to the fraud solution.

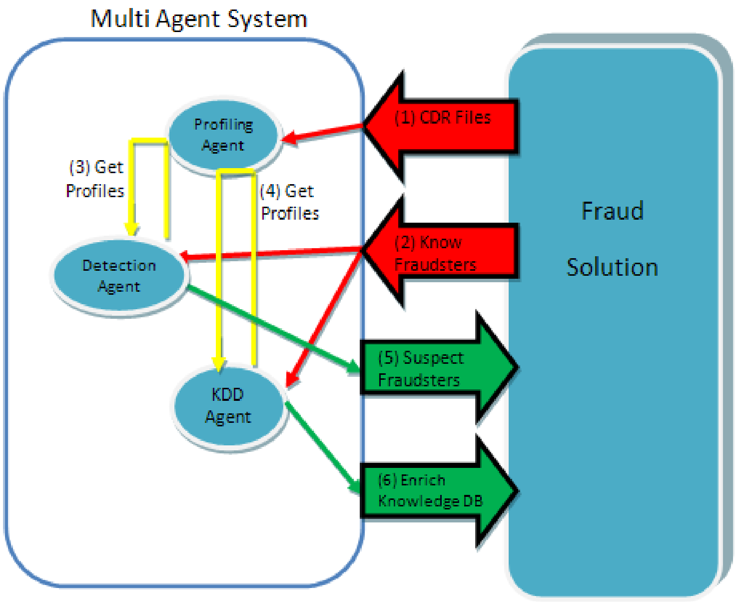


Fig. 1. MAS - Architecture and Process Flow

In the first phase, the fraud solution redirects all the CDR files contents for the MAS (**step 1** in the figure), where the profiling agent uses this content to build the profiles for the subscribers. In this phase the profiling agent should extract from the CDR files contents the necessary information to in order to enrich the identity and behavior features of profiles.

In a second stage, when the fraud solution detects a new fraudster, based on some of the methods previously explained, the fraud solution indicates the MAS that a new fraudster was detected (**step 2** in the figure). Then, this information is used by two agents with different purposes:

- the detection agent will use this information in order to retrieve from the profiling agent the profile of the fraudster and use the profile identity features to detect if the same subscriber tries to re-enter the operator network and the profile behavior features detect other subscribers that have a similar behavior; the fraud solution is then warned of the suspects that this agent detects (**step 3** in the figure).

- the KDD agent will use this information in order to retrieve from the profiling agent the profile of the fraudster and use the profile behavior features and enrich a knowledge database containing all the detected fraudsters profiles (only the behavior features) in order to try to retrieve significant information (patterns, similar behaviors) from this database. The results of this enrichment is then passed to the fraud solution (**step 4** in the figure), so that the fraud analysts have access to this information.

3.3 Profiling

Profiling is an auxiliary technique for criminal investigation. It fits in the Forensic Psychology domain. Profiling consists in a process of individual features inference, usually individuals responsible for criminal actions [4]. The profiling technique should be used as an extension of the criminal analysis [9], elaborating criminal profiles based on previous work. The mains idea to retain is: profiling complements previous work, it does not replace it.

Nowadays, profiling is a very used technique, implemented in police forces all around the world. Experts in this particular technique, like McCrary [7] or Wrightsman [17], point out the fact that the results of the use this technique make excellent prediction factors.

Even though the profiling technique was developed under the Forensic Psychology domain, it is a technique that is being used in the computer information systems, with various purposes. There are applications of profiling in order to detect social networks [6] [5], large scale behavior analysis [13], security applications [8] [18] and data transaction prediction models [3].

3.4 Applying profiling

Applying the profiling technique using the information contained in the CDRs, it is possible to build a profile of the operator users. The profile is divided in:

- **Behavior profile -** profiling about the user behavior in the network: which services the user uses and in which ratio, the periods of the day, week month he uses the services.
- **Identity Profile -** profiling about the user identity: the user's social network and the community he belongs to, geographical location.

For each network user, the profile should be continuously updated, and the evolution of the profile should also be recorded, because the evolution is by itself a profile input. As it was previously explained, the profiling technique should be used as an extension of the criminal analysis. The same applies in this case. The profiling does not substitutes the fraud detection methods previously explained; it uses and depends on them. The fraud detection methods are vital to detect the fraud cases and identify fraudster users. Only then, the profiling technique results will be used. Once detected a fraud situation and identified a fraudster, the fraudster profile will be used to:

– Enrich a knowledge database. Afterwards, it will be necessary to define methods that allow analysts retrieve important and relevant information from these databases. This information will give the operators the indicators regarding the fraud attacks they suffer. Who, when, where and how fraudsters attack the operators are questions that will have some answers thanks to this knowledge database.
– The identity profile of the fraudulent user should be saved and used to identify future attempts of the same user to use the network services.
– The behavior profile, including the evolution history, will be compared to other users, in order to detect users that are likely to commit fraud in a near future.

3.5 Agent-Based Knowledge Discovery in Data

In the past twenty years, agents and KDD (Knowledge Discovery in Data) have emerged separately as two prominent, dynamic and exciting research areas [19]. In recent years, an increasingly remarkable trend in both areas is that integrating agents into distributed data mining systems in order to extract knowledge from data in a faster, efficient way.

The KDD process consists in:

– **Data acquisition -** cleaning (remove noise and inconsistent data), integration (combining possible multiple sources), selection (decide which data is relevant) and transformation (transformation, consolidation and aggregation);
– **Data mining -** essential process where intelligent methods are applied in order to extract data patterns;
– **Pattern evaluation -** identify truly interesting patterns representing knowledge;
– **knowledge presentation -** visualization and knowledge representation.

The use of this KDD process to build the user profiles and enrich the knowledge database is due to the huge volume of the input data for this solution, the CDR files. The KDD process is, already, a scientific area with a huge range of applications and successful implementations, allowing the development of automatic procedures and autonomous agent reasoning (programming).

An agent-based approach to the resolution of this problem is a major step through because:

– the ability of an agent-based system to develop autonomous tasks is, also, very well funded in the Multiagent Systems community, which contributes to the soundness of the system to develop;
– the huge amount of data (CDR) available is incompatible with a human-only approach, but, on the other hand, is the first requirement to the use of KDD processes.

Applying the KDD process to the proposed solution: in the data acquisition phase CDR files are read and processed: remove useless information (Figure 1-(a)), select the relevant information and make the necessary transformations(Figure 1-(b)); it's in the data mining phase that the profiles will be built: necessary to define the methods to extract the profiles(Figure 1-(c)); in the pattern evaluation phase, the profiles will be analyzed and compared, in order to identify patterns (Figure 1-(d)); at last, in the knowledge presentation phase, the gold is to enrich the knowledge database (Figure 1-(e)), providing methods for the telecommunication operator analysts to retrieve important and useful information.

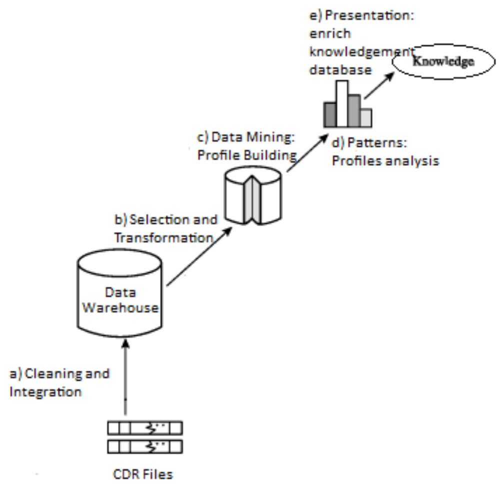


Fig. 2. KDD process

4 Conclusion

MAS applications are ideal in order to build system designed to solve complex problems, which could not be solved by an individual agent. In a MAS application it is possible to solve a complex problem using different methods, skills and knowledge, distributing information and resources by the different agents. By supporting the solution proposal in a MAS model, it is possible to take advantage of all these facts giving it flexibility and extensibility properties.

Profiling is a technique that is used in several areas with good results. By applying this technique in the fraud problem, telecommunications operators will be able to cut down their losses and also get a better understanding of the problem. Dividing the profile in two, behavior and identity, allows serving two different purposes: identify possible fraudsters based on previous fraudsters profiles and to detect previous fraudsters that are using the network services again. Maintaining a profile history is also important in order to understand the evolution of a fraudster in the network.

The knowledge database, even if it has no direct impact in the detection of fraud, is very important for operators to better understand the problem they are facing, and optimize their resources (computer systems, analysts, thresholds) on detecting fraud.

As for the input data that will be used to build the profiles, the CDRs, it must be taken into account their huge number and, consequently, the size of the data that will input the methods and structures defined for profiling purposes. This was the main reason that led to the use of the KDD technique.

References

1. Communications fraud control association. http://http://www.cfca.org.
2. A. Bond and L. Grasser. Readings in distributed artificial intelligence. San Mateo, USA, 1988. Morgan Kaufmann Publishers.
3. I. V. Cadez, P. Smyth, and H. Mannila. Probabilistic modeling of transaction data with applications to profiling, visualization, and prediction. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 37–46, New York, NY, USA, 2001. ACM.
4. E. CORREIA, S. LUCAS, and A. LAMIA. Profiling: Uma técnica auxiliar de investigação criminal. *Análise Psicológica*, 2007.
5. K. Dasgupta, R. Singh, B. Viswanathan, D. Chakraborty, S. Mukherjea, A. A. Nanavati, and A. Joshi. Social ties and their relevance to churn in mobile telecom networks. In *EDBT '08: Proceedings of the 11th international conference on Extending database technology*, pages 668–677, New York, NY, USA, 2008. ACM.
6. N. Du, B. Wu, X. Pei, B. Wang, and L. Xu. Community detection in large-scale social networks. In *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 16–25, New York, NY, USA, 2007. ACM.
7. G. McCrary. *Le profilage criminel à l'interieur et à l'extérieur du tribunal*. PUF, 2001.
8. J. McHugh, R. McLeod, and V. Nagaonkar. Passive network forensics: behavioural classification of network hosts based on connection patterns. *SIGOPS Oper. Syst. Rev.*, 42(3):99–111, 2008.
9. L. Montet. *Le profilage criminel*. PUF, 2002.
10. J. T. O'Brien. Telecommunications fraud. *The FBI Law Enforcement Bulletin*, 1998.
11. G. Olson, T. Malone, and J. Smith. Coordination theory and collaboration technology. Prentice Hall International Inc., 2001.
12. S. Russell and P. Norvig. Artificial intelligence, a modern approach. USA, 1995. Prentice Hall International Inc.
13. S. J. Stolfo, S. Hershkop, C.-W. Hu, W.-J. Li, O. Nimeskern, and K. Wang. Behavior-based modeling and its application to email analysis. *ACM Trans. Interet Technol.*, 6(2):187–221, 2006.
14. W.-G. Teng and M.-C. Chou. Mining communities of acquainted mobile users on call detail records. In *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*, pages 957–958, New York, NY, USA, 2007. ACM.

15. G. Weiss. Multiagent systems, a modern approach to distributed artificial intelligence. In *MIT Press*, USA, 1999.
16. M. Wooldrige. An introduction to multiagent systems. USA, 2002. John Wiley and Sons.
17. L. S. Wrighsman. *Forensic psychology.* Wadsworth, 2001.
18. K. Xu, Z.-L. Zhang, and S. Bhattacharyya. Internet traffic behavior profiling for network security monitoring. *IEEE/ACM Transactions on Networking*, 16(6):1241–1252, 2008.
19. Z. Zhu, W. Song, and J. Gu. A multi-agent and data mining model for tcm cases knowledge discovery. In *CCCM '08: Proceedings of the 2008 ISECS International Colloquium on Computing, Communication, Control, and Management*, pages 341–346, Washington, DC, USA, 2008. IEEE Computer Society.

Knowledge Acquisition and Intelligent Agency on the Web of Data

Elder Rizzon Santos[1], Rosa Maria Vicari[1] and Helder Coelho[2]

[1] Instituto de Informática - Universidade Federal do Rio Grande do Sul (UFRGS)
Po. B 15.064 - 91.501-970 - Porto Alegre - RS - Brazil.
[2] Departamento de Informática - Faculdade de Ciências da Universidade de Lisboa, Portugal
{ersantos, rosa}@inf.ufrgs.br, hcoelho@di.fc.ul.pt

Abstract. Using AgentSpeak-DL, we propose an agent model to autonomously query the web of data to gather additional knowledge related to the current set of beliefs. The knowledge acquisition mechanism uses additional assertions specified with the beliefs as inputs to construct the queries. The assertions come from a description logic approach to specify the agent belief base. Information from the queries is used to establish an ontology relating agent's beliefs to knowledge from the web of data. Main contribution of this research is the specification of a mechanism to enhance the agent's knowledge with semantic web techniques.

Keywords: web of data, knowledge-based agents, query specification.

1 Introduction

This paper describes a research on agent theory, focusing on autonomous agents capable of interacting on the semantic web. Our goal is to increase the availability of knowledge to the agent and, thus, the possible courses of action it can take. We consider agents as autonomous, intentional systems, modeled by the BDI agent theory [1]. This model formalizes how a rational agent go from its beliefs and desires to actions. Besides the interactions with its application specific environment, our agent interacts with the web of data as a medium to obtain knowledge.

The web of data, also known as the semantic web [2], can be viewed as an extension to the current web in which information receives a computational meaning. This makes the content adequate for processing by software applications, in contrast to traditional web where information is usually available in natural language and is adequate for human beings. As described in [2], the semantic web is a place where applications will consume and generate knowledge, ideally acting autonomously on behalf of the user. Following an application perspective, this work shows a possible way towards that vision.

One of the main results of the semantic web effort is its ontological knowledge representation language – OWL. We limit the agent to deal only with OWL – regarding knowledge obtained from the web, since it allows a better integration with the agent's belief base. In this case, the expressiveness and semantics allow the agent to construct queries and perform inferences using a concept hierarchy, restrictions and properties.

One of the key points to integrate agent and semantic web technology is located on their common ground, the knowledge representation. The approach used in this work is based on AgentSpeak(DL) [3], which modifies the original language [4], and its concretization in terms of agent architecture, JASDL [5].

The remaining part of the paper is organized as follows: section 2 relates the knowledge acquisition mechanism to the state-of-the-art in terms of agents integrated on the semantic web and in terms of tools needed to explore the web of data; in section 3 we specify the knowledge acquisition by detailing the query construction and the result processing - an example of utilization is presented; finally section 4 presents our concluding remarks and future work.

2 Related work

Regarding the use of the web of data to provide knowledge to intelligent applications, [6] provides an approach based on information theory. The goal is to allow the agents to naturally deal with concept alignment, uncertainty and utilization of services. Information theory is applied to provide quantifications for trust, reputation and reliability to the information being considered. Negotiation is a fundamental part of the model; it is the process that the agent undergoes to exchange information with peers and services – as a provider and as a subscriber.

Every reasoning process is contextualized and affected by the set of norms and contracts that the agent currently complies with. Context represents previous agreements, previous illocutions or any code that aligns the ontology between the peers in order to interpret an action. This agency model can be summarized as a utilitarian approach to knowledge based agents. The agent is always part of an information exchange – a negotiation, and all its reasoning and actions are designed to maximize the outcome.

The knowledge acquisition mechanism described in this paper differs from [6] in the theoretical and philosophical views of agency and, consequently, in the concretization of the model. Our goal is to provide relevant knowledge related to agent's beliefs, so that the agent is able to perform reasoning with this related knowledge according to its domain. One might argue that the utility approach, as attested by [6] can be used to model and provide a solution to our problem.

We consider this point of view but this line regards agency as a decision problem, and we view agents as a software that displays intelligent behaviour in terms of flexible autonomous action. As described by [7], flexibility is composed by pro-activeness (to take the initiative to reach its goals), reactiveness (to perceive and timely react to environment changes) and social ability (the interaction with other agents and perhaps with humans too). We are aware of the difficulties that come along with this vision but we still consider it as the approach to agency to be followed on this research.

On the subject of agency and the semantic web, AgentSpeak-DL and JASDL are also taken into consideration. In summary, AgentSpeak-DL [3] provides the formal semantics for AgentSpeak with Description Logics. The authors state that the main results are more expressive queries to the belief base, a refinement of the belief update process, more options for plan retrieval and native support for agent communication. JASDL provides the implementation of the concepts described in [3] on the Jason [5] framework. JASDL (Jason AgentSpeak-Description Logic) extends the Jason platform providing agent-oriented programming with ontological reasoning. In JASDL each ontology that the agent is aware of receives a label that is utilized to provide semantic annotations in the form of "semantic enriched" literals – a literal that corresponds to an axiom of an ontology.

This mechanism extends the traditional belief base (formed by a list of ground literals), allowing it to reside partially in the ABox of an ontology. Following this idea, the belief base is specified by two knowledge representations (OWL and standard AgentSpeak). These modifications affect important reasoning, namely belief revision and option selection (plan searching). The impacts on the agent life cycle and implementation decisions are discussed in detail in [5].

Our work was implemented with JASDL mainly due to its associated agent theory, integration of OWL to the reasoning and also because of its extensibility mechanism. In this framework, it is straightforward to modify any aspect of the agent reasoning cycle. The ease to customize the stages of reasoning is paralleled to the complexity inherent to the design of an agent theory.

When customizing the reasoning, the developer must be aware of the impact that the modification has on the life cycle and on the compliance to the BDI theory. Mindful of the complexities involved in the customization, we adhere to JASDL's approach since it supports the utilization of knowledge acquired on runtime to perform meta-level reasoning and further belief functionality (future work).

Still on agent architectures and models for the semantic web, there is the Nuin agent architecture. Its goal is to provide a "practical architecture for deliberative agents for the Semantic Web" [8]. The architecture is based on AgentSpeak(L) to provide the deliberation mechanism and the definition of an agent through an RDF model. Nuin provides a general architecture, leaving to the agent developer the decision on the knowledge representation and reasoning.

A scripting language is available to define agent's plans using terms of an ontology referenced by its URI (Uniform Resource Identifier). AgentSpeak's events abstraction is also treated as implemented by the developer. The interpreter provided by the architecture does not modify the original AgentSpeak interpreter in any manner. An important aspect of Nuin is that it is developed to be easily deployed and extended; it was implemented using software engineering design patterns. In theory, the agent architecture could be used with any knowledge representation, not only description logics.

The idea of allowing any kind of knowledge representation to be pluggable into agent architecture is very tempting but the theoretical foundations and the impact of such endeavor must also be considered. A key issue that arises is that inference services are limited only to query and to update the knowledge base. Important functionalities from each specific reasoning engine are ignored. If we consider OWL, for example, the agent is not aware of subsumption, generalization and specific TBox and ABox reasoning capabilities.

Functionalities such as ontology integration, alignment and modularity cannot be integrated to the architecture either. Regarding Nuin, it could be used in our work as concrete agent architecture but it considers OWL and its particularities superficially, without a clear account of the implications to the agent reasoning.

Another stream of works under automated uses of the web of data focuses on the delivery of knowledge to end users. Tripleshop [9] handles user queries and process the results according to a set of constraints and the application of pre-defined reasoning tasks. The query construction mechanism aids the user on the specification of SPARQL queries, allowing the user to define the set of URIs to be queried. A preliminary set of URI's is given to the user considering the SELECT and WHERE clauses. The query is executed on the Swoogle search engine and the results, after processing, establish a workable dataset for the user.

Swoogle [10] is one of the first search engines designed for the semantic web. It discovers and indexes RDF documents following the classic web search approach. Google is used to crawl the web and discover RDF and OWL files. These files are ranked with two custom algorithms built on top of an abstract model defining how an agent accesses the semantic web. The search engine can be accessed by a regular web site and also through a SPARQL endpoint.

Watson [11] is considered to be a gateway for semantic web data since it takes into account semantic details and semantic quality of the indexed data. The semantic details considered in the Watson architecture are related to OWL and RDF constructs, which are used to process and relate different documents. Quality of the semantic data is assessed in terms of expressivity, language, level of axiomatization, and measures of concepts and individual quantities. Sources for the Watson crawler can be diversified through a plug-in model. The gateway can be accessed through a web site and also through SPARQL. In [12] semantic web applications that use Watson services are presented. Our mechanism uses Watson as the main search engine, due to the wider availability of services and to Swoogle as a secondary SPARQL endpoint.

3 Acquiring knowledge from the web of data

The knowledge acquisition process here described has the ambitious goal to allow the agent to gather additional contextual information for its beliefs. Obtaining contextual information from the semantic web or from the web itself does not guarantee the validity and truthfulness of it. At the same time one cannot disregard this kind of knowledge that is also considered by some cognitive scientists as part of human cognition [13]. Thus, mechanisms to perceive the agent's context and to select the most appropriate knowledge to be considered must be integrated to the reasoning.

Moreover, we see context as an important factor to help on the balance between reactivity and pro-activeness. In this section, we describe a first step on this direction, with the construction of a related knowledge ontology. Considering the JASDL framework, descriptive knowledge is applied during the plan selection stage of the practical reasoning cycle. As presented in [5], the idea is to use the subsumption inference so that specific plans can be applied on more general situations. Thus, increasing the range of possibilities through a relaxation of the constraints during the selection.

Our approach to knowledge acquisition is focused on semantically enhanced beliefs [5] (beliefs associated to a concept defined in an ontology). We divide the knowledge acquisition process into two stages: (1) query construction and execution and (2) result processing.

Query construction

Most of the search engines that handles web of data content (OWL, RDF and RDF-S) provides two forms of queries: keyword search and SPARQL search. Keyword search comes from traditional web search mechanisms and provides a simple string matching. SPARQL [14], on the other hand, is a W3C recommendation for querying the web of data. It is a query language for RDF, allowing queries to OWL content due to the possible translation from OWL to RDF.

Considering OWL-DL – a variant of the description Logic SHOIN(D) [15], the construction and execution of SPARQL to query this content seems like a viable first step but not an ideal one. The main reason for that is that under RDF, there is no clear distinction between schema and data, and in OWL-DL such distinction (T-Box and A-Box) is clear and important to provide more expressiveness to the representation language. This situation reflects the ongoing research on the semantic web and will possibly be followed by standards designed specifically to OWL and its variations (Lite, DL, Full, OWL 2…).

Taking into account this current state of query in the web of data, we developed a mechanism compatible with such standards, which allows the direct use with current web of data repositories. We use the current set of beliefs as inputs to construct SPARQL queries. The results are processed according to OWL semantics to construct a related knowledge ontology.

Since we are limiting our approach to deal with OWL knowledge, we are bound to the limitations of terminological and assertive knowledge. Early works on description logics [16] [17] provide an in-depth view on the complexities and trade-offs of working with this kind of logic. In this work, we view the web of data and description logics as tools to be used by autonomous agents.

We adopted AgentSpeak-DL as the agent language and JASDL as the agent architecture to provide a proof of concept implementation of our work. Both the language and the architecture were developed having an integration of OWL and semantic web to agents in mind. JASDL allows the definition of "semantically enhanced" beliefs, which are beliefs associated with an ontology. It is possible to relate beliefs to class assertions, object and data property relations, and the all different axioms [5]. These relations, except for the all different axiom – used to declare that all individuals from a specified set are different from each other, are the inputs of our mechanism.

Given a belief *B* semantically enhanced by an assertion *A A* represents a set of possible assertions that can be applied to beliefs. In the case of JASDL, it can be Class Assertion (*CA*), Object Property Assertion (*O*) and Data Property Assertion (*D*).

Let *Sub* be the set of sub-classes of *CA* and *Sup* the set of super-classes of *CA*. Each *Sub* and super class of *CA* in itself is already knowledge related to *B* and is considered under the plan selection of the reasoning cycle. In our mechanism, *Sub* is utilized as a term for basic keyword matching search and as a restriction under SPARQL queries. A more restrictive query is formed by using group graph pattern match where all the variables in the query pattern must be bound for every solution.

A less restrictive query is constructed with pattern matching of alternatives. It is worth to note that in OWL-DL concept's relations (object or data) are coded with the *subClassOf* construct. Thus, queries based on class assertions also take into account possible relations of the considered belief *B*. According to this description the possible SPARQL queries are generated as follows:

1. Query using the set sub and applying restrictions in the same way, as defined in *CA*. This query is more precise, and should be used by the agent to further specialize its knowledge. The expected results should provide more information about a previously established context, and less information about different contexts. We adopted the FILTER modifier in order to allow similar names to the result set. A simple modification of the string (to remove the ^ character) provides exact matches. In the following parameter "i" is used to specify case insentiveness.

```
SELECT DISTINCT ind, type
WHERE{
      ?ind rdf:subClassOf ?sub_0 . FILTER regex(?sub_0,
       "^Sub[0]", "i")
      ?ind rdf:subClassOf ?sub_1 . FILTER regex(?sub_1,
       "^Sub[1]", "i")
      …
      ?ind rdf:subClassOf ?sub_n . FILTER regex(?sub_n,
       "^Sub[n]", "i")
      ?ind rdf:type ?type .}
```

2. Query using the set sub and applying optional restrictions regarding *CA* definition. In opposition to the previous query, this one allow a broader scope of results, useful to provide different views about a subject, gathering different contexts. When this pattern is applied, the finding of one match already yields a possible solution.

```
SELECT DISTINCT ind, type
WHERE{
      ?x rdf:subClassOf ?sub_0 . FILTER regex(?sub_0,
       "^Sub[0]", "i")
UNION {?x rdf:subClassOf ?sub_1 . FILTER regex(?sub_1,
       "^Sub[1]", "i") }
…
UNION {?x rdf:subClassOf ?sub_n . FILTER regex(?sub_n,
       "^Sub[n]", "i") }
      ?ind rdf:type ?type .}
```

A belief B, defined by an object property assertion *O* or a data property assertion *D*, is used by the query mechanism to gather different uses of the properties, namely concepts that are subclasses of the property and individuals, when available.

3. Query using *O* and *D* to gather individuals that use the properties. This knowledge gives a general view of how the property is used on different domains. In the agent perspective, this query also provides a broad view of the context but focused on the usage of a specific property. We demonstrate in this query the possibility to gather more knowledge using the utilization domain of the property (*_:R owl:someValuesFrom …*). This point will be deeply explored on future work. Another point presented is the possibility to acquire individuals that are sub classes of the individual that complies to the query *(?wholvl2 rdfs:subClassOf ?who*). In this case, we can also apply filters to give more flexibility to string matching.

```
SELECT DISTINCT ?who ?type ?wholvl2
WHERE {
       ?who rdfs:subClassOf _:RST .
       ?wholvl2 rdfs:subClassOf ?who .
       _:RST a owl:Restriction .
       _:RST owl:onProperty [O or D].
       _:RST owl:someValuesFrom ?type . }
```

Result Processing

From the results obtained with the queries, we propose the automatic elaboration of a related beliefs ontology. Clearly, the results can yield a number of different possibilities for inference that are domain specific. In terms of software engineering, the queries and the ontology construction are defined as plans, which can be reused and modified as required by each application. Another possibility would be to specify the procedures as Jason's internal actions but then we would loose the declarative aspect of agent design.

The automatic generated ontology relates beliefs to their respective related knowledge. So far, we simply developed a process to build an ontological structure, disregarding similarities and consistency verifications with the belief base. This constitutes a future work, where similarity measures are applied to the related knowledge, establishing a properly analyzed ontology. Consistency with the belief base is not fundamental, since we are dealing with related knowledge, which may contradict the agent's beliefs. Similarity measure will tend to 0 in such cases.

Another approach to deal with this situation is to consider the related knowledge as contextual information to be used by the agent when his standard set of actions fails. Learning mechanisms can be employed to evaluate new solutions based on newly acquired context information. Later, the evaluation's results can be integrated to the belief revision function, going towards automatic learning based on semantic web information. This is a complex problem and requires specific research to handle its particularities. Nevertheless, the verification of this hypothesis is part of our ongoing work.

Queries that obtain knowledge from class assertions (1 and 2) return individuals and their respective types. This information is expressed in the ontology by a property relation and by an individual instantiation. We add a *hasRelatedClassAssertion(type[0..n])* property to belief *B* and instantiate the individual *ind[n]* as a subclass of *type[n]*. *type[0..n]* and *ind[0..n]* are the results from the SPARQL queries. In JASDL, this is achieved by using the following code:

```
jasdl.ia.define_class(request, “relatedonto:type[n]”);
// creates the type[n] class
+hasRelatedClassAssertion(B, type[n]);
+type[n](ind[n]);
?request(X)[o(self)]; // test goal used to provide the
unification of request and the individuals.
```

Queries that obtain knowledge from property relations (3) return individuals and their type. On (3) we showed how to gather further knowledge descending one more level on the hierarchy. Here we will not add such knowledge since the process remains the same, that is, to add only one more property relation to the belief. Thus belief *B* will have a property *hasRelatedUsage(?type[n])*, and an individual labeled *ind[n]* will be instantiated as a subclass of *hasRelatedUsage(?type[n])* and *B*. Similarly to the previous JASDL code fragment, such definition can be implemented as follows:

```
jasdl.ia.define_class(request, “relatedonto:type[n]”);
// creates the type[n] class
+hasRelatedUsage(B, type[n]);
+type[n](ind[n]);
+hasRelatedUsage(ind[n]);
?request(X)[o(self)];
```

It is worth noting that the resulting ontology is rather simple, and constitutes only an one level hierarchy with the respective individuals and their URIs. This is a preliminary result that will be developed to more expressive ontologies through the use of restrictions, axioms and OWL inference.

Example

To exemplify the process we will show code excerpts from our proof of concept agent and one of its domain ontologies. This agent has a simple task, which is to maintain the profile of a learner by following the IMS Learner Information Package standard [18]. A simplified version of IMS-LIP was modeled in an OWL ontology, focusing on the most important concepts of our application domain. Refer to [19][20] for further contextualization of the domain. Figure 1 shows the hierarchy of concepts from the IMS-LIP ontology.

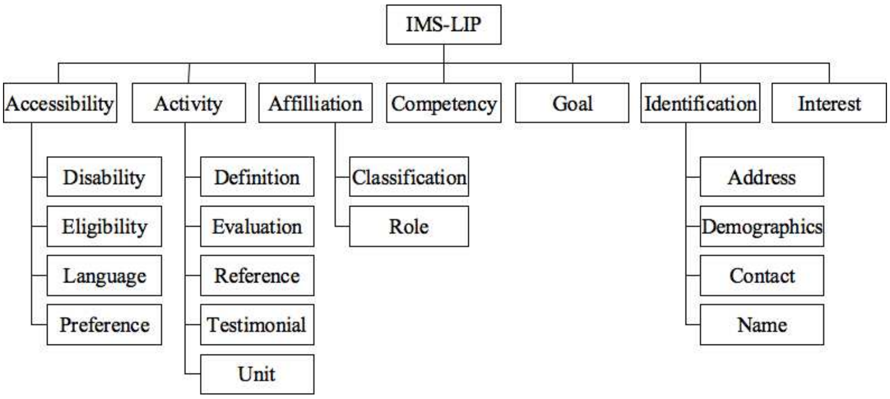


Figure 1. Hierarchy of concepts from the agent's domain ontology (IMS-LIP).

Figure 2 illustrates the ontology's utilization to specify a set of individuals (diamonds) that represent learner's accessibility details. Both object and data property relations are represented by dotted arrows, together with their name. Diamonds represent individuals, and rectangles represent data (strings, numbers, dates, etc.).

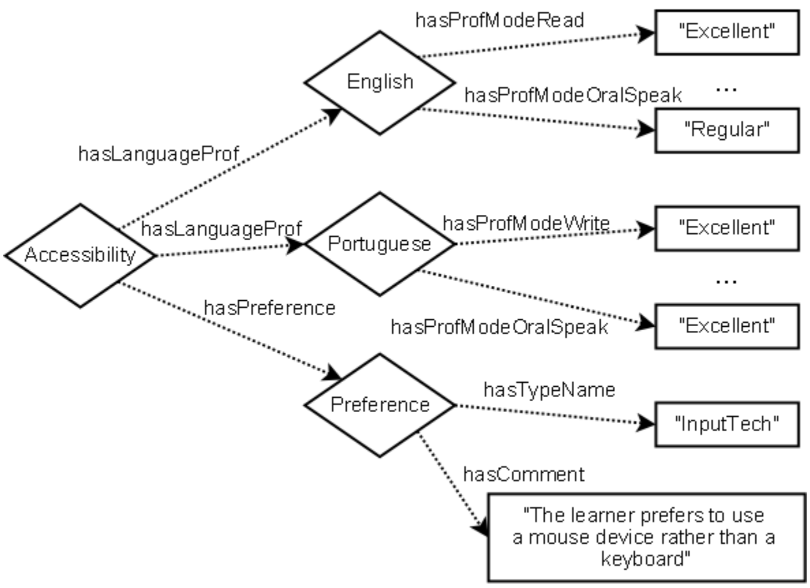


Figure 2. Individuals representing accessibility details.

The learner profile is managed by a plan, which modifies the belief base (which represents the profile) according to perceptions from the environment. We will not detail the perception mechanism since our focus is to show how beliefs can generate the queries during the knowledge acquisition process. Next we show a commented plan fragment that explains how semantically enhanced beliefs are created in JASDL, considering IMS-LIP as the domain ontology.

```
…
+goal(goal_A)[o(imslip)]; // defines the belief with a
                             class assertion
+hasDescription(goal_A, "The learner must study aspects
of trigonometry") [o(imslip)]; // defines the belief as
                              a data property assertion
+hasDate(goal_A, "2009:04:15") [o(imslip)]; //another
    data property assertion. The annotation [o(imslip)]
    specifies the respective ontology
+hasGoal(goal_A, goal_B) [o(imslip)]; //an object
 property assertion stating a hierarchy between beliefs
…
```

Figure 3 illustrates the resulting individual Goal A (a complete version), which represents a fragment of the belief-based leaner profile.

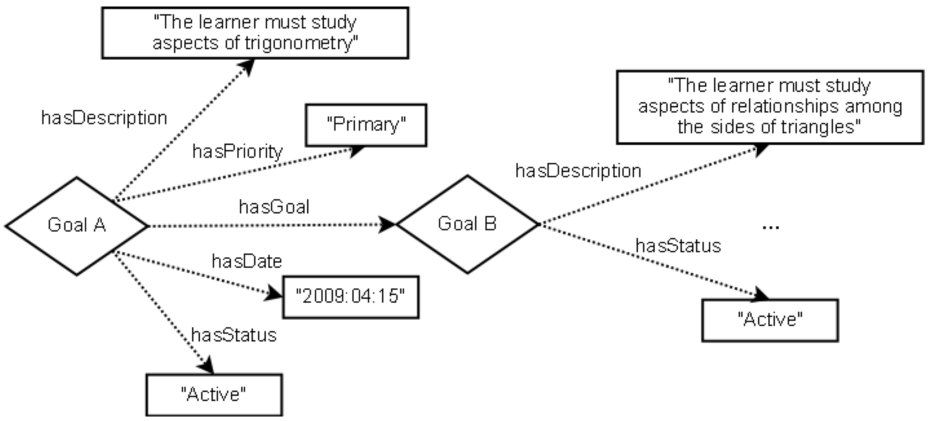


Figure 3. Individual Goal A resulting from the agent's belief base.

The one class assertion based query from the beliefs shown above (goal_A) is:

```
SELECT DISTINCT ind, type
WHERE{
    ?ind rdf:subClassOf ?goal .
     FILTER regex(?sub_0,"^goal", "i")
    ?ind rdf:type ?type .}
```

This example demonstrated the application of our approach to a practical problem. It is still limited to technical issues, focused on modeling the agent, and further evaluation of the approach must be performed.

4 Conclusion

This paper presented the preliminary results of an ongoing research on autonomy and intelligent behaviour on the semantic web. The main result is a knowledge acquisition mechanism that uses the agent's beliefs as inputs to gather knowledge related to them from the web of data. This is achieved by the automatic construction of SPARQL queries and the processing of the resulting dataset to construct a related knowledge ontology.

Relating the work to the current state-of-the-art agents and the semantic web, our main contribution is the specification of how an autonomous agent might use its view of the world (beliefs) to enhance its own knowledge. How this knowledge is used is highly dependent on the application domain. Nonetheless, we provided a simple process to store the knowledge in a compatible and extensible way. Our specification of the mechanism attempted to be as practical as possible, focusing on agent oriented programming aspects contextualized by our point of view about agent theories.

Future work based on this research has several directions. The most straightforward is to further develop the acquisition mechanism. It can be extended by providing more query constructions, focusing on restrictions and axioms and exploring more hierarchy levels. In terms of restrictions we showed one query that considers them superficially but SPARQL is expressible enough to provide richer queries. On the axioms side, we still need to explore disjoint assertions, and there is the possibility to integrate the mechanism to the rules. The same happens with the result processing side, since as the queries become more expressive, the ontology construction must follow the modifications. Next we aim to apply the concept similarity to provide a measurement for the automatically constructed ontology.

Building on the acquisition mechanism, we will investigate the impact of using related knowledge on the agent's practical reasoning. Specifically, it is possible to define belief revision functions that consider knowledge from external sources, using probabilistic functions, user context, pre-defined ontologies, and so on. Our line of work guides us towards mechanisms that use the web of data as a leverage to aid the agent with its tasks but also restricts it so that a balance between pro-activeness and reactivity is achieved.

Finally, a higher goal of this research is to study the interaction among meta-level reasoning and practical reasoning. We envision meta-level reasoning as a creative based process, applying concept blending strategies [21] to the belief base together with knowledge acquired from the web of data. There is only a few partial formalizations of the theory. The work described in [22] is one of the latest approaches and serves as an inspiration to our research.

Acknowledgments.
The authors would like to thank CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and CNPq (National Council for Scientific and Technological Development) for the scholarship that supported the development of this research and also the reviewers for their insightful comments.

References

1. Bratman, M. E; Israel, D. J.; Pollack, M. E. Plans and resource-bounded practical reasoning. J. Computational Intelligence, v. 4, pg 349-355, 1988.
2. Berners Lee, T.; Hendler, J.; Lassila, O. The Semantic Web. Scientific American, v. 284, n.5, p. 35-43, 2001.

3. AgentSpeak (DL). Moreira A. F. et al. Agent-Oriented Programming with Underlying Ontological Reasoning. In: Proceedings of DALT 2005, p.155–170.
4. Rao, A.S. AgentSpeak(L): BDI agents speak out in a logical computable language. In W. Van de Velde and J. Perram (eds), Proc. Seventh Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96), Eindhoven, The Netherlands, number 1038 in LNAI, pages 42–55. Springer-Verlag, 1996.
5. JASDL. Klapiscak T., Bordini R. JASDL: A Practical Programming Approach Combining Agent and Semantic Web Technologies. In: Declarative Agent Languages and Technologies VI, 6th International Workshop, DALT 2008, Revised Selected and Invited Papers. LNAI, v. 5397.
6. Debenham, J.K.; Sierra, C. Merging intelligent agency and the Semantic Web. J. Knowledge-Based Systems, v. 21, n. 3, pg 184-191, Elsevier, 2008.
7. Wooldridge M. Intelligent Agents. In: Weiss G. Multiagent Systems, The MIT Press, April 1999.
8. Dickinson I., Wooldridge M. Towards practical reasoning agents for the semantic web. In: Proceedings of AAMAS '03. New York, NY, USA. ACM, p.827–834, 2003.
9. Finin, T.W; Sachs, J.; Parr, C.S. Finding Data, Knowledge, and Answers on the Semantic Web. In: Proceedings of FLAIRS 2007, AAAI, pg. 2-7, 2007.
10. Ding, L.; Finin, T.; Joshi, A.; Pan, R.; Cost, R. S.; Peng Y.; Reddivari, P.; Doshi, V. C.; Sachs, J. Swoogle: A Search and Metadata Engine for the Semantic Web. In: Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management, ACM, 2004.
11. D'Aquin, M. et al. Toward a New Generation of Semantic Web Applications. J. IEEE Intelligent Systems, v. 23, n. 3, pg. 20-28, IEEE, 2008.
12. D'Aquin, M. et al. What Can be Done with the Semantic Web? An Overview Watson-based Applications. In: Proceedings of SWAP 2008. CEUR Workshop Proceedings, v. 426, 2008.
13. Sutton, J. Distributed Cognition: domains and dimensions. In: Pragmatics & Cognition, chapter 3, pg. 45-56. Eds. Dror, I. E.; Harnad, S; 2008.
14. Prud'hommeaux, E.; Seaborne, A. SPARQL Query Language for RDF. (January, 2008) http://www.w3.org/TR/rdf-sparql-query/
15. Horrocks, I.; Sattler, U. A Tableaux Decision Procedure for SHOIQ. In: Proceedings of IJCAI 2005, pg. 448-453.
16. Doyle, J.; Patil, R. S. Two theses of knowledge representation: language restrictions, taxonomic classification, and the utility of representation services. In: Artificial Intelligence, 1991, v. 48, n. 3, pg. 261-197, Elsevier Science Publishers.
17. Donini, F. M.; Lenzerini, M.; Nardi, D.; Nutt, W. Tractable Concept Languages. In: Proceedings of IJCAI 1991, pg. 458-465.
18. IMS Global Learning Consortium. IMS Learner Information Package. (2005) http://www.imsglobal.org/profiles/lipv1p0p1/imslip_sumcv1p0p1.html
19. Baez, M. et. Al. OBAA project: An approach to interoperable learning objects based on Web and digital television. In: Proceedings of the 9th WCCE, 2009. To appear.
20. Boff, E., Santos, E.R., Vicari, R.M.: Social agents to improve collaboration on an educational portal. In: IEEE International Conference on Advanced Learning Technologies, IEEE Computer Society (2006) 896–900.
21. Fauconnier, G., Turner, M. The Way We Think: Conceptual Blending and the Mind's Hidden Complexities. Perseus Publishing, 2003.
22. Pereira, F.C. Creativity and AI: A Conceptual Blending Approach. Mouton de Gruyter, Berlin, 2007.

A Study of Agents with Self-awareness for Collaborative Behavior

Sheng-Wen Wang[1], Chuen-Tsai Sun[1], Chung-Yuan Huang[2]

[1] Department of Computer Science
National Chiao Tung University
1001 Ta-Hsueh Road, Hsinchu 300, Taiwan
sww0915@gmail.com, ctsun@cs.nctu.edu.tw

[2] Department of Computer Science and Information Engineering
Chang Gung University
259 Wen-Hwa 1st Road Taoyuan 333, Taiwan, Republic of China
gscott@mail.cgu.edu.tw

Abstract. In the history of artificial intelligence (AI), primary agent focuses have been external environments, outside incentives, and behavioral responses. Internal operation mechanisms (i.e., attending to the self in the same manner as human self-awareness) have never been a concern for AI agent. We propose to address this core AI issue by proposing a novel agent cognitive learning model (ACLM) having similarities with human self-awareness, and to apply the proposed model to the Iterative Prisoner's Dilemma (IPD) in cellular Automata networks. Our goal is to show the ability of a cognitive learning model to improve intelligent agent performance and support collaborative agent behavior. We believe additional simulations and analyses will indicate enriched social benefits, even in cases where only a few agents achieve limited self-awareness capabilities.

Keywords: Agent, self-awareness, Iterative Prisoner's Dilemma.

1 Introduction

The term self-awareness refers to experiences in which an individual's attention is pointed to the self [1]. Eastern and western philosophers and psychologists have studied the self-concept for many years [2-5] and have made self-awareness a central issue in cognitive science and educational psychology [6, 7].

Simulations and artificial societies are being used to develop and test Artificial Intelligence (AI) learning models (e.g., machine learning, neural networks, and evolutionary computing), to mimic human cognitive and behavioral models, and to establish intelligent agents [8-10]. However, most models offered to date focus on outer environments rather than inner operations, with some addressing the relationship between outside incentives and behavioral responses. Our research plan is to analyze the benefits of self-awareness mechanisms for AI agents.

Our goal is to refine and introduce an agent-cognition learning mechanism (ACLM) to overcome deficiencies in traditional AI learning approaches that emphasize self-schema for internal learning. Furthermore, we will address the artificial societal conflict between the public good and private interests resulting from agent environments and goals when proposing an agent self-awareness model that is consistent with cognitive learning models. Finally, we will discuss how self-awareness resolves the problem of collective irrational behaviors and establish model validity and stability via analyses of individual performances and collaborative behaviors.

2 Agent Cognition Learning Model (ACLM)

We use the world model Learning — putting the learning focus (Attention) at the outer environment to discuss the inadequacy of Russell's [11] general model of learning agents. Doing so requires addressing the importance of self-learning in order to narrow the gap between AI agent and human intelligence. Our proposed cognition learning model is based on using self-schema as an agent's internal learning focus, which can be compatible with existing agent systems. According to our proposed model, agents attend to both their world model and self-schema; achieving inner learning via self-schema awareness moves agents closer to human intelligence. The model also offers a unique design concept to solve the high-level intelligence challenges that agents based on the world model are incapable of solving.

2.1 The Proposed Model: ACLM

We modified our design concept as a result of our analysis of the world model, using Russell's general agent model to propose a new agent cognition learning model composed of three elements: performance, world model learning, and self-schema cognition (Fig. 1). The performance element is responsible for selecting external actions. The world model learning element is in charge of integrating traditional learning components (whose focus is limited to external environments) in order to improve learning efficiency. World model learning requires knowledge about the learning element and feedback on agent performance, which it uses to determine how the performance element should be modified for better performance in the future. The self-schema cognition element (which uses prior experiences to add information to a knowledge structure) can help agents understand, explain, and predict self-behavior. Our model supports coordination between world model learning and self-schema cognition to present the most favorable method for improving performance. Agents eventually possess both external and internal learning concepts. According to our proposed ACLM, agents will be capable of self-discovery and self-awareness via the addition of various schemas that can improve and promote efficiency by means of co-ordination between external learning and internal cognition, thereby moving closer toward a human intelligence model.

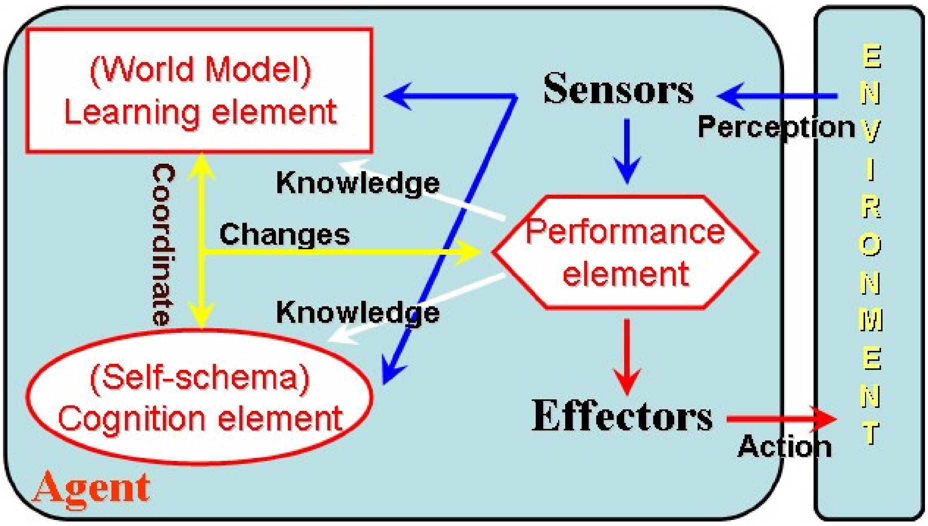


Fig. 1. Agent cognition learning model.

3 Experimental Design

The environment that any agent exists in will have many other agents, therefore the designer of a specific agent must refrain from dominating resources or profits in a manner that causes harm to the overall agent population. In response to this conflict between collective and individual agent goals, we propose an agent learning model in which the superego focuses on self-awareness achievement, based on our belief that any agent who owns self-awareness can make its life better by acting on its private interest, which in turn will benefit other agents in the form of cooperative behavior. This rational behavior has been observed among IPD strategic agents, therefore for our research platform we adopted an IPD environment with social networks that correspond to our experiment is aimed at observing the acts of learning agents with self-awareness and the effects of those actions on performance results.

3.1 Simulation Model

The simulation model shown in Figure 2 uses the two-layer concept, in which the combination of the IPD game and social networks serves as the research platform. The upper layer is the IPD (adopting the evolutionary computing approach) and the lower layer consists of the cellular automata social networks. Each upper agent adopts a pure strategy—that is, it uses the same policy for all coworkers. Besides, the Memory-1 deterministic strategy on its memory ability, there are 16 strategies can be chosen. To support observations of the emergent behaviours of strategic agents, each agent has its own unique colour. For lower-layer social networks, the cellular automata creation method made use of 2-D spatial relations in which each agent establishes links with its adjacent cells. When those links extend k steps it is called a “radius-k neighbourhood” consisting of surrounding coworkers. Subsequently, the radius-k neighbourhood of any agent can be modified by breaking off a fraction of its original links. This creates an equal number of new links (shortcuts) and randomly adds to the neighbourhood a set of individuals taken from the entire system.

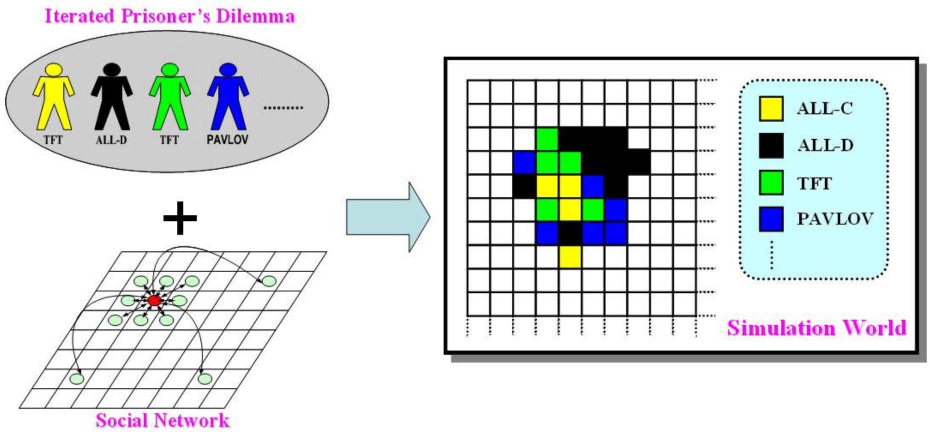


Fig. 2. Simulation model.

3.2 Agent Self-Awareness Model Using Superego Idea

Based on an analysis of intelligent agent and learning agent personalities using Freud’s Three Components of Personality, we concluded that they do not have what we would consider ids or superegos. If an agent did in fact have a superego, it would support an understanding of societal expectations and the earlier emergence of collaborative behaviour. We therefore view superego as an awareness goal to resolve conflicts between collective and private interests in artificial societies. We therefore adjusted the personality model for agents in our proposed ACLM in favor of a learning model that regards the superego as a self-aware goal according to the concepts of external learning and internal cognition—in other words, to add the self-schema cognition element to the ACLM.

As shown in Figure 3, our version of superego awareness consists of four sequential steps: self-observation, self-recognition and social expectation analysis, rational calculation, and self-adjustment. To test our idea we established an experimental model using the superego awareness unit and a control unit that go through an elementary evolutionary process (Fig. 4). The experiment consisted of eleven steps:

1. Establish environmental parameters (e.g., strategy color maps, social network parameters, interaction rules) and evolutionary parameters (e.g., population size, selection rules, mutation rate and rules, crossover rate and rules).
2. Randomly generate populations and establish two kinds of social networks.
3. Select coworkers.
4. Calculate fitness scores with coworkers.
5. Use evaluation rules to give reputations to coworkers.
6. If any coworkers have not yet been selected, go to step 3. Once all coworkers have been selected, go to step 7.
7. Collect agent recognition information from coworkers. (reputation)
8. Perform social expectation analysis to determine what coworkers expect from agents (social expectations).
9. Use rational calculations to determine the degree of matching between reputation and social expectations. If below an established threshold, do nothing; otherwise perform self-adjustment.
10. Use self-adjustment procedure to select a suitable social expectation strategy.
11. Select candidate agents for the next generation and reset reputation and expectation values to zero.

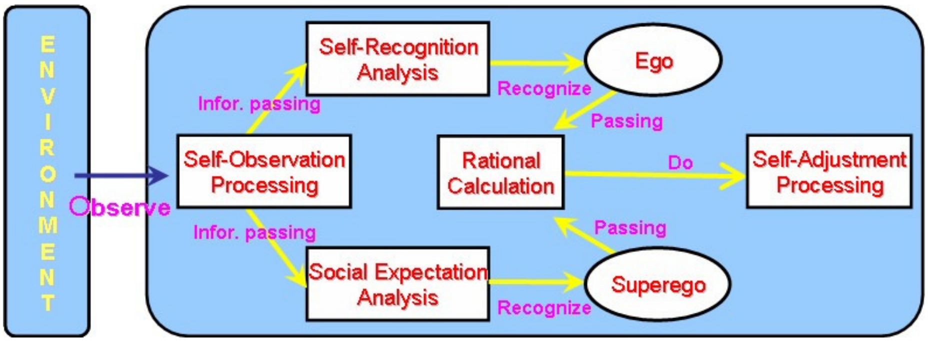


Fig. 3. Agent self-awareness model.

4 Results

We used cellular automata social networks in our experiments. In social networks, the control group is the elementary evolutionary IPD model (no self- aware agents), and the experimental group had self-aware agents in the simulated environment) at ratios of 1.0, 0.5, 0.3, and 0.1 (i.e., a ratio of 1.0 means that all agents are self-aware).

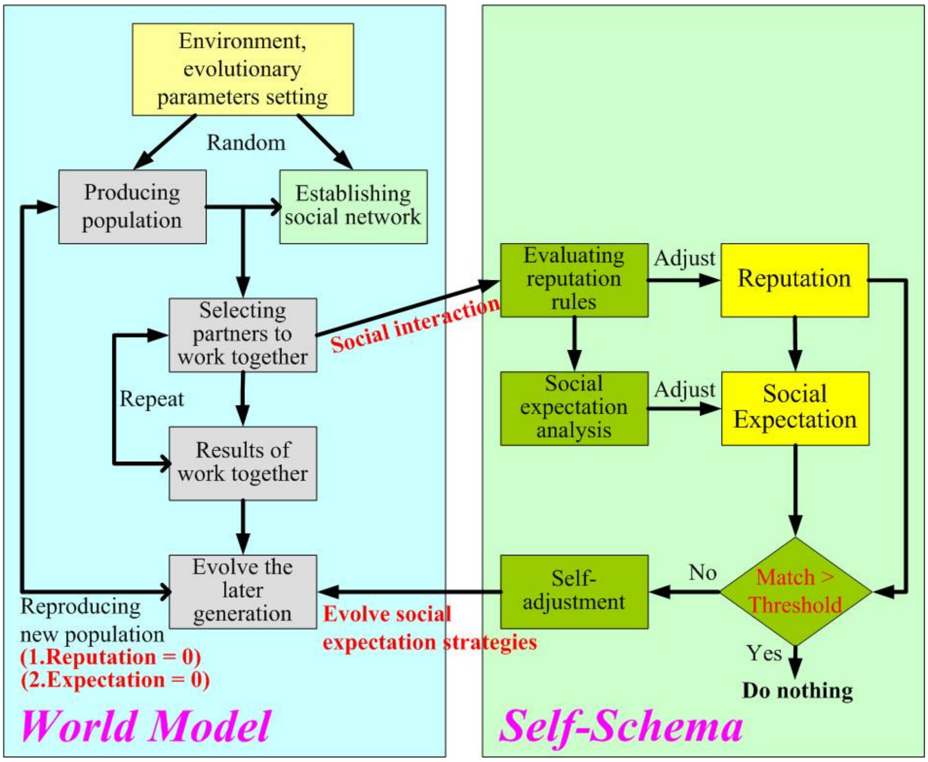


Fig. 4. Experimental procedure (world model plus self-schema).

4.1 A Few Agents with Self-Awarness that Can Improve Whole Interest

Experimental results for the first social network topology are shown in Figure 5. The five squares on the right side represent the ratios of self-aware agents. The black curve (CA: without any self- awareness agents) has some interesting implications: during early periods of evolution, individuals randomly choose strategies for working with their partners. After several generations, these individuals tend to betray their partners in order to maximize their own fitness; when most of the agents change their strategies to defection, the society falls into a self-destructive cycle that causes all social benefits to decrease rapidly. As these social benefits decreases, eventually so do private benefits, and after a few more generations, defection agents return to cooperation strategies, thereby matching the game theory concept that mutual cooperation is a better strategy for agents in iterative games. Renewed mutual cooperation triggers increases in social benefits, and the entire society moves toward an evolutionary equilibrium. According to the evolutionary dynamics of strategic agents in the control group, the simulation model matches the results of rational analysis in game theory, thus verifying the effectiveness of the simulation model.

According to the curve CA_Mix (1.0) on the figure 5, if all agents have self-awareness capabilities, all social benefits will increase and a group of agents will not fall into a destructive cycle that indicates distrust among agents. However, such an experimental setup is unrealistic. Instead, our goal is to add a limited number of self-aware agents into an existing agent system that lacks any self-aware agents, with the expectation that the introduced agents will speed up the process by which cooperative behavior emerges. Although CA_Mix (0.5), CA_Mix (0.3) and CA_Mix (0.1) may not achieve a stable state as quickly as CA_Mix (1.0), they will support a faster reduction in the chaos phenomenon, indicating that the proposed self-awareness model does produce improvement in overall social benefits. Using curve CA_Mix (0.1) as an example, even if only one agent among ten has self-awareness capability, both social and individual benefits eventually emerge at a time period that is sooner than if none of the ten agents had that self-awareness capability.

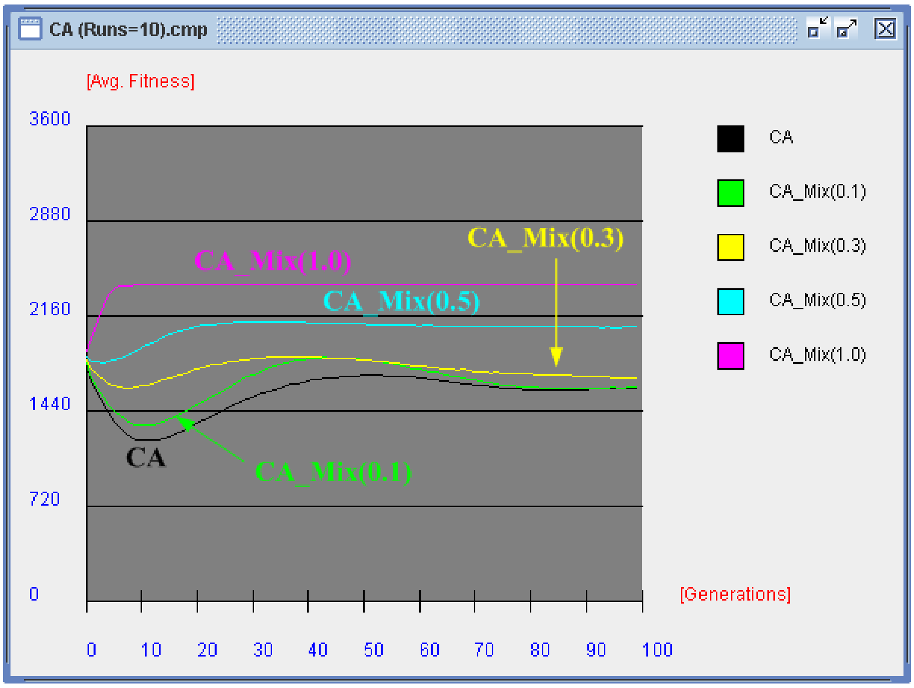


Fig. 5. Comparison with mixing partial self-awareness agents in cellular automata.

4.2 Emergence of Social Behavior

A total of sixteen single memory-strategy agents were used in our experiments. For investigating IPD model behaviour, all representative strategies that were analyzed and discussed can be classified as ALL-C, ALL-D, TFT, and PAVLOV, defined in an earlier section. We will discuss these four strategies/categories in terms of the two social networks used in the experiment.

Figure 6 illustrates reactions among the four strategies according to this topology. At the first evolutionary step, no significant difference was noted in terms of quantity. In the third generation we observed dramatic increases in ALL-D agent numbers and less dramatic decreases in the numbers of ALL-C and PAVLOV agents resulting from the growth in ALL-D agents. TFT agents, which began to emerge when ALL-D quantities achieved a certain level, checked and balanced the growth of ALL-D agents while coexisting with PAVLOV and ALL-C agents. After approximately 20th generations, TFT agents exceeded ALL-D agents; as the number of TFT agents increased, the number of ALL-D agents decreased rapidly. At approximately the 30th generation, the TFT versus STFT asynchronous memory problem began to emerge, thus triggering began the vicious circle of despiteful breach. At this point the number of PAVLOV agents started to increase because they do not suffer from memory synchronization failure. At the 60th generation the number of TFT agents becomes less than the number of ALL-D agents, and the ALL-D agents start to increase once again while the number of PAVLOV agents decrease. Finally, at the 80th generation the number of TFT agents once again exceeds ALL-D agents, and the artificial society achieves a dynamic equilibrium in which the numbers of PAVLOV and ALL-C agents remain stable (referred to as the evolutionary stable strategy, or ESS), while ALL-D and TFT continue to exist in a checks-and-balances relationship.

Cellular automata-associated results for our experiment group are shown in Figures 7 (1.0 mix ratio) and 8 (0.1 mix ratio). As shown in Figure 7, no ALL-D agents were observed at the beginning of the evolutionary process, since the cellular automata was filled with self-aware agents. Since ALL-D agents are not good matches for social good strategies, the self-aware agents quickly determine that an ALL-D existence is not permitted by their superegos, thus triggering the self-adjustment/ strategy modification mechanism of the self-awareness model. Evolutionary equilibrium is achieved at about the 3rd or 4th generation.

Figure 8 presents the most important results for our experiment, in which we added self-aware agents to the cellular automata social network at a mix ratio of 0.1. We observed that at the beginning of evolution, the ALL-D strategy was not as vigorous as that noted for the control group in Figure 6. Furthermore, a comparison of peak ALL-D numbers (at approximately the 15th generation) in Figures 6 and 8 indicate 700 ALL-D agents in the control group out of 2,500 strategic agents in the simulation without self-aware agents and 550 ALL-D agents in the experimental group (0.1 mix ratio of self-aware agents)—a significant difference of 150 agents, and an indication that even the addition of a small number of self-aware agents can speed up the process toward achieving equilibrium. Another phenomenon we observed is that the number of PAVLOV agents exceeded ALL-D and TFT agents for a certain period of time, but then decreased, suggesting that PAVLOV agents are not successful when competing against ALL-D agents, even at small numbers of ALL-D agents.

5 Conclusion

In this paper we introduced an Agent Cognition Learning Model (ACLM) and Agent Self-Awareness Model that we hope will be useful to researchers in the fields of artificial intelligence (AI), cognitive psychology, economics, and social behavior. We used AI principles to increase the thinking capabilities of agents as a means of repairing the flaws of existing intelligent agents and learning agents whose learning focuses were established according to world model guidelines. Instead, we used principles from cognitive psychology to establish a personality model that allows agents to achieve self-improvement through self-awareness, using a Prisoner's Dilemma mathematical model to address the conflict between public good and private interest in an artificial society. We eventually hope to clarify the importance of uniting internal cognition with external learning, and to revise our ACLM to offer a new approach for intelligent agents.

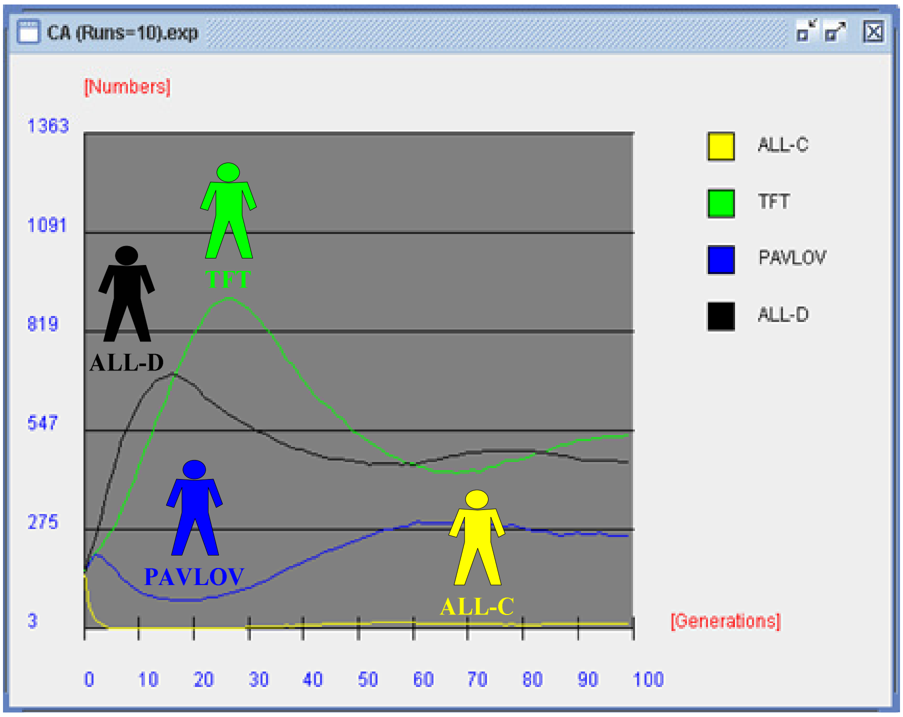


Fig. 6. Four well-known strategies in cellular automata.

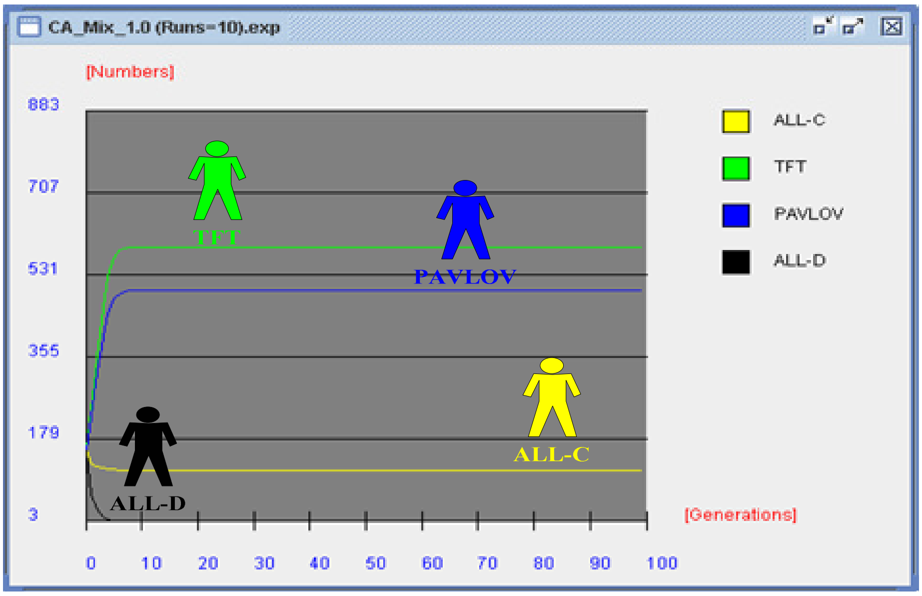


Fig. 7. Four well-known strategies in cellular automata (mixing self-aware agents with ratio 1.0).

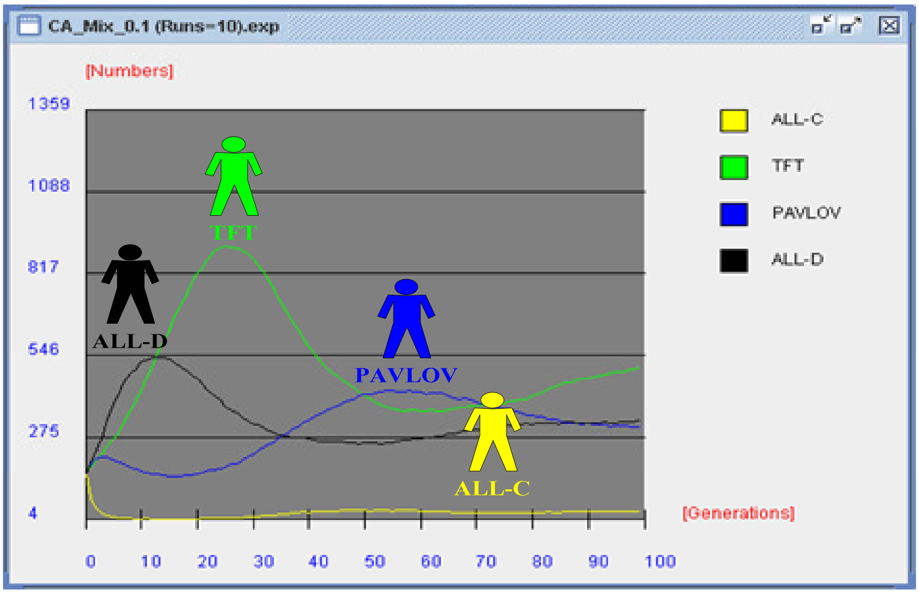


Fig. 8: Four well-known strategies in CA (Mixing self-aware agents with ratio 0.1)

Acknowledgement

This work was supported in part by the Republic of China (ROC) National Science Council (NSC97-2221-E-182-046), Chang Gung University (UERPD270281), and Chang Gung Memorial Hospital (CMRPD260022).

References

1. Duval, S. and Wicklund, R. A., *A theory of objective self-awareness*, 1972, New York: Academic Press.
2. McCarthy, J. and P.J. Hayes, Some philosophical problems from the standpoint of artificial intelligence, in *Machine Intelligence 4*. 1969: Edinburgh U. Press. p. 463–502.
3. Sirgy, M.J., Self-Concept in Consumer Behavior: A Critical Review. *Journal of Consumer Research*, 1982. 9(3): p. 287.
4. Markus, H. and E. Wurf, The Dynamic Self-Concept: A Social Psychological Perspective. *Annual Review of Psychology*, 1987. 38(1): p. 299-337.
5. Barter and Susan, The Construction of the Self: A Developmental Perspective. *Journal of Cognitive Psychotherapy*, 2001. 15: p. 383-384.
6. Boccaran, Goles E., Martinez S. Picco P., *Cellular Automata and Cooperative Phenomena*. 1993, Boston: Kluwer Academic Publisher.
7. Axelrod, Robert, and W. Hamilton, The Evolution of Cooperation. *Science*, 1981. 211 p. 1390-96.
8. Michie, D. J. Spiegelhalter, C. C. Taylor, John Campbell, eds. *Machine learning, neural and statistical classification*. Ellis Horwood.
9. Mitchell T., Buchanan B., DeJong G., Dietterich T., Rosenbloom P.,Waibel A., Machine Learning. *Annual Review of Computer Science*, 1990. 4(1): p. 417-433.
10. Wooldridge, M. and N.R. Jennings, Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, 1994.
11. Russell, S.J., and P. Norvig Artificial intelligence: a modern approach. 1995, Englewood Cliffs, NJ Prentice-Hall.

Chapter 10

SSM - Social Simulation and Modelling

On the Intelligence of Moral Agency

Helder Coelho[1] and Antônio Carlos da Rocha Costa[2]

[1] LabMAg, Faculdade de Ciências da Universidade de Lisboa, 1749-016 Lisboa, Portugal, hcoelho@di.fc.ul.pt

[2] Escola de Informática, PPGINF, Universidade Católica de Pelotas, 96.010-000 Pelotas, RS, Brazil, rocha@atlas.ucpel.tche.br

Abstract. More advanced and complex applications, such as serious games, where physical and virtual environments interchange with human and artificial agents and along heavy social simulations, require another sort of architectures. With the enlarging autonomy comes an increased need to ensure that their behaviour is in line what we expect from them. Therefore, a combination of intelligence and ethics becomes mandatory, and this means new design principles and technical requirements for the social agency and the presence of trust and confidence. Mentality, before the sole key concern, is now mixed with morality and within the social spaces where autonomous agents act on our behalf. In order to model new agent behaviours with qualities we need other kind of more intricate mental models, able to support moral reasoning capabilities. Today, the pressing quest is which are the crucial building blocks and mechanisms for those inovative agents.

Keywords: moral agency, moral agents, norm innovation.

1 Introduction

> "The function of Moral is to guide action intuitively and unconsciously."
> J. Greene, 2007.

Environment, cognition, emotions, peer pressure, values, pride and the social relations all influence our decisions between choosing right over wrong. Any decision an agent makes when it comes to prefer a good or a bad behaviour reveals his true character. This implies also agents must have an explicit conception about the outcomes of their actions and the capability to classify and assess them accordingly.

Agency is only the capacity of an agent to act in a world, yet moral agency is the responsability for making moral judgements about the acting choices, and morality refers to a certain code of conduct and a system of actions and reactions directed to keep everyone behaving according to it. In brief, Moral refers to those explicit and implicit rules and actions able to govern agents´s social behaviour.

A clear understanding of how cultural changes interact with individual agent actions is central to informing democratically and humanely guided efforts to influence cultural evolution. The example of norm innovation can help us to understand the complex 2-way dynamics of sociality (how new conventions spread in social systems), because norms emerge at the aggregate level (and immerge into the minds of agents) to fix the future behaviour of agents and the whole functioning of the society.

Moral systems are composed of four kinds of regulative elements: moral norms, moral values, moral judgements and moral actions. Norms are (conventional, social) rules or patterns of behavior, serving to maintain order and to guarantee social regulation. But, norms (and institutions) have a short life. Harmonization occurs and social order is restored. Norm innovation depends on the mechanisms by which new norms are conceived, the conditions under which they are spread, the extent to which they evolve as they are distributed through all the society, the circumstances under which they become institutionalized, and, the process through which they decay, are lost and replaced by new ones.

Moral values are of two kinds (reference and assessment values), and serve the purpose to set standards of quality and direction to the agent behaviours, and they are both closely connected to moral norms. Reference values are high level values that the society adopts in order to characterize itself in general terms (democracy, liberty, progress, adherence to heritage, religiousness, etc.). Reference values tend to be the defining elements of norms, in the sense that norms are conceived and adopted to control behaviours so as to keep the society adherent to those reference values.

Assessment values are operational values with which behaviours are dynamically evaluated, as a consequence of their compliance or not with norms. Behaviours that comply with norms are assessed positively, and behaviours that do not comply are assessed negatively. The intensity with which the behaviours comply or deviate from the norms are reflected in the magnitude of the assessment value assigned to the behaviours (bad, very bad, good, very good, etc.).

Moral judgements are rational opinions with which agents classify each other´s behaviours, according to their compliance or not to the current set of moral norms. As a result of a moral judgement, a behaviour is marked as compliant or not to the set of moral norms, and a moral norm value of the assessment kind is assigned to it. Moral judgements may be combined with other kinds of rational judgements to form moral reasoning, which are the special kinds of social reasoning through which the agents decide and/or justify the moral actions that they take.

Moral actions are regulative control actions (meta-actions) that the agents emit in order to influence each other about the adequacy of other behaviours to the current set of moral norms. Such regulative actions are of two special kinds, either punishments or rewards, and tend to assign additional costs (punishments, interdictions) or to supply new resources (rewards, permissions) to the agents´s actions, according to the moral evaluation (praise or blame) made about them, and to the moral values associated to them.

Concretely, moral actions may take either the form of behaviour control, affecting the possibility of the agents´s actions, or the form of organizational control, affecting the way agents adopt social roles to each other. How moral systems evolve? How are they represented in the mind of an agent? How are moral actions concretely realized and become effective in a certain context?

Any society can be viewed as organized along two main levels: on the bottom, the economical-material infrastructure, and at the top, the moral-cultural superstructure. There is an ongoing flow between the two levels (micro-macro) of norms and values: moral values of reference (high-level values) and moral values of evaluation (low-level or operational values). The arrival of new norms and the renewal of existing ones are related with the adaptation of reference moral values to the current working of a society. Norm innovation is guided by the evolution of reference values (with moral-cultural character) which are chosen as a consequence of political-economical dispute around the economical-material values.

More research and experimentation is necessary on questions of transmission, transformation and contribution of the mental constructs to understand the dialectical relation between social structures and individual agency and collective interaction, say the dynamics of sociality. This new knoeledge will have an influence on how the artificial mind of an agent may be architectured.

2 State of the art

The topic of moral agents became hot in recent years (see IEEE Intelligent Systems Journal, July/August 2006), due to the original scientific contributions coming from Cognitive Neuroscience, Evolutionary Psychology, or even Philosophy. Damasio´s group, at the University of Southern California, covered the social spaces (individuals in relation to others), the physiological roots of some social emotions (happiness, pride, compassion). Hauser, at Harvard University, suggested our moral judgements are derived from unconsciousness, intuitive processes that operate over the causal-intentional structure of actions and their consequences (Koenigs et all 2007). He believes we have a moral organ, a sort of faculty, able to embed a universal moral grammar (Mikhail 2007), a tool to build up specific moral systems, able to generate judgements about permissible and forbidden actions prior to the involvement of our emotions and systems of conscious, rational deliberation. Accoding to Hauser, moral rules have two ingredients, a prescriptive theory or body of knowledge (social conventions, norms, ceremony manners) about what one ought to do, and an anchoring set of diverse emotions.

Within Artificial Intelligence, Cognitive Science and Social Psychology, several authors in the last decade (Bazzan et all 1999), (Bordini et all 2000), (Allen et all

2000), (Ribeiro and Costa 2003), (Floridi and Sanders 2004), (Dimuro et all 2005), (Wiegel et all 2005), (Allen et all 2006), (Kowalski 2007), (Anderson et all 2006), (Guarini 2006), (Moor 2006), (Bringsjord et all 2006), (Wiegel 2006), (Costa and Dimuro 2007), (Savarimuthu and Purvis 2007), (Franco et all 2008), (Hegselmann 2008), (Lotzmann et all 2008), (Will 2009) start to advance ideas about Hume (sentiments), Hume/Kant (sentiments and reasoning) or Rawls (action grammars) moral creatures, building up artificial virtues (enforcement agencies). Around all these contributions we can also find a diversity of agent´s architectures devised to engender specific characters and personalities, namely recent work by (Wiegel 2006) on the necessary building blocks of an agent, (Lorini and Castelfranchi 2007) on the cognitive structure of surprise, and (Mascarenhas e all 2009) on cultural agents.

3 Moral machine

The mixture of reasoning and emotion is behind the generation of moral behaviours and judgements about gains and losses of an agent (Koenigs et all 2007). Meanwhile, emotions work as weights, pushing us more for one side that the other. The same occurs with mentality, where each mental state (eg. a belief or a desire) is constrained by a set of attributes and values (Antunes 2001), (Corrêa and Coelho 2004). The generation of actions will be responsible for producing acts, linked to utilitarian (focus on consequences) or deontological (focus on rules) judgements.

The discussion on the precise building blocks of a second generation of moral agents was advanced for the first time by (Wiegel 2006), around the construction of the SophoLab Project and it was supported on the BDI model and the JACK agent language. He clarified the general structure and organization, the design principles and the technical requirements, in particular the emergent behaviour, the redundant degrees of freedom, the absence of any central director of action, and the local scope of control. The first generation moral agents supported forms of interconnectedness (team work), multipurpose goals, but it was limited in embodiment and epistemology, the two crucial features to be explored later on. Research done around moral grammars by (Mikhail 2007) and (Hauser 2006) was not yet conclusive, from the point of view of the agent´s autonomy and its place in the heart of the organization.

At the same time, the attempt to construct artificial agents, governed by norms, was also one of the aims of the EEC EMIL (Emergence in the Loop, Simulating the Two Way Dynamics of Norm Inovation) project, started in 2006 (Andrighetto et all 2007), with the main focus on norm innovation (Lotzmann et all 2008). So, the agent design of EMIL-A was guided by the norm formation process (information transfer structure). The functional description, around the first prototype in NetLogo, was unable to reveal the mental side and to explain how autonomy and potency were not taken into full account (Lotzmann 2008). The planning (making capability) was very simple and the decision-taking module was no more than a trivial utility function. Other experiments by (Andrighetto et all 2008) adopted also simple agent architectures and did not reflect upon the ideas based upon the dialogue between cognition and affection.

Proposals, done in the pasto n drives and will inclusion were not taken into care, and even no explanation was given on how cognitive and affective states may interact (through layers) to engender a moral reasoning capability, a hint defended by Hauser and Damasio (Koenigs et all 2007) and by Cognitive Neuroscience at large (Greene 2005).

Agent autonomy depends heavily on the power-of ability and it is dependent on the will mental state (Coelho and Coelho 2009), a missing point of BDI model to allow a kind of insurgent agents with direct action potency. The introduction of a deontic element by (Wiegel 2006), extending the standard BDI model through the deontic-epistemic-action logic (DEAL framework), will not be sufficient to capture the whole flavour of a moral agency, and the same argument can also be applied to the use of BOID model (Broersen et all 2001), more keen to engender personalities. The moral conduct of an agent requires more than the means-ends analysis, the so-called planning capability of the BDI model.

A social space, in progress, associated with a serious game for managing human resources, requires a moral agency with much more advanced features. Before, in the serious game around the management of natural resources (say, water), the simplified BDI was adopted (Adamatti et all 2009), and as a consequence social interactions among agents were of poor quality. In other experimentation conducted by (Costa and Dimuro 2007) and (Franco et all 2008) moral sentiments were not taken into account because the selected scenario was not demanding in ethical considerations. On the contrary, in this particular case study, the personality (temper, character) of the agents was one of the major concerns. This year a proposal for the design of cultural agents, by (Mascarenhas 2009), mixed several sub-systems of a general agent for synthetic characters and it was near of the basic ethics machine.

4 Case study: norm-innovation

Let us select the case study of norm innovation in a society to explain how the morality Works in general. Norms play two roles, the constitutive one, to generate emergent social behaviour, and the regulative one, to be a source of social order (engender the social structure). The example of road traffic and pedestrian crossing, implemented by (Lotzmann et all 2008), show how norms modify behaviours.

A society, or even a small social community, can be seen as a complicate system of agents. Complex systems are composed of many different interacting autonomous elements and governed by social laws, with non-linear relations and network structures. So, complexity can be classified as physiological, as we look to the size or as social as we look to interactions. Let us zoom now on these systems composed of many interacting intelligent autonomous agents (AA´s).

AA´s are able to adapt and evolve with a changing environment, making independent decisions. They form new mental objects and processes, consequent to the emergent behavior of the whole system they are a part of, and act on the basis of these particular objects and processes. When a change happens at some level, it requires some further change in agents´s individual behaviours and beforehand in their minds, in order to allow the new pattern spread over that system. Norm creation and behaviour evolution work together, and the micropsychology selects behaviour, institutions and evolution to be looked at.

Individual mental change allows agents to modify their behavior accordingly. Yet, innovation in social systems is a bi-directional process: 1) bottom-up, emergence of new entities or phenomenons at the aggregate level and from the interactions among agents impose creation of structures; and, 2) top-down, immergence of entities or phenomenons in the minds of the agents, ie. the insurgence in their minds of a new mechanism, representation or process that leads the agents to modify their behaviours in conformity with the emerged effect. Sociality is viewed along two loops: the emergence of structures at the macro level (social relations, groups) and the immergence of norms at the micro level (individual interaction).

The behavior regulation is done by norms (rules, conventions, patterns) which are central in the role theoretic concept of individual action and decision taking. There are two main types of norms, those for coordination, and those for obligations, prescriptions and directives on commands. The regulation is made by 1) change of action and 2) roles (defined by attributes, behavior and social relations). And, the fine tuning is achieved by 1) processes among agents, 2) normative transmission, and 3) transformation.

Moral agency is the agent´s responsibility for making moral judgements and taking actions that comport with morality. Moral decisions (foundation of morality) are triggered by reason (evidence, facts) and also by emotion. Both cooperate for generating moral sentiments.

Moral judgement uses a utilitarian calculus for choosing between right and wrong. Behind a moral decision there is always an interplay of thought, emotion, prevision, empathy, anguish and ambivalence.

How is normative governance effected? Societies (groups) are regulated by different sorts of mechanisms associated to 1) decaying and spreading of norms, and also to 2) transformation and internalization of norms.There are three main classes of normative agent-based simulation models. The point of departure was done by (Axelrod 1986) along the game-theoretic approach. It is still good for explaining the dynamics of norms, and the strategic adaptation of agents to changing environmental conditions.

The second direction of simulation models is the AI approach. It follows (Conte and Castelfranchi, 1995) and demonstrates the effects of norms. It includes norms that exceed strategic adaptation, which can be interpreted as an internalized property (Castelfranchi et all 1998), (Castelfranchi 1999). The weak point: agents are normative automata and no mechanisms for transmission and transformation problems are allowed.

Actualy, there is a convergence of both traditions: the models of (Verhagen 2001) and (Savarimuthu 2007) include elements of the other line of thought, and a partial answer to the questions of transformation, transmission and contribution. Both models replicate not only the findings, but also the shortcomings of the classical role theory.

5 Prospect for a moral agent

The interplay of mentality, sociality and morality reveals the definite anatomy of those smart creatures able to think about, to interact with others in a society, and also to decide upon good and evil. Which is the most suitable architecture for an agent with these three qualities?

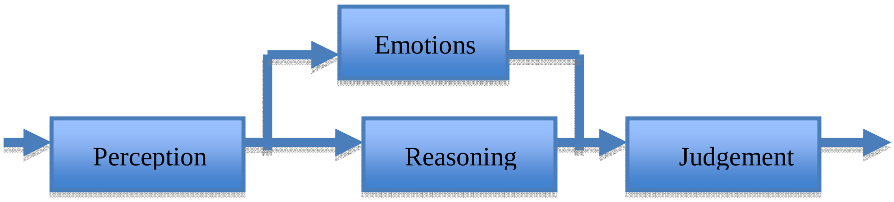


Fig. 1 Simplified sketch of a moral creature influenced by Hume and Kant ideas.

Agents can be reduced to simple bit strings when genetic programming is adopted in social simulation of complex scenarios. In what concerns symbolic programming, an agent can be more elaborated than a decision (utility) function. For example, in a risky environment, (Castelfranchi et all 2006) adopted a one-layer structure by mixing an extended BDI with an emotion manager for modeling cautious agents. In the serious game (mixing human and artificial agents) of water management (Adamatti et all 2009), an artificial agent has a behavioural profile linked to one or more strategies regarding a certain role (BDI model was simplified), no learning and planning modules were available, and only reduced decision making skills were offered, and again a one-layer structure was adopted. In another serious game on participatory management of protected areas (Briot et all 2008), conflict dynamics was taken care and a more advanced decision capability was implemented, but agents had no mentality and affective power. When designing cultural agents, (Mascarenhas 2009) updated an old architecture of social intelligent agents for educational games and proposed to combine a memory store with a reactive device and a deliberative machine, without forgetting the motivational states of the other agents. However, serious games require moral agents in order to be acceptable (serious) by users.

A moral agent, as it was defended by Hauser (see figure 1) and Green, is a mix of cognitive and affective capabilities, but no architecture was till today presented as the definite one, despite several design attempts from Hume, Kant or Rawls, and the trials made by the community of Agents. Several questions needed to be answered: What makes a moral (norm-abiding, virtuous, conventional) agent? By what mechanisms can moral behaviour (abstract values) spread or decay from one agent to another (like memes)? How are explicit morals implemented and added to the overall architecture?

The human mind is not completely rational in order to win when facing reality. Therefore, an artificial mind is also forced to avoid traps and to take into account other skills apart of reason, such as sensorial perception, intuitions, emotions and socio-cultural constructs. For example, prefer a short cut when facing discomfort, assign a value based upon some preconceived opinion, or go on with some previous agreement can disturb good decisions. The answer is to adopt a set of heuristics, like to maintain the long term plan, delay action in order to think about several alternatives, and to perform as an outsider when looking to that issue for the first time. Be less moralist when analyzing what is or not right, use those issues involved in the decision (more cooperative), and follow devil´s lawyer to embody relevant data (otherwise absent) is a recipe to avoid catastrophes.

A moral agent needs to get a more intricate way of thinking (Kowalski 2007) than a simple reactive (assimilate observations of changes in the environment) or a proactive one (reduce goals to sub-goals and candidate actions). Why? It is not sufficient to embody a goal-based or a value-based model. We need a mix of intuitive (low level) and deliberative (high level) processes, and also the ability to think before acting (pre-active) when choosing between right or wrong, ie. capability to think about the consequences of the candidate actions (generate logical consequences of candidate actions, helping to decide with heuristics or decision theory between the alternatives). The classic component based on the observe-think-decide-act cycle (present in the BDI model) is unable to deal with morality because we get different kinds of goals (achievement, maintenance) and, at the same time, preferences and priorities are requested. The one-layer structure is no longer the solution because we arrive at our ultimate moral (utilitarian, where results maximize the greatest goods, or deontological, where any moral evaluation is independent of consequences) judgements by a mix of emotions and conscious reasoning. As a matter of fact, emotions drive behaviours like weights, and play a critical mediating role in the relationship between an action´s moral status and its intentional status. A moral ability may be seen as a set of rules (a grammar according to Hauser) to constrain the behaviour of the agent: each rule having two ingredients, the body of knowledge and the set of anchored emotions, which are going to interplay.

Every decision an agent makes, when it comes to choosing between right or wrong, reveals his true character (subjectivity): Humean model with emotions behind judgements, or Rawlsian model, with emotions and reasons after judgements have only one layer and trade-offs are not allowed. There is always a sentiment of avoidance in violating what seems to be reasonable, ie. the possibility to have access to the outcomes (classifications) of the agent actions.

A moral agent associates always reason with emotion, social values and cultural-situational knowledge before making a decision. Therefore, its more-than-one-layer architecture, integrating micro and macro levels, requires an extended (with will and expectations) BDI model, the addition of emotional machinery to deal with sentiments, a library of contexts to situate any evaluation, heuristics to avoid wrong decisions (mind traps), a sort of universal moral grammar (Mikhail 2007) to fix any sort of moral system and action generation, and also modules concerning decision taking, constraint satisfaction (reinforcement) learning and planning. The organization with interconnected multiple layers seems inevitable on account of the balance between reasoning and emotion and the assembling/tuning of composite judgements (embedded in preference criteria).

6 Conclusions

The research and experimentation around the intelligence of moral agency is betting on understanding and managing complexity in social systems with autonomous agents. The selection of norm innovation, as a topic, is also helping us to comprehend now how new conventions and principles of right (and wrong) action emerge and spread in those systems to get social order. Norms are complex social artifacts because of the role in connecting emergence and immergence, through a movement between micro and macro levels. Applications such as regulation of e-communities or realistic serious games for managing human capital are eager of new agent models and architectures with ethical concerns and some sort of subjectivity.

Several open questions frame our current research: How do actors produce and are at the same time a product of social reality? How an idea (memes) for a behavior that becomes a norm gets invented in first place? Which ideas are accepted and which are rejected driven by adaptation and evolution? How many are slowly assembled from diverse data in a single mind? Answers, from Cognitive Neurosciences, Moral or Evolutionary Psychology, point to a strong focus on a context sensitive approach to agency and structure, the interplay of which leads to emergent phenomena, underlining the generative paradigm of computational social science. Agent-based modeling and simulation can be of great help in order to allow a better comprehension of this sort of complexity.

References

Adamatti, D., Sichman, J. and Coelho, H. An Analysis of the Insertion of Virtual Players in GMABS Methodology Using the Vip-JogoMan Prototype, Journal of Artificial Societies and Social Simulation, JASSS in press, 2009.

Allen, C., Varner, G. and Zinser, J. Prolegomena to any Future Artificial Moral Agent, Journal of Experimental Theoretical Artificial Intelligence, Volume 12, pp. 251-261, 2000.

Allen, C., Wallach, W. and Smit, I. Why Machine Ethics? IEEE Intelligent Systems, Volume 21, Number 4, July/August, 2006.

Anderson, M. and Anderson, S. L. Machine Ethics, IEEE Intelligent Systems, Volume 21, Number 4, July/August, 2006.

Anderson, M., Anderson, S. L. and Armen, C. An Approach to Computing Ethics, IEEE Intelligent Systems, Volume 21, Number 4, July/August, 2006.

Andrighetto, G., Campenni, M., Conte, R. and Paolucci, M. On the Immergence of Norms: a Normative Agent Architecture, Proceedings of AAAI Symposium, Social and Organizational Aspects of Artificial Intelligence, Washington DC, 2007.

Andrighetto, G., Campenni, M., Ceccone, F. and Conte, R. Conformity in Multiple Contexts: Imitation vs. Norm Recognition, Proceedings of the World Congress of Social Simulation, Fairfax VA, July, 2008.

Antunes, L. Agents with Decisons Based in Values (in Portuguese), PhD Thesis, Faculty of Sciences, Universidade de Lisboa, 2001.

Antunes, L., Balsa, J., Urbano, P. and Coelho, H. The Challenge of Context Permeability in Social Simulation, Proceedings of the Fourth European Social Simmulation Association Conference, Toulouse, September, 2007.

Antunes, L., Balsa, J., Urbano, P. and Coelho, H. Exploring Context Permeability in Multiple Social Networks, World Congress on Social Simulation (WCSS-2008), George Mason University Fairfax (USA), July 14-17, 2008.

Axelrod, R. An Evolutionary Approach to Norms, American Political Science Review, Volume 80, 1986.

Bazzan, A. L. C., Bordini, R. H. and Campbell, J. Moral Sentiments in Multi-Agent Systems, in Intelligent Agents V, Springer-Verlag LNAI Nº 1555, pp. 113.131, 1999.

Bordini, R. H., Bazzan, A. L. C., Vicari, R. M. and Campbell, J. Moral Sentiments in the Iterated Prisoner´s Dilemma and in Multi-Agent Systems, Brazilian Journal of Economics, Volume 3, Number 1, 2000.

Bringsjord, S., Arkoudas, K. and Bello, P. Towards a General Logicist Methodology foe Engineering Ethically Correct Robots, IEEE Intelligent Systems, Volume 21, Number 4, July/August, 2006.

Briot, J.-P., Vasconcelos, E., Adamatti, D., Sebba, V., Irving, M., Barbosa, S., Furtado, V. and Lucena, C. A. Computer-Based Support for Participatory Management of Protected Áreas: The SimParc Project, Proceedings of XXVIIIth Congresso f Computation Brazilian Society (CSBC´08), Belém, Brazil, July 2008.

Broersen, J., Dastani, M., Hulstijn, J., Huang, Z. and van der Torre, L. The BOID Architcture: Conflicts between Beliefs, Obligations, Intentions and Desires, Proceedings of the Fifth International Conference on Autonomous Agents and MultiAgent Systems (AAMAS01), Montreal, Quebec, Canada, pp. 9 – 16, 2001.

Castelfranchi, C., Conte, R. and Paolucci, M. Normative Reputation and the Costs of Compliance, JASSS, Volume 1, Number 3, 1998.

Castelfranchi, C. Prescribed Mental Attitudes in Goal, Adoption and Norm-Adoption, Artificial Intelligence and Law, Volume 7, pp. 37-50, 1999.

Castelfranchi, C. The Theory of Social Functions: Challenges for Computational Social Science and Multi-Agent Learning, Journal of Cognitive Systems Research, Volume 2, pp. 5-38, 2001.

Coelho, F. and Coelho, H., Meta Agency and Individual Power, Web Intelligence and Agent Systems: An International Journal (WIAS), in print, 2009.

Conte, R. and Castelfranchi, C., Cognitive and Social Action, UCL Press, 1995.

Corrêa, M. and Coelho, H. Collective Mental States in an Extended Mental States Framework, International Conference on Collective Intentionality IV, Certosa di Pontignano (Italy), October 13-15, 2004.

Costa, A. R. and Dimuro, G. P. A Basis for an Exchange Value-Based Operational Notion of Morality for Multiagent Systems, Springer-Verlag LNCS, 2007.

Dimuro, G. P., Costa, A. R. and Palazzo, L. A. M. Systems of Exchange Values as Tools for Multi-Agent Organizations, Journal of Brazilian Computer Society, 2005.

Floridi, L. and Sanders, J. W. On the Morality of Artificial Agents, Minds and Machines, Volume 14, pp. 349-379, 2004.

Franco, M., Costa, A. R. and Coelho, H. Simulating Argumentation about Exchange Values in Multi-Agent Interactions, Proceedings of the 1st Brazilian Workshop on Social Simulation (BWSS 2008), 19th Brazilian Symposium on Artificial Intelligence (SBIA2008), Salvador. October 26-30, 2008.

Grau, C. There is No “I” in “Robot”: Robots and Utilitarism, IEEE Intelligent Systems, Volume 21, Number 4, July/August, 2006.

Greene, J. D. The Terrible, Horrible, No Good, Very Bad Truth about Morality and what to do about It, PhD Dissertation, Princeton University, 2002.

Greene, J. D. Cognitive Neuroscience and the Structure of the Moral Mind, in The Innate Mind: Structure and Contents, S. Laurence, P. Carruthers and S. Stich (Eds.), Oxford University Press, 2005.

Guarini, M. Particularism and Classification of Moral Codes, IEEE Intelligent Systems, Volume 21, Number 4, July/August, 2006.

Hauser, M. D., Chomsky, N. and Fitch, W. T. The Faculty of Language: What is it, Who hás it, and How did it evolve), Science, Volume 298, 22 November, 2002.

Hauser M. D. The Liver and the Moral Organ, SCAN, Volume I, pp. 214-220, 2006.

Hauser, M. D. Moral Minds: How Nature Designed our Sense of Right and Wrong, Ecco/Harper Collins, 2006.

Hegselmann, R. Modelling Hume, a draft for HUME1.0, February, 2008.

Koenigs, M., Young, L., Adolphs, R., Tranel, D., Cushman, F., Hauser, M. and Damasio, A. Damage to the Prefrontal Cortex Increases Utilitarian Moral Judgements, Nature, March 21, 2007.

Kowalski, R. The Logical Way to Be Artificially Intelligent, Lecture Notes in Computer Science, Springer-Verlag, 2007.

Lotzmann, U., Möhring, M. and Troitzsch, K. Simulating Norm Formation in a Traffic Scenario, Proceedings of the Fifth Annual Conference of the European Social Simulation Association, Brescia, September, 2008.

Lotzmann, U. TRASS, A Multi-Purpose Agent-Based Simulation Framework for Complex Traffic Simulation Applications, A. L. C. Bazzan and F. Klügl (Eds.), Multi-Agent Systems for Traffic and Transportation, IGI Global Press, 2008.

McLaren, B. M. Computational Models of Ethical Reasoning: Challenges, Initial Steps, and Future Directions, IEEE Intelligent Systems, Volume 21, Number 4, July/August, 2006.

Mikhail, J. Universal Moral Grammar: Theory, Evidence and the Future, Trends in Cognitive Science, Volume 11, Number 4, 2007.

Moor, J. H. The Nature, Importance and Difficulty of Machine Ethics, IEEE Intelligent Systems, Volume 21, Number 4, July/August, 2006.

Nichols, S. Norms with Feeling: Toward a Psychological Account of Moral Judgment, Cognition, Volume 84, pp. 221-236, 2002.

Nichols, S. Sentimental Rules, Oxford University Press, 2004.

Pereira, L. M. and Saptawijaya, A. Computational Modelling of Morality, Working Report, 2009.

Powers, T. M. Prospects for a Kantian Machine, IEEE Intelligent Systems, Volume 21, Number 4, July/August, 2006.

Ribeiro, M. R. and Costa, A. R. Using Qualitative Exchange Values to Improve the Modelling of Social Interactions, Springer-Verlag LNCS, 2003.

Savarimuthu, B. T. R. and Purvis, M. Mechanisms for Norm Emergence in Multiagent Societies, Proceedings of the 6th International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS-07), Honolulu, Hawaii, 2007.

Urbano, P. Decentralised Consensus Games (in Portuguese), PhD Thesis, Faculty of Sciences, University of Lisbon, 2004.

Verhagen, H. Norms and Artificial Agents, Proceedings of the Sixth Meeting of the Special Interest Group on Agent Based Systems, 2001.

Verhagen, H. Simulation of the Learning of Norms, Social Science Computer Review, 2001.

Wiegel, V., van der Hoven, M. J. and Lokhorst, G. J. C. Privacy, Deontic Epistemic Action Logic and Software Agents, Ethics and Information Technology, Volume 7, pp. 251-264, 2005.

Wiegel, V. Building Blocks for Artificial Moral Agents, Working Report, Proceedings of EthicalALife06 Workshop, 2006.

Wiegel, V. Sopholab; Experimental Computational Philosophy (Simon Stevin Series in the Philosophy of Technology). Delft University of Technology, PhD thesis, 2007.

Will, O. Modelling Hume´s Moral and Political Theory, An Agent-Based Model on the Evolution of Trust in Strangers and Division of Labour Among, Working Report, 2009.

Fuzzy Evaluation of Social Exchanges Between Personality-based Agents

G.P. Dimuro[1], A.V. Santos[1], B.C. Bedregal[2], and A.C.R. Costa[1]

[1] PPGINF, Universidade Católica de Pelotas, 96010-000 Pelotas, Brazil
[2] DIMAP, Universidade Federal do Rio Grande do Norte, 59072-970 Natal, Brazil
{liz,avs,rocha}@ucpel.tche.br,{bedregal}@dimap.ufrn.br

Abstract. Social interactions can be seen as service exchanges, followed by the qualitative evaluation of those exchanges by means of social exchange values. This paper introduces an approach based on Fuzzy Logic for the evaluation of the investment and satisfaction values generated in personality-based social exchanges in multiagent systems. The fuzzy equations of material social equilibrium, associated to the first stage of a social exchange between agents, are analyzed in detail. A case study is presented to illustrate the proposed approach.

1 Introduction and Related Work

According to [1], it is possible to distinguish between three coordination models in Multiagent Systems (MAS): hierarchies, markets and networks, which result in different frameworks for agent societies, driving the way the social interactions between agents are analyzed. This work is in the context of the agent-based simulation [2, 3] of network-oriented societies with collaboration-based social control, based on Piaget's theory of *social exchanges* [4], in the line of the works in [5–12].

In Piaget's theory, social interactions are seen as service exchanges between pairs of agents, followed by the qualitative evaluation of those exchanges by the agents by means of *social exchange values*: the *investment* value for performing a service or the *satisfaction* value for receiving it, called *material values*. The exchanges also generate values of *debts* and *credits* that help to guarantee the continuity of future interactions. A MAS is said to be in *social equilibrium* if the balances of investment and satisfaction values are equilibrated for the successive exchanges occurring along the time.

Social exchange values are of a qualitative nature, since they represent subjective values with which everyone judges the daily exchanges he has (*good*, *bad*, *better than*, *worst than* etc.), which usually cannot be faithfully represented quantitatively, due to the lack of neat objective conditions for their measurement. In fact, one of the main contributions of the exchange values approach to social exchanges, in comparison to the classical, quantitative utility function based approach, is precisely the introduction of the possibility of taking care of the variety of such subjective values.

Notice that the notion of exchange in Piaget's theory, which can be related to the work of Homans [13], in some sense refers to the notion of an exchange viewed as "adoption conditional to reciprocation" of the cognitive model of exchange of Conte and Castelfranchi [14], since the realization of a service by an agent in behalf of other agent presupposes that the first adopts the latter. However, in [14] there is no concern neither with the evaluation of the services performed nor with the generation of debts and credits, as a foundation for the continuity of future interactions.

In [6, 9], agents were modeled with the help of a set of simple *social exchange personality traits* (e.g., egoism and altruism),[3] in such way that the performance of the same service could be evaluated by the agents differently according to their various social exchange personality traits (e.g., the performance of a service that seems to be "good" for an altruist agent may be classified as "bad" by an egoist agent). This way of characterizing agent social exchange behaviors was said to be *personality-based*.[4]

In [6], a representation based on Interval Mathematics [20] was used to capture the qualitative nature of Piaget's concept of scale of exchange values [4]. Although this representation seems to be a compromise between a qualitative and a quantitative representation, making it mathematically operational, and the decision process computationally viable, it lacks the subjectiveness of personality-based evaluations.

The work presented in [10–12] introduced a methodology for the evaluation of services in non-economics exchange processes, in order to propose a mechanism for partner selection and coalition in the context of bioinformatics exchange services. In [7, 8], social exchange values were used to incorporate sociability as a means to improve coordination mechanisms in a virtual agent environment. However, in both works we do not find an effective qualitative, subjective representation of exchange values.

On the other hand, the Fuzzy Set Theory [21] introduced in 1965 by Zadeh [22] offers a natural way of representing vagueness and subjectiveness in everyday life. It is based on the idea that several elements in human thinking are not exact data, but can be approximated as classes of objects in which the transition from membership to nonmembership is gradual rather than abrupt. Besides that, human reasoning sometimes does not follow the two-valued or multivalued logic, but logic with fuzzy truths, fuzzy connectives, and fuzzy rules of inference.

Fuzzy logic has been largely used in Soft Computing, with application in different areas, as, e.g., in the context of Social Simulation. In [23], it was observed that simple agent models, as those normally used with exiting tools, are neither sufficient nor adequate to deal with the uncertainty and subjectiveness that have to be considered in the analysis of values (like, e.g., trust) in human societies. Then, in the agent-based social modeling and simulation presented in [23], Fuzzy Logic was used to naturally specify attributes of the agents representing individuals, the evolution of the agent minds, the inheritance, the relationship and similarity between individuals, etc.

Zadeh has pointed out the incompatibility principle, which states that "complexity and precision are incompatible properties", arguing that the conventional numerical-based approaches are inadequate to model human knowledge in complex processes, like, for example, human personality traits. Then, [17, 18] specified human personality

[3] See [15, 16] for a discussion on applications of personality-based agents in the context of MAS.

[4] The notion of egoism (altruism) in social exchange personality traits is not related to the usual concept of selfish (altruist) agents as in [14], since here those personality traits are simply different forms of evaluating social exchanges. For the same reason, the social exchange personality traits can not be compared to patterns of behavioral, temperamental, emotional, and mental traits that are used in several works in social and human simulation, as, e.g., in [17–19].

facets and traits (according to the Big Five and OCEAN models) as conditional rules in fuzzy agentes (agents that can perform qualitative uncertainty reasoning), in order to perform human behavior simulation. See also [24] for an application of Fuzzy Logic in the simulation of human behavior and social networks, representing behavioral elements, such as stress, motivation or fatigue, and sociological aspects.

Differently from [17, 18], in our previous works [6, 9] the social exchange personality traits were defined by means of probability state transition functions, specifying the way the agents increase or decrease their exchange results, by accepting or refusing exchanges proposals according to their preferences on exchange stages (see Sect. 2 for the two different kinds of exchange stages). However, there was no qualitativeness in the evaluations of the exchanges performed by the agentes. In this paper, those social exchange personality traits are enriched by endowing the agents with a fuzzy mechanism to perform qualitative evaluations of social exchanges, according to tendencies specified by their different personalities.

The aim of this work is then to introduce an approach based on fuzzy logic for the evaluation of the material social exchange values generated in the first stage of the social exchanges between personality-based agents, with the analysis of the fuzzy equilibrium equation associated with this stage. This is the first step for the simulation of self-regulation of social exchanges using personality-based MAS [9], where the social equilibrium is treated as a fuzzy concept.

This paper is organized as follows. Section 2 presents a brief description of the model of social exchanges that we adopt in this work. The fuzzy approach for the evaluation of the material exchange values generated in the first stage of a social exchange process between personality-based agents is introduced in Sect. 3. A case study in presented in Sect. 4. Section 5 is the Conclusion with final remarks.

2 The Modelling of Social Exchanges

According to Piaget's approach to social interaction [4], a *social exchange* between two agents, α and β, is seen as an exchange of services between those agents joint with the subjective evaluation of the services by the agents themselves, by means of the so-called qualitative exchange values.

A social exchange involves two types of stages. In stages of type $\mathrm{I}_{\alpha\beta}$, α realizes a service for β. The *exchange values* involved in this stage are the following: $r_{\mathrm{I}_{\alpha\beta}}$, which is the value of the *investment* done by α for the realization of a service for β; $s_{\mathrm{I}_{\beta\alpha}}$, which is the value of β's *satisfaction* due to the receiving of the service done by α; $t_{\mathrm{I}_{\beta\alpha}}$ is the value of β's *debt*, the debt it acquired for its satisfaction with the service done by α; and $v_{\mathrm{I}_{\alpha\beta}}$, which is the value of the *credit* that α acquires for having realized the service for β. In stages of type $\mathrm{II}_{\alpha\beta}$, α asks the payment for the service previously done for β, and the values related with this exchange have similar meaning. The order in which the stages may occur is not necessarily $\mathrm{I}_{\alpha\beta} - \mathrm{II}_{\alpha\beta}$.

The values $r_{\mathrm{I}_{\alpha\beta}}$, $s_{\mathrm{I}_{\beta\alpha}}$, $r_{\mathrm{II}_{\beta\alpha}}$ and $s_{\mathrm{II}_{\alpha\beta}}$ are called *material values* (investments and satisfactions), generated by the evaluation of *immediate exchanges*; the values $t_{\mathrm{I}_{\beta\alpha}}$, $v_{\mathrm{I}_{\alpha\beta}}$, $t_{\mathrm{II}_{\beta\alpha}}$ and $v_{\mathrm{II}_{\alpha\beta}}$ are the *virtual values* (credits and debts), concerning exchanges that are expected to happen in the future [4].

A *social exchange process* is composed by a sequence of stages of type $\mathrm{I}_{\alpha\beta}$ and/or $\mathrm{II}_{\alpha\beta}$ in a set of discrete instants of time. The *material results*, according to the points of view of α and β, are given by the sum of material values of each agent. The *virtual results* are defined analogously. A social exchange process is said to be in *material equilibrium* [5, 6] if in all its duration it holds that the pair of material results of α and β encloses a given equilibrium point[5]. If just the internal equilibrium of an exchange is considered, then the following equilibrium equations [4] are taken into account:[6]

$$\text{Rule } \mathbf{I}_{\alpha\beta} : (r_{\mathbf{I}_{\alpha\beta}} = s_{\mathbf{I}_{\beta\alpha}}) \wedge (s_{\mathbf{I}_{\beta\alpha}} = t_{\mathbf{I}_{\beta\alpha}}) \wedge (t_{\mathbf{I}_{\beta\alpha}} = v_{\mathbf{I}_{\alpha\beta}}) \quad (1)$$

$$\text{Rule } \mathbf{II}_{\alpha\beta} : (v_{\mathbf{II}_{\alpha\beta}} = t_{\mathbf{II}_{\beta\alpha}}) \wedge (t_{\mathbf{II}_{\beta\alpha}} = r_{\mathbf{II}_{\beta\alpha}}) \wedge (r_{\mathbf{II}_{\beta\alpha}} = s_{\mathbf{II}_{\alpha\beta}}) \quad (2)$$

$$\text{Rule } \mathbf{I}_{\alpha\beta}\mathbf{II}_{\beta\alpha} : \ r_{\mathbf{I}_{\alpha\beta}} = s_{\mathbf{II}_{\alpha\beta}} \quad (3)$$

3 Fuzzy Evaluation of Material Exchange Values

Fuzzy sets constitute the oldest and most reported soft computing paradigm, used to modeling different forms of subjectiveness, uncertainties and ambiguities, often encountered in real life. A fuzzy set is a class of objects characterized by a membership function that assigns to each object a degree of membership ranging from 0 to 1. The algebra and other concepts of classical sets were extended to fuzzy sets, and the various properties of these notions were well established. Fuzzy logic, which is the logic underlying fuzzy set theory, was also deeply discussed in the literature [21, 22].

Definition 1. *Let X be a (classical) set. A* fuzzy subset $\mathbb{F}$ *of X is a set of ordered pairs $\mathbb{F} = \{(x, \mu_{\mathbb{F}}(x)) \mid x \in X\}$, where $\mu_{\mathbb{F}} : X \to [0, 1]$ is the* membership function *that gives the* membership degree *of x in $\mathbb{F}$, with the degrees 1 and 0 representing, respectively, the full membership and the non-membership of the element to $\mathbb{F}$.*

3.1 Services and Scales of Values for the Fuzzy Evaluation of Service Attributes

The performance of a service by an agent α to an agent β generates the material values of investment $r_{\alpha\beta}$ (from α's counterpart) and of satisfaction $s_{\beta\alpha}$ (from β's counterpart). For the evaluation of a service, according to those different points of view, one has to take into account all the factors that may influence such evaluations. It follows that:

Definition 2. *A* service in a social exchange *is defined as a tuple $\mathbb{S} = (a_1, \ldots, a_n)$, where each a_i, with $1 \le i \le n$, $n \in \mathbb{N}$, is a* service attribute *representing an aspect of the service to be analyzed in the evaluation process of the material values generated by the performance of $\mathbb{S}$. The service is denoted by $\mathbb{S}_r(\alpha)$ when the evaluation process of the investment value $r_{\alpha\beta}$ of an agent α is considered. Whenever the evaluation process of the satisfaction value $s_{\beta\alpha}$ of an agent β is analyzed, the service is denoted by $\mathbb{S}_s(\beta)$.*

[5] Notice that Piaget's notion of equilibrium has no game-theoretic meaning, since it involves no notion of game strategy, and concerns just an algebraic sum.

[6] See also [5, 6, 9–11], for more details on this modeling.

The set of attributes is application dependent, and may be different for the evaluation of the service from de points of view of either the agent that performs the service or the one that receives the service. An attributes is represented by a *linguistic variable*, whose value is expressed qualitatively by a *linguistic term* specified by a membership function μ, which constitutes the *fuzzy scale* for the qualitative evaluation of the attribute (see, e.g., the fuzzy scale for the evaluation of the attribute *complexity*, shown in Fig. 1, where the possible values for the complexity are: *low*, *medium* and *high*).[7] The limit value of the x-axis in an evaluation scale is denoted by $N \in \mathbb{N}$.

A fuzzy scale is said to be *decrescent* if greater the measure in the x-axis, worse the attribute evaluation based on the linguistic terms of the scale (see, e.g., the scale of Fig. 1). Otherwise, the fuzzy scale is said to be *crescent*. A fuzzy scale for the evaluation of an attribute a (linguistic variable), with possible values (linguistic terms) $T_1, \ldots, T_m$ is denoted by $T^a = \langle T_1, \ldots, T_m \rangle$, where $m \in \mathbb{N}$ is the fuzzy scale cardinality. The notation $T_k \in T$ means that "the linguistic term T_k is in the fuzzy scale T".

3.2 Social Exchange Agent Personality Traits

The modeling of agent social exchange personality traits considered in this paper is inspired on [6, 9], where the authors introduced a set of agent personality traits for the application in the context of social exchanges in MAS, determining the way the agentes decide on accepting or refusing an exchange proposal. In this paper, those social exchange personality traits also determine different forms in which the agents evaluate the services exchanged, generating qualitative fuzzy exchange values.

Then, an agent is said to be *egoist* if it overvalues the investment for the performance of a service and depreciates the satisfaction for a service received from other agent. On the contrary, an agent is said to be *altruist* whenever it overestimates its satisfaction due to a received service and undervalues its investment when performing a service for other agent. A *tolerant* agent is the one that realizes evaluations according to the common sense, that is, according to the average of the evaluations realized by the local people in analogous situations (obtained by, e.g., an statistical analysis).

A *personality factor* is associated to each agent personality trait, determining the influence of such personality trait in a attribute evaluation. For a decrescent fuzzy scale, the personality factors of tolerance, egoism and altruism, denoted by γ_{tol}, γ_{ego} and γ_{alt}, respectively, satisfy the following constraints:

$$0 \leq \gamma_{ego} < 1 - \delta \quad \text{(depreciation factor)} \tag{4}$$

$$1 - \delta \leq \gamma_{tol} \leq 1 + \delta \quad \text{(neutral factor)} \tag{5}$$

$$1 + \delta < \gamma_{alt} \leq 2 \quad \text{(overvaluation factor)} \tag{6}$$

where $0{<}\delta{<}1$ are the variation limits. The inferior limite equal to 0 in Eq. (4) means a depreciation of 100%, and the superior limit equal to 2 in Eq. (6) represents a overvaluation of 100%. The personality factors for a crescent fuzzy scale are defined analogously.

3.3 Fuzzy Evaluation of de Services

For the evaluation of a service attribute a using a fuzzy scale it is necessary first to realize a *normalization* process, which takes into account the personality factors.

[7] We use trapezoidal-shape membership functions in the configuration of fuzzy scales.

Definition 3. *Let $V(a)$ be the measured value of a service attribute a, N be the limite value of the x-axis of a decrescent fuzzy scale and* max *be the limit value of a according to the common sense. The normalized value of the service attribute a, for an agent with personality factor γ, is defined as:*

$$V_{nor}(a) = min\{N, V'(a)\}, \text{ where } V'(a) = \frac{V(a) \times N}{\max} \times (2 - \gamma). \quad (7)$$

The service attribute normalization for a crescent fuzzy scale is defined analogously.

The normalized value of the service attribute a is then evaluated in a fuzzy scale, obtaining the fuzzy evaluation of the service attribute, denoted by $\mu(a)$ (see Fig. 1).

Consider a social exchange between a pair of agents α and β, which is started with a service $\mathbb{S}_r(\alpha)$, performed by α to β, defined as $\mathbb{S}_r(\alpha) = (a_1, \ldots, a_n)$. Then, a set of conditional rules is obtained through the crossing of the individual fuzzy evaluation results of the service attributes $a_1, \ldots, a_n$, using the MAX-MIN inference rule [21].

Let $T^i = \langle T_1^i, \ldots, T_{k_i}^i \rangle$ be the fuzzy scale for the evaluation of the attribute a_i of the service $\mathbb{S}_r(\alpha) = (a_1, \ldots, a_n)$ that an agent α performs to an agent β. Let $T^r = \langle T_1^r, \ldots, T_m^r \rangle$ be the fuzzy scale for the evaluation of the investment value $r_{\alpha\beta}$ by α. Then, the fuzzy evaluation of the investment value $r_{\alpha\beta}$ is determined by the MAX-MIN inference rule applied over a rule base "**If** ... **Then**" of type:[8]

If a_1 **is** T_j^1 **and** a_2 **is** T_l^2 **and** $\ldots$ **and** a_n **is** T_p^n
Then $r'_{\alpha\beta}$ **is** $T_q^{\prime r}$

with $T_j^1 \in T^1, T_l^2 \in T^2 \ldots T_p^n \in T^n, T_q^r \in T^r$.

For the evaluation of a certain rule, firstly we evaluate each condition of type a_i **is** T_k^i, with $i = 1, \ldots, n$, as been $\mu_i(V_{nor}(a_i))$, that is, the fuzzy evaluation of the normalized value of the service attribute a_i using the fuzzy scale T^i, where $V_{nor}(a_i)$ is determined by Eq. (7). From those values, the evaluation of $r'_{\alpha\beta}$ **is** $T_q^{\prime r}$ is obtained as:

$$\min\{\mu_1(V_{nor}(a_1)), \ldots, \mu_n(V_{nor}(a_n))\}. \quad (8)$$

The investment fuzzy value $r_{\alpha\beta}$ is then determined from the evaluation of all rules of such type. For each term T_v^r, with $v = 1, \ldots, m$, one obtains then value

$$\max\{T_v^{\prime r}, T_v^{\prime\prime r}, \ldots, T_v^{\omega r}\}, \quad (9)$$

where $\omega \leq k_1 \times \ldots \times k_n$, and k_i is the cardinality of the fuzzy scale T^i for the evaluation of each service attribute a_i. These values give rise to a cut in the linguistic term T_v^r, and, therefore, a fuzzy region in T^r. In this region, if necessary, according to the application, it is possible to apply a method of *defuzzification* (e.g., the *centroid* method [21]), in order to obtain a crisp value for the investment $r_{\alpha\beta}$.

The same methodology is applied to obtain the fuzzy satisfaction value $s_{\beta\alpha}$ of the agent β for receiving the service $\mathbb{S}_s(\beta) = (b_1, \ldots, b_n)$, with attributes $b_1, \ldots, b_n$.

4 A Case Study

Consider a pizza delivery system, where there are two points of view for the evaluation: (i) of agent α, which performs the service of delivering a pizza for an agent β, generating an investment value $r_{\alpha\beta}$, where the aspects to be considered in the evaluation of

[8] In this paper, we consider the Gödel t-norm. [21]

Table 1. Rules for the fuzzy evaluation of the investment value $r_{\alpha\beta}$ and the satisfaction value $s_{\beta\alpha}$

$r_{\alpha\beta}$ $(d \times cl)$	low	medium	high	$s_{\beta\alpha}$ $(t \times ct)$	cheap	medium	expensive
close	low	medium	medium	fast	high	high	medium
far	medium	medium	high	medium	high	medium	low
very-far	medium	high	high	slow	medium	low	low

Table 2. Equations of the fuzzy material equilibrium of the stage $I_{\alpha\beta}$ $(r_{\alpha\beta} \times s_{\beta\alpha})$

$I_{\alpha\beta}$	low	medium	high
low	$r_{\alpha\beta} = s_{\beta\alpha}$	$r_{\alpha\beta} < s_{\beta\alpha}$	$r_{\alpha\beta} << s_{\beta\alpha}$
medium	$r_{\alpha\beta} > s_{\beta\alpha}$	$r_{\alpha\beta} = s_{\beta\alpha}$	$r_{\alpha\beta} < s_{\beta\alpha}$
high	$r_{\alpha\beta} >> s_{\beta\alpha}$	$r_{\alpha\beta} > s_{\beta\alpha}$	$r_{\alpha\beta} = s_{\beta\alpha}$

the service are the *complexity* (cl) of the pizza elaboration process and the *distance* (d) of β's residence, where the pizza should be delivered; (ii) of agent β, which receives the service, generating a satisfaction value $s_{\beta\alpha}$, where the aspects to be considered in the evaluation are the *delivery time* (t) and the *cost* (ct) (the amount paid by β for the pizza). Then, this service is defined by: $\mathbb{S}_r(\alpha) = \{d, cl\}$ and $\mathbb{S}_s(\beta) = \{t, ct\}$.

Agents with different personality factors are considered, and we analyze in detail the evaluations realized by α_3 (with egoism personality factor $\gamma_{\alpha_3} = 0.3$) and β_{15} (with altruism personality factor $\gamma_{\beta_{15}} = 1.5$).

4.1 Fuzzy Evaluation of the Investment Value

For the fuzzy evaluation of the investment value of the pizza delivery service $\mathbb{S}_r(\alpha) = \{d, cl\}$, let the fuzzy scales for the evaluation of the service attributes d (delivery distance) and cl (complexity), and of the investment value $r_{\alpha\beta}$, be given, respectively, by:

$T^d = \langle \texttt{very-close}, \texttt{close}, \texttt{far}, \texttt{very-far} \rangle$; $T^{cl} = \langle \texttt{low}, \texttt{medium}, \texttt{high} \rangle$;

$T^r = \langle \texttt{low}, \texttt{medium}, \texttt{high} \rangle$.

The rule base for the fuzzy evaluation of the service investment value $\mathbb{S}_r(\alpha) = \{d, cl\}$, based on the fuzzy evaluation of its service attributes, is shown in Tab. 1.

Consider an instance of the service $\mathbb{S}_r(\alpha) = \{d, cl\}$, whose attributes are measured as $V(cl) = 40$ and $V(d) = 5$. The normalization process of these service attributes, realized by the egoist agent α_3 (with egoism factor $\gamma_{ego} = 0.3$), using Eq. (7), results, respectively, in $V_{nor}(cl) = 6.80$ and $V_{nor}(d) = 5.66$, for $\max_{cl} = 70$ and $\max_d = 15$. For the tolerant agent α_{11} (with tolerance factor $\gamma_{tol} = 1.1$), the normalization process of those same attribute values results in $V_{nor}(cl) = 3.6$ and $V_{nor}(d) = 3$, respectively.

Figures 1 and 2 show, respectively, the fuzzy evaluations of the normalized values $V_{nor}(cl) = 3.6$ and $V_{nor}(d) = 3$ of the attributes cl and d, according to their respective fuzzy scales[9], realized for agentes with different personality factors. In particular, compare the neutral evaluation realized by the tolerant agent α_{11} with the overvalued evaluation realized by the egoist agent α_3, in order to see the influence of the personality factors in the agent evaluations. The fuzzy evaluations of the service attributes cl and d, realized by the egoist agent α_3, result, respectively, in the following fuzzy values:

[9] In all fuzzy scales we adopt $N = 10$.

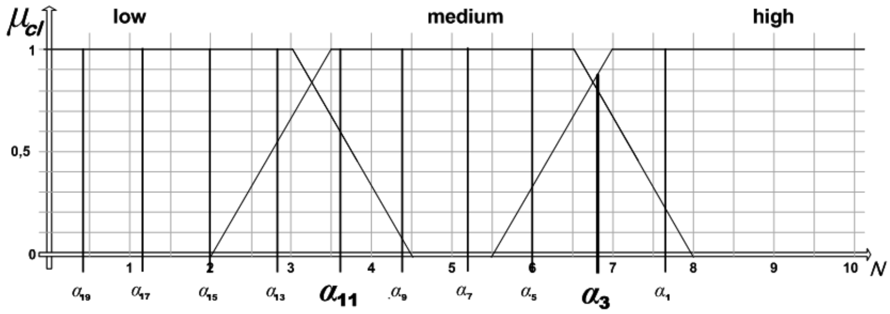


Fig. 1. Fuzzy evaluation of the service attribute *complexity* (cl)

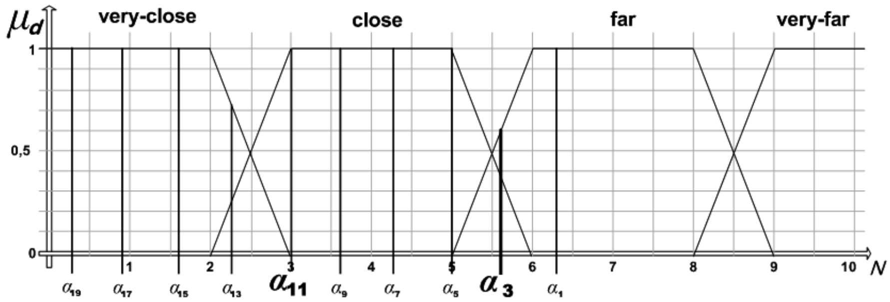


Fig. 2. Fuzzy evaluation of the service attribute *delivery distance* (d)

"cl **is** `medium`" with degree 0.8 and "cl **is** `high`" with degree 0.86
"d **is** `close`" with degree 0.34 and "d **is** `far`" with degree 0.66.

With those fuzzy values, we then evaluate the rules of Tab. 1 that are applicable in this case. Then, using Eq. (9), the fuzzy value of the investment of α_3 is obtained as:

"$r_{\alpha_3\beta}$ **is** `medium`" with degree $\max\{0.34, 0.34, 0.66\} = 0.66$ and
"$r_{\alpha_3\beta}$ **is** `high`" with degree 0.66,

generating the fuzzy region of Fig. 3. The output crisp value of the investment, if necessary, can be obtained by using the centroid method, resulting in 6.852866.

4.2 Fuzzy Evaluation of the Satisfaction Value

For the fuzzy evaluation of the satisfaction value of the pizza delivery service $\mathbb{S}_s(\beta) = \{t, ct\}$, let the fuzzy scales for the evaluation of the service attributes t (delivery time) and ct (cost), and of the satisfaction value $s_{\beta\alpha}$, be given, respectively, by:

$$T^t = \langle \texttt{fast}, \texttt{medium}, \texttt{slow} \rangle; \;\; T^{cl} = \langle \texttt{cheap}, \texttt{medium}, \texttt{expensive} \rangle;$$
$$T^s = \langle \texttt{low}, \texttt{medium}, \texttt{high} \rangle.$$

The rule base for the fuzzy evaluation of the service investment value $s_{\beta\alpha} = \{t, ct\}$, based on the fuzzy evaluation of the respective service attributes, is presented in Tab. 1.

Consider an instance of the service $s_{\beta\alpha} = \{t, ct\}$, whose attributes are measured as $V(t) = 30$ e $V(ct) = 40$. The normalization process of these attributes, realized by the

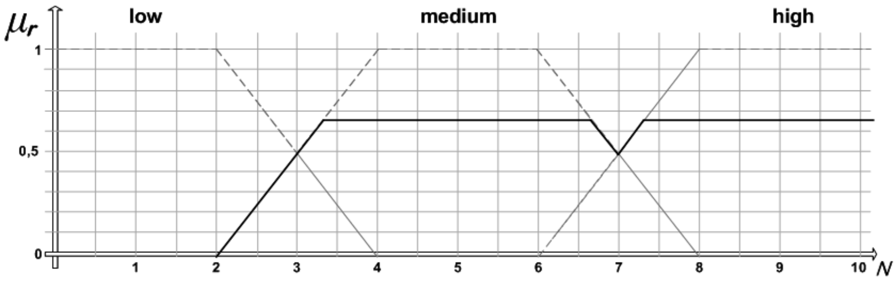


Fig. 3. Geometric representation of α_3's fuzzy investment value

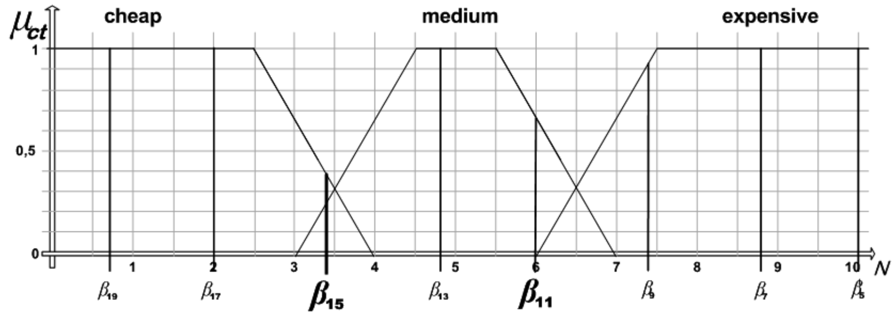


Fig. 4. Fuzzy evaluation of the service attribute *cost* (ct)

altruist agent β_{15} (with altruism factor $\gamma_{alt} = 1.5$), using Eq. (7), results, respectively, in $V_{nor}(t) = 2.5$ and $V_{nor}(ct) = 3.33$, for $\max_t = 60$ and $\max_{ct} = 40$. For the tolerant agent β_{11} (with tolerance factor $\gamma_{tol} = 1.1$), the normalization process of those same attribute values results in $V_{nor}(t) = 4.50$ and $V_{nor}(ct) = 5.99$, respectively.

Figures 4 and 5 show, respectively, the fuzzy evaluations of the normalized values $V_{nor}(t) = 4.50$ and $V_{nor}(ct) = 5.99$ of the attributes t and ct, according to their respective fuzzy scales, realized for agentes with different personality factors. In particular, compare the neutral evaluation realized by the tolerant agent β_{11} with the underestimated evaluation realized by the altruist agent β_{15}, which is another example of the influence of the personality factors in the agent evaluations.

The fuzzy evaluations of the service attributes t and ct, realized by the altruist agent β_{15}, result, respectively, in the following fuzzy values:

"t **is** `fast`" with degree 0.75 and "t **is** `medium`" with degree 0.25,
"ct **is** `cheap`" with degree 0.44 and "ct **is** `medium`" with degree 0.22.

With those fuzzy values, we then evaluate the rules of Tab. 1 that are applicable in this case. Then, using Eq. (9), the fuzzy value of the satisfaction of β_{15} is obtained as:

"$s_{\beta_{15}\alpha_3}$ **is** `high`" with degree $\max\{0.44, 0.22, 0.25\} = 0.44$ and
"$s_{\beta_{15}\alpha_3}$ **is** `medium`" with degree 0.22,

generating the fuzzy region of Fig. 6. The output crisp value of the satisfaction, if necessary for any application, can be obtained by the centroid method, resulting in 2.866667.

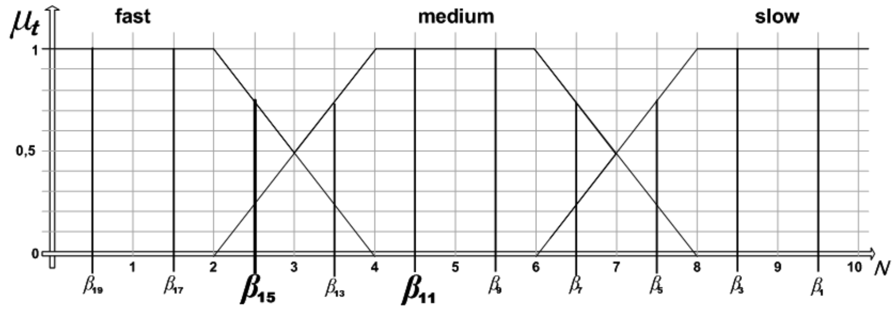


Fig. 5. Fuzzy Evaluation of the service attribute *delivery time* (t)

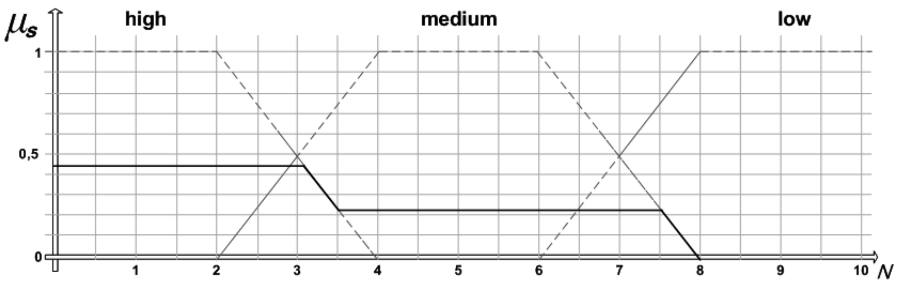


Fig. 6. Geometric representation of β_{15}'s fuzzy satisfaction value

4.3 Equations of Fuzzy Equilibrium

With the results obtained in Sections 4.1 and 4.2, we evaluate the *fuzzy material equilibrium* in the first stage of a social exchange between the agents α and β, generalizing the first equality of **Rule** $\mathbf{I}_{\alpha\beta}$ (Eq. (1)), whose rule base is given in Tab. 2. In this table, it is possible to observe the notion of *fuzzy equilibrium* (the case in which the investment of agent α and the satisfaction of agent β are "equivalent" with a certain degree) or the two notions of *fuzzy disequilibrium* (the cases in which the investment of α is either "less than" or "much less than" the satisfaction of β with a certain degree, or vice-versa). Let the fuzzy investment and satisfaction values be given, respectively, by:

"$r_{\alpha_3\beta_{15}}$ **is** `medium`" with degree 0.66 and "$r_{\alpha\beta}$ **is** `high`" with degree 0.66, and
"$s_{\beta_{15}\alpha_3}$ **is** `medium`" with degree 0.22 and "$s_{\beta_{15}\alpha_3}$ **is** `high`" with degree 0.44.

We evaluate the rules of Tab. 2 that are applicable in this case, using Eq. (9). Then, the equations of fuzzy material equilibrium of the stage $I_{\alpha_3\beta_{15}}$ are evaluated as follows, generating the fuzzy region of Fig. 7:

"$I_{\alpha_3\beta_{15}}$ **is** $r_{\alpha_3\beta_{15}} < s_{\alpha_3\beta_{15}}$" with degree 0.44 and
"$I_{\alpha_3\beta_{15}}$ **is** $r_{\alpha_3\beta_{15}} = s_{\alpha_3\beta_{15}}$" with degree $\max\{0.22, 0.44\} = 0.44$ and
"$I_{\alpha_3\beta_{15}}$ **is** $r_{\alpha_3\beta_{15}} > s_{\alpha_3\beta_{15}}$" with degree 0.22.

5 Conclusion

Social exchange values have been applied in different contexts in MAS [5–12]. However, since social exchange values are qualitative values for subjective concepts, one

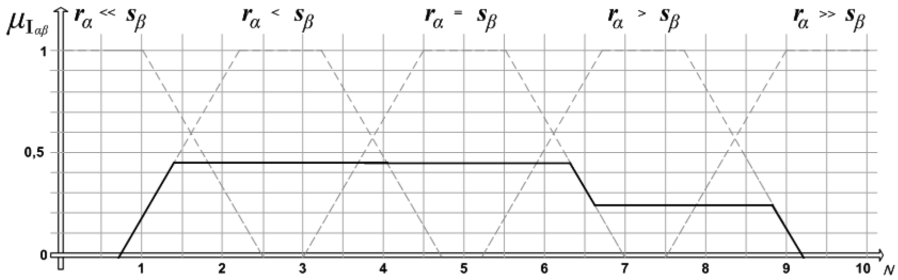


Fig. 7. Fuzzy material equilibrium of the stage $I_{\alpha_3\beta_{15}a}$

has that their computational representation is not so trivial, as discussed in Sect. 1. On the other hand, Fuzzy Sets constitute the oldest and most reported soft computing paradigm, been well-suited to model different forms of uncertainties, ambiguities and subjectiveness, often encountered in real life. Fuzzy Logic has been applied in Social Simulation in order to cope with complex agent-based models for the simulation of human behavior [23, 24], and, in particular, human personality traits [17, 18].

This work introduced an approach based on Fuzzy Logic for the evaluation of the material social exchange values generated in the first stage of a social exchange, with an application in personality-based MAS, where social exchange personality traits are defined as the way the agents qualitatively evaluate the services performed and/or received, after the decision on accepting or refusing an exchange proposal. The equations of fuzzy material equilibrium associated to this stage were also established. A case study was analyzed allowing to evaluate the potentiality of the proposed approach.

Although for simplicity just a small number of service attributes were considered, we remark that the approach is flexible enough to allow the addition of other different attributes. Moreover, the approach can be applied for the subjective, imprecise or vague evaluation of any other kind of service performable by agents.

Ongoing work is concerned with the fuzzy evaluation of the virtual values, which we intend to do using linguistic modifiers [21], and then the three equations of fuzzy equilibrium may be established. Future work will be concerned with the development of a fuzzy perception mechanism for BDI agentes [25], in order to incorporate fuzzy perception in the social exchange simulator described in [9].

Acknowledgements. We thank the support received from CNPq (Proc. 473201/07-0, 307185/07-9, 304580/07-4, 307879/06-2).

References

1. Dignum, V., Dignum, F.: Coordinating tasks in agent organizations or: Can we ask you to read this paper? In Noriega, P. et al., eds.: Coordination, Organizations, Institutions, and Norms in Agent Systems II. Number 4386 in LNAI. Springer, Berlin (2007) 32–47
2. Gilbert, N.: Agent-based Models. SAGE, Los Angeles (2008)
3. Gilbert, N., Troitzsch, K.G.: Simulation for the social scientist. Open University Press, Milton Keynes (2005)

4. Piaget, J.: Sociological Studies. Routlege, London (1995)
5. Costa, A.C.R., Dimuro, G.P.: A minimal dynamical organization model. In Dignum, V., ed.: Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models. IGI Global, Hershey (2009) 419–445
6. Dimuro, G.P., Costa, A.C.R., Gonçalves, L.V., Hübner, A.: Centralized regulation of social exchanges between personality-based agents. In Noriega, P. et al., eds.: Coordination, Organizations, Institutions and Norms in Agent Systems II. Number 4386 in LNAI, Berlin (2007) 326–343
7. Grimaldo, F., Lozano, M., Barber, F.: Coordination and sociability for intelligent virtual agents. In Sichman, J. et al., eds.: Coordination, Organizations, Institutions, and Norms in Agent Systems III. Number 4870 in LNAI, Berlin, Springer (2008) 58–70
8. Grimaldo, F., Lozano, M., Barber, F., Vigueras, G.: Simulating socially intelligent agents in semantic virtual environments. Knowledge Engineering Review **23**(4) (2008) 369–388
9. Pereira, D.R., Gonçalves, L.V., Dimuro, G.P., Costa, A.C.R.: Towards the self-regulation of personality-based social exchange processes in multiagent systems. In Zaverucha, G. et al., eds.: Proc. of 19th Brazilian Symposium on Artificial Intelligence. Volume 5249 of LNAI. Springer, Berlin (2008) 113–123
10. Rodrigues, M.R., Luck, M.: Evaluating dynamic services in bioinformatics. In Klusch, M. et al., eds.: Cooperative Information Agents X. Number 4149 in LNCS. Springer (2006) 183–197
11. Rodrigues, M.R., Luck, M.: Cooperative interactions: an exchange values model. In Noriega, P. et al., eds.: Coordination, Organizations, Institutions and Norms in Agent Systems II. Number 4386 in LNAI, Berlin, Springer (2007) 16–23
12. Rodrigues, M.R., Luck, M.: Effective multiagent interactions for open cooperative systems rich in services. In: Proc. of 8th Intl. Conf. on Autonomous Agents and Multiagent Systems, Budapest (2009) 1273–1274
13. Homans, G.C.: The Human Group. Harcourt, Brace & World, New York (1950)
14. Conte, R., Castelfranchi, C.: Cognitive and Social Action. UCL Press, London (1995)
15. Castelfranchi, C., Rosis, F., Falcone, R., Pizzutilo, S.: A testbed for investigating personality-based multiagent cooperation. In: Proc. of the Symp. on Logical Approaches to Agent Modeling and Design, Aix-en-Provence (1997)
16. Castelfranchi, C., Rosis, F., Falcone, R., Pizzutilo, S.: Personality traits and social attitudes in multiagent cooperation. Applied Artificial Intelligence **12** (1998) 649–675
17. Ghasem-Aghaee, N., Ören, T.I.: Towards fuzzy agents with dynamic personality for human behavior simulation. In: Proc. of the 2003 Summer Computer Simulation Conference, Montreal, July 20-24, 2003, San Diego, SCS (2003) 3–10
18. Ören, T.I., Ghasem-Aghaee, N.: Personality representation processable in fuzzy logic for human behavior simulation. In: Proc. of the 2003 Summer Computer Simulation Conference, Montreal, July 20-24, 2003, San Diego, SCS (2003) 11–18
19. Carbonell, J.G.: Towards a process model of human personality traits. Artificial Intelligence **15**(1,2) (1980) 49–74
20. Moore, R.E.: Methods and Applications of Interval Analysis. SIAM, Philadelphia (1979)
21. Ross, T.J.: Fuzzy Logic, with Engineering Applications. John Wiley, West Sussex (2004)
22. Zadeh, L.A.: Fuzzy sets. Information and Control **8** (1965) 338–353
23. Hassan, S., Garmendia, L., Pavón, J.: Agent-based social modeling and simulation with fuzzy sets. In Corchado, E. et al., eds.: Innovations in Hybrid Intelligent Systems. Number 44 in Advances in Soft Computing. Springer, Berlin (2008) 40–47
24. Sabeur, E., Denis, G.: Human behavior and social network simulation: Fuzzy sets/logic and agents-based approach. In: Proc. of the 2007 Spring Simulation Multi-Conference, Norfolk, 2007, San Diego, SCS (2007) 102–109
25. Wooldridge, M.: Reasoning about Rational Agents. MIT Press, Cambridge (2000)

A Multi-agent Model for Panic Behavior in Crowds

Robson dos Santos França, Maria das Graças Bruno Marietto, and Margarethe Born Steinberger

Federal University of ABC, Santo André, São Paulo, Brazil,
{robson.franca,graca.marietto,mborn}@ufabc.edu.br

Abstract. This paper presents a conceptual model for the panic in crowds' phenomenon. The proposed model is based on social science theories related to collective behavior. Such model could be applied in two dimensions: (i) to assist in proposing new structures or variations for collective panic situations, checking the viability of their existence and inner-working; (ii) to get a better understanding of social, anthropological and psychological foundations, etc. which drive and maintain the panic in crowds type of collective behavior. One of the challenges to be faced by this study is the integration of different theories in a coherent and robust way, since many of them have contradictory positions. Besides, thanks to the fact that these theories show a higher degree of abstraction, adjustments will be made in order to achieve the computability of the proposed conceptual model.

1 Introduction

The study of human groups' behavior, which is controlled by institutionalized rules and traditions, is the focus of Sociology. In turn, the research field called Collective Behavior deals with situations which human behavior is not based on current and socially accepted cultural norms. Specifically, the panic in crowds' phenomenon is a kind of collective behavior which involves a certain amount of people being exposed to a dangerous situation. In such situation there is always an imminent risk and a feeling of urgency for action.

Studies regarding panic in crowds are important because they assist in achieving a clearer and detailed understanding of the social theories, and could be used as a basis of algorithms which will uncover new ways of solving computational problems. Also, the studies' results could help in predicting the birth of collective panic and/or highlighting actions that might reduce material and mostly human losses. The understanding of this behavior makes it possible to design safer and more efficient public places such as theaters, movie theaters, residential buildings and stadiums.

However, even with the importance of studies about panic in crowds phenomenon there is a restrict amount of simulations that deals with this kind of collective behavior [1]. In order to help in filling this gap, a multi-agent conceptual model for collective panic is proposed. The first contribution of this model is the fact that it could be used for simulation building which offers a more precise comprehension of the phenomenon dynamics, taking some aspects into consideration such as social, psychological and anthropological elements. The second contribution refers to the application of Multi-agent Systems (MAS) theory for model building because MAS allow us to establish a relationship between a program and an individual, so it will be to simulate an artificial world formed by interactive computational entities. This multi-agent modeling process is able to transpose from a crowd in panic behavior to a similar conceptual model, with a proper theoretical-technical framework for modeling and understanding complex social processes such as coordination, formation of coalitions and groups, conventions and norms' evolution, micro-to-macro linking, and so on.

The remaining of this paper is structured as follows: in Section 2 there is a brief description of the theory which this paper is based on. Section 3 is the core of this paper, featuring a proposal for a conceptual model for panic in crowds. Related works and a single comparison can be found in Section 4. Finally, in Sect. 5, some conclusions are presented.

2 Collective Behavior

The main focus of the sociological studies is the culturally-based human groups, more specifically certain groups that behave according to well-established and institutionalized rules. On the other hand, the field of collective behavior deals with collectivities that contradict or reinterpret society norms and standards. Among the classical definitions for collective behavior we cite [2]: "*collective behavior is the set of social behaviors which the usual conventions stop driving the actions and the individuals transcend, exceed or collectively subvert the standards and the institutionalized structures*".

In the collective behavior field, there is a consensus in defining at least three classes of this kind of behavior: the public, the mass and the crowd. The crowd is a localized collective behavior which individuals are close enough to make a physical contact. Some authors classify the crowds in: (i) Active crowd, such as mutinies, lynching mobs and rebellions; (ii) Casual crowd, such as crowds that get together to watch a display window being decorated; (iii) Conventional crowd, for instance an audience for a soccer game or any other recreational activity; (iv) Expressive crowd, for instance the dancing crowds at carnaval and some religious groups; (v) Panic crowd, which is formed when people are exposed to a dangerous situation and that leads them to create the perception of need to stay away from danger in a social and shared way, such as earthquakes and fires. The next subsection will cover the structural and functional aspects of the panic crowd behavior.

2.1 Panic Crowd Behavior in an Interactionism Approach

The theoretical model presented herein is based on the emergent-norm interactionist approach presented in the Sec. 4. Its basic scheme is displayed at Figure

1. The steps were placed sequentially just for didactical purposes. However, a person could skip, repeat or take a different step order.

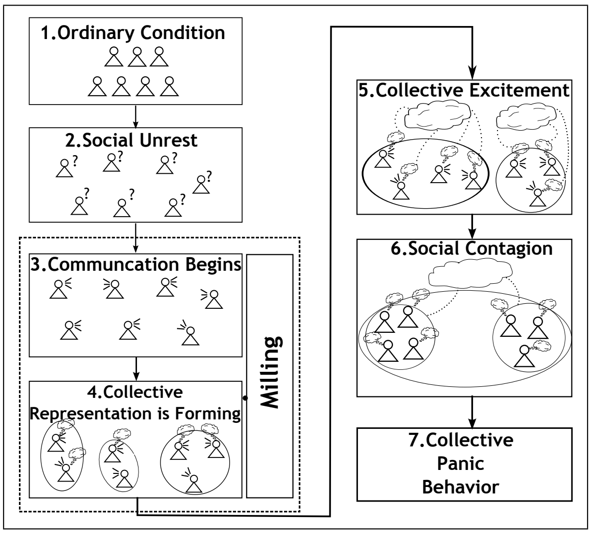


Fig. 1. Panic in Crowds Collective Behavior Overview.

Initially, the individuals are in an **ordinary condition** which norms and social structures are aligned with what is socially accepted. When is perceived a disorder on the established social structure the individuals feel uneasy and apprehensive, trying to understand the non-structured and ambiguous situation that just happened. A disturb is an event that presents itself as an imminent threat to the agent's life, like a fire alarm or a smoke cloud, and such event draws the person's attention and forces him to act, leading to a **social unrest**. After that, the individuals start looking for information that could help them in redefining the current context. In these conditions, people become more susceptible to rumors, since there is a feeling of uncertain and insecurity. The conventional behavior starts to break down. Since there is a need to understand the situation, people engage themselves in a **milling** process, looking the other individuals' reactions and comparing those reactions with your own set of expectations. In this process, a need for searching a sanctioned and socially-built meaning arises, into a relatively non-structured situation [2]. The milling is important because draws the individual attention to the situation and actions from the collectivity, taking himself out of focus e pointing his attention to the other agents actions.

With a higher focus on themselves, the individuals respond faster and more direct to each other, preparing the environment for the formation of a shared understanding of what is happening. Then, the collectivity pass to the **collective excitement** stage, when individual representations are combined and synthesized by the group, helping in the formation of a collective representation/image of the situation. This shared representation increases the individual's susceptibility, reducing his capability of creating different impressions from the collectivity. That way he could enter in a line of conduct socially forbidden that he could not conceive and execute, such as pushing and running over people. The **social contagion** is an intense form of collective excitement, promoting a fast dissemination of the collectively formed representation, empowering the social cohesion and making room for a collective action. Finally, when the individuals have built a collective representation of the situation it is possible to choose and execute actions. At this moment, the collective crisis created by a struggle for survival comes to an apex, and the **collective panic** is installed. Since the participants don't share traditional expectations about the way they should behave, the results are uncertain.

3 A Model for Panic Behavior

In order to build the proposed model for this paper, the following elements were taken into consideration: (i) the agent's architecture, which represents a person in a panic situation; (ii) three environments where the agents' main interactional aspects will happen; (iii) a socially-built system called Group Mind, which defines how each individual representation of the situation can be socially changed and synthesized by the collectivity, in order to make a shared context.

3.1 Environments Description

The agents of the model proposed by this paper have access to three environments (Fig. 2): Physical Environment, Communication Environment and Group Mind. The General Environment covers these three environments.

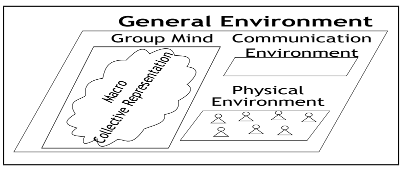
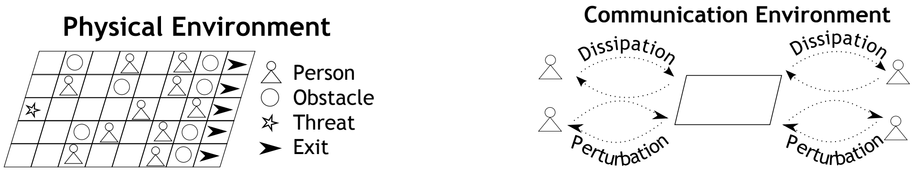


Fig. 2. General Environment and its Components.

The Physical Environment (Fig. 3(a)) is the place where the agents will physically interact with each other and with other objects such as furniture, obstacles, walls, etc. There are specific points where the threat will appear, as well as exit points.

In the Communication Environment (see Fig. 3(b)) messages will be dissipated and the agents can be disturbed by them, according to parameters of cybernetics theories inspired by Norbert Wiener [3]. There is no direct information



(a) Physical Environment (b) Communication Environment

Fig. 3. Physical and Communication Environment

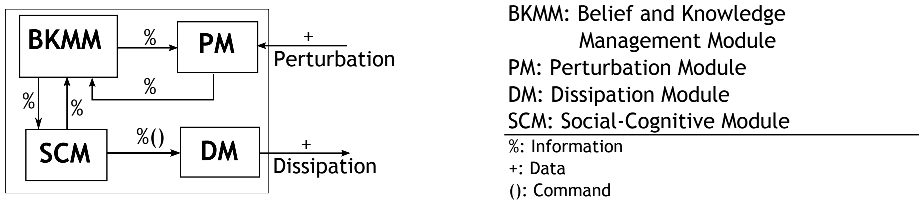
exchange among the agents. Instead, whenever an agent needs to communicate the messages are dissipated into the environment. In the same way, each agent is autonomous to check the Communication Environment in order to define whether the message is relevant for reaching its goals.

The Group Mind manages a collectivity-built framework made of expectations networks that are formed by the repetition of similar events. Expectations are incidents of expected behaviors by an agent in spite of a situation, and these expectations guide their actions [4]. Agents in society expect certain types of behavior from the others, and they deal with different types of behavior as expectation deviations. In this proposed model, each person makes its own expectations network, and it is stored in the Micro Collective Representation Module (Subsection 3.2, item B). Finally, the current situation is qualitatively-new information that cannot be identified by just looking at one of the forming units (the individual expectations networks). This emerging process of current context's shared representation is controlled by the Group Mind, which makes abstractions, generalizations, averages and schemes from the individual expectations. The resulting social structure (a net of shared expectations) is stored in the Macro Collective Representation.

In order to make the situation being processed and interpreted it is required that its base elements are socially built and act like significant symbols. In this model such symbols are: a goal, an object and a line of action. The goal restricts the actions allowed for the agent. For each goal there is a set of possible actions that the agent might engage. The object refers to the entity which the agents will be acting upon. Objects can be physical (a fire blaze), an abstract concept (sense of justice, homeland and family) or a junction of these two things. A line of action is a suggestion of how to act in an ambivalent and uncertain situation.

3.2 Architecture of the Person Agent

The Person agent represents an individual in a physical environment that will pass through a panic situation. Its architecture is formed by four modules, as seen at Fig. 4: Perturbation Module, Belief and Knowledge Management Module, Social-Cognitive Module and Dissipation Module. These modules are described below.

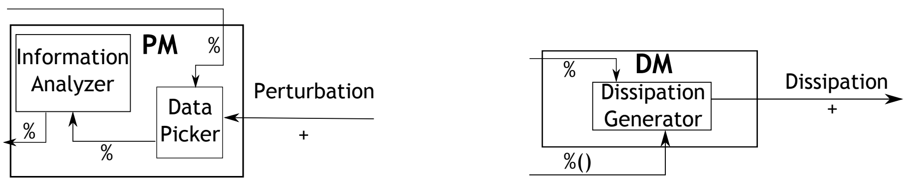


(a) Person Agent. (b) Agent Modules.

Fig. 4. Person Agent Architecture Overview.

A. Perturbation Module (PM) The Figure 5(a) shows PM's architecture, which is formed by Data Picker (DP) and the Information Analyzer (IA). The DP constantly scans the Communication Environment, looking for disturbing messages. In order to do so it analyzes - considering some information from the Rule Set (see item B) - whether the message is related to individual's action on reactive, functional or dynamic perspective.

When a message is accepted by the DP, it is sent to the Information Analyzer which, at first, checks the information formatting (syntax) in an Agent Communication Language. After that, a semantic analysis is performed, in order to check the way the information was externalized as a gesture or speech, and the mood (lovely, aggressive, neutral) [4]. After the syntactical and semantical analysis, the information is stored in the Belief Base or in the Knowledge Base (see item B) and that happens whether the information can be proved by physical evidences perceived by the agent (knowledge), or the information was caught from other agent without evidence (belief).

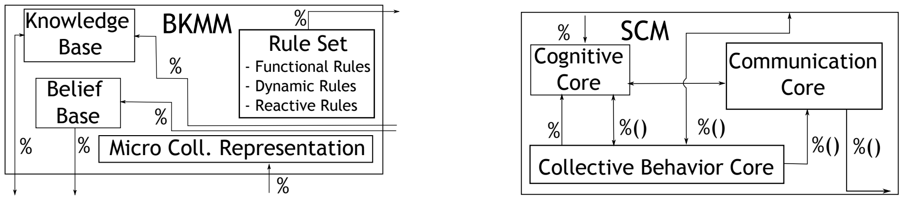


(a) Perturbation Module (b) Dissipation Module

Fig. 5. Perturbation and Dissipation Modules

B. Belief and Knowledge Management Module The belief and knowledge management module works as a manager of the information bases required for the proper agent operation. This module is comprised of Knowledge Base, Rule Set, Belief Base and Micro Collective Representation. Figure 6(a) shows an overview of this module.

The Belief Base takes the information that has not been proved by the agent. For instance, if the agent decides to leave the environment and he does



(a) Belief and Knowledge Management Module.

(b) Social-Cognitive Module.

Fig. 6. Belief and Knowledge Management Module and Social-Cognitive Module

not know the exit position, he will inquire the nearby agents. The information provided by the other agents will be treated as a belief until the agent gets closer to an exit and testifies that the information is logical. On the other hand, the Knowledge Base deals with three kinds of information. At first there are variables that establish the agent's inherent features. They are called *Internal State Variables* and are presented in Table 1. The second group of variables, called *Physical-External Variables*, describes the information that is directly affected or is perceived from the physical environment. A list of these variables can be found in Table 2. Lastly there are *Social State Variables* which represent the agent's social condition and they are affected by the agents' interactions. Some of these variables are presented in Table 3.

Internal State Variables	
Name	**Description**
health	The agent's vital status. It depends on how much pressure the agent is taking.
speed	The displacement length in each time step (meters per second).
expEvent	The level of experience for the simulation's kind of panic event. It affects most of the agent's social decisions.
nervousness	The ratio between idealSpeed and speed. It can trigger agent's permissiveness increasing.
uncertainty	For each information available, how much certainty the information has.

Table 1. Internal State Variables

Physical-External Variables	
Name	**Description**
temperature	Localized temperature measured by the agent (Celsius).
pressure	How much pressure is forcing the agent (Newtons per meter). It directly affects health
dirThreat	The direction to the threat, for escaping procedures.
hazardLevel	It is based on temperature, pressure, free space availible and experience, how dangerous the situation is.

Table 2. Physical-External Variables

Social State Variables	
Name	**Description**
SUSCEPTIBILITY	It defines how much the agent is susceptible to the other agents' opinions.
PERMISSIVENESS	The level of acceptance of the socially inadequate actions in spite of the panic situation.
CBSTATUS	The agent's collective behavior status. Which collective behavior step the agent is currently engaged.

Table 3. Social State Variables

The rule set defines the agent operation in his life cycle. These rules are classified in: (i) Functionals, which establish the agent's identity and the pursue of his goals, such as Walking and checking its own health; (ii) Reactives, related to agent's survival and usually time-constrained, such as running away from the fire and throwing water at the fire; (iii) Dynamics, which rely on learning, such as learning how to put out the fire or how to safely escapes from the environment.

The MICRO COLLECTIVE REPRESENTATION stores the representation/image of the current situation locally and individually created by each agent, but taking a mutual feedback process among the agency elements into consideration. The tendency is that during this collective process a symbol convergence will happen and that will make common imaginary and unified actions. Thus, these individual imaginaries will help in the formation of the MACRO COLLECTIVE REPRESENTATION, located at the GROUP MIND (see item C).

C. Social-Cognitive Module This module is responsible for coordinating the agent PERSON other modules' actions, managing their autonomous and private process. It is made of the following cores (see Fig. 6(b)): COGNITIVE CORE (COGC), COLLECTIVE BEHAVIOR CORE (CBC) and COMMUNICATION CORE (COMC). The CogC stands in continuous processing, managing information and guiding actions so the agent can pursue its goals. As long as the individual is in a situation that does not pose as a threat to its life (see Fig 1, item 1), the CogC leads the agent to a certain behavior that it accepts the rules and roles established in the society. However, if an event that poses a threat is triggered, the CogC passes his duties to the CBC. This replacement makes the agent act in a collective way, engaging in the collective behavior.

In order to quantify the threat, the agent checks his experience for this kind of event (fire) and the hazardous level he assigned for that situation. At this moment, the functional and dynamic rules remain strong, and the reactive rules remain weak. Besides, the individual is lacking in information to follow a certain line of action. In order to go to next step (social unrest), the uncertainty level assigned for the situation must be higher than a certain threshold, which implies that the agent doesn't know what is happening, so he feels that he needs more information about the event.

Going into a social unrest (Fig. 1, item 2), the agent searches for information that helps him in analyzing what is happening. Its uncertainty level rises since it is unable to understand the event by himself, so it engages in the milling process (Fig. 1, item 2). At this point, the agent increases his communication with the others, trying to build his own Micro Collective Representation (Fig. 1, item 4). At the same time the personal value variable is affected, increasing the agent's acceptance for external thoughts. The agents become less aware of themselves as individuals and more aware of the others. The dynamic rules (e.g. learning how to perform an operational task) become weaker because the sense of urgency is stronger in a dangerous situation than in an ordinary condition.

Collective excitement (Fig. 1, item 5) begins when the permissiveness starts to interfere on the agent's choices. At this point the agents can choose socially unacceptable actions, such as running over people. Functional rules loose their strength (mostly because permissiveness is rising) and reactive rules get stronger. When the agents define a goal and an object for action, the macro collective representation starts to develop and to establish. This step is called social contagion (Fig. 1, item 6) because the communication and interaction among agents are in such condition that some individuals - not yet engaged in collective behavior - are attracted by the group, and they are induced to be part of this process. The reactive rules become the strongest rules for the agent. Since the permissiveness is high, the agents can choose actions treated as socially improper. Dynamic rules, such as learning how to escape are limited.

Finally, the collective panic behavior (Fig. 1, item 7) is installed when the agents choose a line of action to be followed by the collectivity. The agents are fully engaged in the collective behavior, and they will stay on that condition until they do not feel threatened. The ComC receives all requests for communication from the CogC and the CBC and puts those requests in a queue for being dispatched by the Dissipation Module.

D. Dissipation Module (DM) Whenever the agent needs to send a message to the other agents, this module is requested. The DM (Fig. 5(b)) receives the message from the Communication Core. Inside the DM the Dissipation Generator (DC) prepares the message to be dissipated on the environment by encoding, adding other relevant data, such as the message format (using an ACL) and how it should be expressed in the environment: if it is a gesture or a speech and how the message mood is (lovely, cold, etc.).

4 Comparative Analysis of Panic Behavior Models

After analyzing Collective Behavior literature, three theoretical approaches could be found to explain panic crowd phenomenon. The first one, called **Contagion Theory**, holds that the individual in a crowd loses his conscious personality and obeys all suggestions from the crowd, similar to a hypnotized person following a hypnotizer. Social sciences researchers such as Le Bon [5] followed this approach. The simulation model presented in [6],[7] were based on Contagion Theory for their modeling.

There is a second approach for collective behavior which deals with it in a more rational and objective-aware fashion and it is based on the Symbolic Interactionism theory, which considers that collective behavior is an outcome of the interactions among individuals, which are able to evaluate the information they receive and to decide on its usage at the present situation. In the **Emergent Norm Theory** [2] (employed by this paper) the agents assign positive or negative values to the information they receive and that leads to the development of an interactive cognition. This approach analyzes the agents' micro-properties that help in the social systems' formation. It also analyzes the behavior patterns at the group level. As examples of computational works based on this approach there is [8].

A variant of this second approach, commonly called **Structuralism**, inverts the formula and emphasizes the social structures studies and their impact of these structures on the individuals. It is a macro-to-micro approach since it considers the social changes are triggered from the social (macro) to the individuals (micro). Social Science researchers such as [9] embraced the structural theory approach. After analyzing the current literature, works that had computationally implemented the macro-to-micro approach for panic crowd situations were not found.

The Table 4 shows a comparison based on the following elements: reactive approach, cognitive approach, micro-level explicitation, macro-level explicitation, communication forms and panic behavior explicitation. The reactive and cognitive approaches are related to how individuals are modeled in the system. In a reactive standpoint the agents do not have well-defined symbolic constructions of their internal processes, neither of the environments where they are inserted. The behavior is of a stimulus-response type. On the other hand, a cognitive standpoint allows the agent to make an environment and collectivity members' explicit representation, bringing memory and reasoning about the past chosen actions and planning the future actions. The micro-level and macro-level explicitation parameters refer to the presence of components that represent the collectivity's micro and macro levels in an explicit way. The micro level deals with the agents and their interactions. The macro level displays the main forces that keep stability and force changes to society, making the social structures and the collective behaviors explicit [4]. The communication form parameter deals with the interactions among the collectivity agents. These interactions can happen (i) in an indirect manner, when the information is exchanged through the environment, (ii) in a direct manner, when there is direct information exchange among the agents, (iii) with the perturbation and dissipation processes, as proposed by Luhmann [10]. Finally, the panic behavior explicitation parameter deals with the usage of the collective behavior formation stages explained in Sec. 2. If such stages are modeled and the transition is described in detail for each agent, so they could behave more realistically, and the simulation could be closer to the real phenomenon. Or if a physical or mathematical model is adapted to fit collective behavior empirical data.

In [6] the main concept is moving groups of people from a place in a panic situation to a safer place. Emotions are used to guide the agents and they are used as clauses for the agent's state change. The crowd movement is ruled by

Table 4. Panic Behavior Modeling Comparison Matrix

	First Category		**Second Category**	
Features	[6]	[7]	[8]	Proposed Model
Reactive or Cognitive Approach	REA[1]	REA	COG	COG
Micro-level explicitation	Movement Formulas	Social Force Model	Individual Behavior Model and Global Rules	Collective modeled agents
Macro-level explicitation	Not available	Not available	Not available	Imaginary and Group Mind formation
Commun. forms	Indirect message exchange	No message exchange	Not available	Perturbation and dissipation
Panic Behavior Modeling explicitation	Not available	Not available	Not available	Framework based on Symbolic Interactionism

formulas [1]. The Micro-Level explicitation lies on crowds' formula-applied movement and pushing. There is no macro-level explicitation. On the communication side, the groups of people put pheromones on the environment to qualify it as safe or unsafe, promoting an indirect communication. There is no panic behavior explicitation, although the usage of emotions to aid the agent choices.

The authors in [7] present a crowd behavior model based on social forces and a mathematical model which uses physical forces and certain social behaviors as attractive or repulsive forces. There is no message exchange among the agents (since their interactions are based on differential equations and a simple force system application) either a macro-level explicitation. The micro-level explicitation described in this paper is entirely based on formulas and physical forces. After applying such formulas the agent can move or stay still.

The authors in [8] describe an individual behavior model for agents. Although there is no explicit way of interaction or communication, each agent is aware of himself and the other agents. Besides, there is an internal database inside each agent and a global database that stores state information of all agents. However there is not an explicit panic behavior framework. The agent applies rules based on three parameters (Crowd Density, Tension Level and Sensory Input) and his own database, then he attempts to escape and decides how he is going to escape. Finally, there is no macro-level explicitation.

The model proposed by this paper employs an indirect communication, by perturbation and dissipation of information. At the micro-level, agents make their choices based on information they could physically prove (knowledge) and information obtained from other agents and without physical evidence (belief). The macro-level explicitation is based on the imaginary and Group Mind formation. Finally, all steps of collective behavior formation are described and modeled for the agent.

[1] Movement ruled by formulas; behavior ruled by emotions.

5 Conclusion

This paper proposed a conceptual model for the panic in crowds' phenomenon. It also presented the steps related to the structure and operation of collective panic in a systemic and computable way, from its genesis up to its execution apex. This model integrates theories from social scientists such as [11, 2, 10], and that integration of distinct collective panic theories (in a coherent and computable way) is one of the challenges for this research because most of those theories presents a higher abstraction degree and, in some cases, even contradictory points.

Three theoretical approaches were detected and presented in Sec. 2, along with some computational works that adopt each one of these approaches. Specifically, the model presented by this paper is in the second group, following an emergent-norm interactionist approach. For future works, this conceptual model will be implemented using the Swarm framework [12]. A simulation will be performed and the resulting data will be analyzed along with theoretical and empirical datasets.

References

1. Berends, P., Romme, G.: Simulation as a research tool in management studies. European Management Journal **17**(6) (1999) 576–583
2. Turner, R.H., Killian, L.M.: Collective Behavior. Prentice-Hall (1957)
3. Wiener, N.: Cybernetics: Or the Control and Communication in the Animal and the Machine. The MIT Press, Cambridge, MA (1948)
4. da Silva, V., Marietto, M., Ribeiro, C.: A Multi-agent Model for the Micro-to-Macro Linking Derived from a Computational View of the Social Systems Theory by Luhmann. LNCS (2008)
5. Le Bon, G.: The Crowd: A Study of the Popular Mind. The Macmillan Co. (1896)
6. Banarjee, S., Grosan, C., Abraham, A.: Emotional ant based modeling of crowd dynamics. In: Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, IEEE Computer Society Washington, DC, USA (2005) 279
7. Helbing, D., Farkas, I., Molnar, P., Vicsek, T.: Simulation of pedestrian crowds in normal and evacuation situations. Pedestrian and Evacuation Dynamics (2002)
8. Pan, X., Han, C., Dauber, K., Law, K.: A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. AI & Society **22**(2) (2007) 113–132
9. Smelser, N.J.: Theory of Collective Behavior. Free Press (1963)
10. Luhmann, N.: Social Systems. Stanford University Press (1996)
11. Park, R.E.: An Outline of the Principles of Sociology. Barnes and Noble (1939)
12. (http://www.swarm.org), S.: Swarm main page (Dec. 2008)

A Critical Study of the Coherence between EBMs and ABMs in the Simulation of the Hawks-Doves-LawAbiders Society

Fernanda Mendez Jeannes[1], Antônio Carlos da Rocha Costa[1]

[1] Programa de Pós-Graduação em Informática, CEPOLI
Universidade Católica de Pelotas (UCPel)
Rua Félix da Cunha 412, 96010-000 – Pelotas – RS - Brasil
fejeannes@gmail.com, rocha@ucpel.tche.br

Abstract. We are interested in using Equation-based Models (EBMs) as formal specifications of macro-level features of social simulations, so that they could support the subsequent development of more detailed Agent-based Models (ABMs). In this work, we tried to establish the way a particular macro-level feature present only in the ABM of a certain society, namely, the density of the population in its environment, may influence the coherence between the results of the two models about the evolution of the population profile. We took as case study the HDL (Hawks, Doves, and LawAbiders) society. In our simulations, higher values of population density tend to produce less coherence between the two models, because of the interference between the population density parameter and the particular way the agents were programmed to wander in the environment. The work indicates the possibility of a systematic study of the interdependencies of ABMs and EBMs when they are used complementarily.

Keywords: Social simulation, Equation Based Models, Agent Based Models

1 Introduction

Social simulations may be addressed on at least two levels: the macro social level, which focuses on the global behavior of the society, and the micro social level, which *additionally* studies the local behavior of the individuals of the population.

The macro social level views the society as a whole, focuses on populations and sub-populations, with the predominance of the global view of the system, usually assuming that individuals are similar, with the same scale of values and even same behavior.

The micro social level focuses on individuals, their interactions, actions and behaviors, how they are born and die, and how macro level features are generated from (*emerge* from) the mass of individual interactions.

In this work, we make use of Equation Based Models (EBMs) to model directly the macro-level, while we use Agent Based Models (ABMs) to model *directly* the micro-level, and *indirectly* the macro-level via emergent features.

The problem then arises of the compatibility between the macro-level results produced by the two kinds of models. Equation Based Models (EBMs) make use of differential or difference equations [1, where the state of the system is represented by a set of numerical variables, and the future state of the system is determined from its current state through a set of difference or differential operators that model the dynamics of the system.

The Agent Based Models (ABM) work with the basic concept of an autonomous agent, which controls its own actions based on its perceptions of the environment, and that acts on that environment. Thus ABMs are composed of a set of agents that demonstrate individual behaviors, which collectively form the system.

ABMs and EBMs can be seen as rival models for social simulation [2]. Accordingly, the critical aspects that differentiate the two kinds of models are, among others:

- Individuals are characterized either separately (ABMs) or in an aggregate way (EBMs), which implicitly suggests that the best applicability of the ABMs is to the micro-level, while the best applicability of the EBMs is to the macro-level.
- In EBMs applied to the macro-level, the observables of the system concern aggregate information derived from the equation-driven evolution of aggregate information themselves.
- In ABMs applied to the micro-level, individuals interact with each other through their behavior in the environment, so that the observables of the system concern aggregate information derived from the log of individual behaviors.

The choice between one model or another, in the development of a social simulation, should be made, according to [2], in a case by case basis, centering around practical considerations (need to have control of behaviors of individuals in the simulation, need to take into account aspects of the interaction between individuals and the environment, etc.).

In this paper, we take the point of view that EBMs and ABMs can be treated as complementary, rather than rival models, so that the idea of choice between them becomes meaningless. We claim that if enough features can be identified that allow for a safe control of the degree of coherence between the macro-level results produced by the two kinds of models, than EBMs of the macro-level can be used as preliminary formal specifications of the ABMs that should be subsequently developed.

In such position as a formal specification of an ABM, an EBM can then be taken as a yardstick for the verification of the correctness of the definition and implementation of the ABM, possibly helping in the establishment of the faithfulness of the ABM (see, e.g., [3] for the problem of the faithfulness of social simulation models).

Such purpose requires, however, an assessment of the degree of coherence between the results that each kind of model may provide.

This paper focus on a particular issue, regarding that degree of coherence, namely, the impact that the density of the population of the ABM (that is, the number of agents per unit area in the environment) may have on the coherence of the macro-level results of the two models, in the cases where the casual spatial encounters between individuals are important for the functioning of the social system.

To have a concrete case study, we analyze the influence of the population density on the coherence between macro-level results provided by EBMs and ABMs of the HDL (Hawks-Doves-LawAbiders) kind of social system [1].

This article is presented as follows. Section 2 presents the problem studied for both models, the HDL society. In sections 3 and 4, are presented the EBM and ABM models of the HDL society, respectively. In Section 5 the results obtained with the simulations are compared, focusing on the issue of the evolution of the population profile. Section 6 presents the conclusions regarding this work.

2 The HDL Society

Martinez Coll describes the evolution of the society modeled here as the necessary evolution of the "Hobbesian state of nature", given a Bioeconomic perspective on it [4]. The "Hobbesian state of nature" is one in which members of the society are always competing for resources, so that, as the resources always belong to someone, conflicts between individuals arise all the time.

The members of this society are allowed to adopt one of three fighting strategies, respectively called the Hawks, Doves and LawAbiders strategies.

- The Doves never try to get resources from others, but expect them to leave voluntarily; a dove abandons its resource as soon as it is attacked; if two doves compete for the same resource, one of them wins by chance or persistence;
- The Hawks always attack and try to get resources from others; a hawk only gives up its action if it is badly hurt;
- LawAbiders never attack to get resources from others, but always defend themselves when attacked; a LawAbider may succeed or not in its defense, and thus may lose its resources or get new ones in each conflict.

During the evolution of the society, mechanisms such as inheritance, imitation and indoctrination may be used to spread the best strategy, so that the different strategies are adopted by the members of the society according to the degrees of success that the strategies are giving to their adopters in the conflicts.

Each strategy involves costs and gains. Any conflict between individuals with their respective strategies will result in a new balance of resources, following the rule of profit and loss governing that meeting. For example, when two hawks meet, they always fight and the cost of their fight is high. In general, when two individuals with the same strategy meet, both have the same chance of winning or losing resources.

3 Equation Based Model

The evolution of each strategy at time t, for the strategies $i = H, D, L$, of the respective sub-populations of Hawks, Doves and LawAbiders, is given [1] by the equation

$$P_i(t+1) = P_i(1+F_i(t)) \tag{1}$$

where the size of the sub-population is represented by the percentage variables P_H, P_L and P_D, and the measure of fitness at time t by the variables $F_H(t)$, $F_D(t)$ and $F_L(t)$. The fitness of each strategy corresponds to how well that sub-population is doing in the environment, i.e., the degree of success of the strategy in relation to others.

The yield $Y_i\ (t)$ that a population *i* presents in a time *t* is given by the equation

$$Y_i(t) = \sum_j R_{ij} P_j(t) \tag{2}$$

where the gain R_{ij} is given by the rules of fight between an individual follower of the strategy *i* and an individual that follower the strategy *j*, as defined in Table 1.

The yield of the entire population of the system $Y_S(t)$ is obtained from the current profile of sub-populations

$$Y_S(t) = \sum_i Y_i(t) P_i(t) \tag{3}$$

Therefore, the fitness of the sub-population *i* at time *t* is given simply by the difference between its yield and the total yield of the population

$$F_i(t) = Y_i(t) - Y_S(t) \tag{4}$$

The higher the yield of a sub-population in relation of the others, the greater the probability of its strategy being imitated by members of the other sub-populations, or inherited by its descendants.

4 Agent Based Model

We developed an ABM[1] where individuals wander in the environment and the conflicts occur as they meet each other.

In each conflict, the result of the conflict is decided in accordance with the strategies adopted by the fighters. Following [1], the costs and rewards involved in the fights were defined as in Table 1:

[1] NetLogo was used for the development of the simulations studied in this work.

Table 1. The pay-off matrix of the conflicts in the HDL society

	Dove	Hawk	LawAbider
Dove	2,2	0,10	1,6
Hawk	10,0	-5,-5	2.5,-2.5
LawAbider	6,1	-2.5,2.5	5,5

Table 1 is obtained directly as a suitable intuitive numerical representation of the informal definition of the conflict rules ([4], cf. Sec. 2) and establishes a *bridge* between the macro-level EBM and the micro-level ABM, as it is *embedded both* in the procedure that calculates the results of the conflicts in the ABM, and in the R_{ij} parameters of the EBM equations.

Our ABM of the HDL society was implemented based on the information in Table 1. However, many additional non-EBM related decisions have had to be made, because at the micro-level several *particularities* of individuals (which are not explicitly indicated in the EBM) have be considered, specially, when such features interfere with technical issues relating to the *programming* of the agent behaviors.

This is an important issue because most of those micro-level particularities of the individuals often impact in strong ways the overall behavior of the multiagent society and, thus, the coherence between the macro-level results obtained through the ABM and those obtained through the EBM. In the following, we consider some of them.

At each step the individual chooses a random direction to move. But before that, it must observe if there are other individuals at the final destination. If there is no other agent in that point, the individual may move. If there is only one other agent, the individual moves to that point and a conflict happens. If the point that the individual has chosen to move to is already occupied by two agents, then we have decided to make it not move to that place, remaining in their place of origin, so that only conflicts between two individuals may occur at each position. The fact that an individual does not move to a place where there are already two other individuals is directly related to cost accounting, which is calculated for only two individuals. Clearly, this is an issue that is not taken into account in the EBM, since the environment is also not an issue for it (cf. [2]).

Also, the ABM has to define the spatial features of the individual moves in the environment. Those spatial features impact not only the probability of the occurrence of interactions between individuals, but also the scope of the *local observations* about the success of each strategy that each individual may make.

For instance, if the walking proceeds with steps that are too small, the size of the visited region in a given time may turn out to be too small compared with the overall size of the environment, possibly leading to problems like: a too limited sampling of the society, or an excessive repetitive number of conflicts with the same sub-group of individuals occupying a restricted part of the environment, etc., all leading to an insufficient, even plainly incorrect, assessment of the current yields that the various strategies are producing in the society of as a whole. This is clearly another problem in the development of the ABM that the EBM can not help to solve

In the ABM, another important aspect had to be defined independently of the EBM, which is the procedure by which each individual takes the decision to change or not to change the strategy it is currently adopting, in consequence of its perception of the current yields in the society. The EBM gives no hint on such strategy, it only specifies the resulting percentage of strategy changes that occur in the society at each moment, but not the procedures the individuals may adopt to make their private decisions.

In the ABM model, such decision procedures have to be made explicit. In the ABM implementation that we developed, each individual keeps a memory of the number of conflicts it has participated in, and of the balance of costs and rewards accumulated in those conflicts, and use either one of those information to determine when it is time to decide about changing strategy or not, and what such decision should be.

To choose the best strategy to change to, the individual has to accumulate various kinds of information. As examples, we have the number of individuals of each sub-population found in the past period under evaluation, in order to constitute a sample of the current distribution of individuals among the sub-populations of the society; also, the average balance of costs and rewards obtained by each such sample of sub-populations, so that an estimate of the yield of each sub-population can be calculated.

This calculation of such estimate of the yield of each sample sub-population was programmed as follows: at each conflict, the individual updates both the balance of costs and rewards and the number of individuals in the sample of the sub-population of the individual it has met. When it comes the time to consider a possible change of strategy, the individual proceeds as follows: it compares the yield of the sample of its own population to the total yield of the total sampled population. If the yield of the individual's own sampled sub-population is in disadvantage with respect to the yield of the total sampled population, the individual changes its strategy to that strategy which it has found with highest yield, in the sampled population; otherwise it keeps its strategy.

Alternative decision procedures had to be tested, in order to determine the best one, which is the one that best matches the rate of changes in the population profile, defined by the EBM in the parameter $1+F_i(t)$ that controls the variation of the size of the population P_i, in the equation (1). A criterion requiring an unconditional change to the best strategy at each decision moment gives a rate of change in the population profile that is usually too high, compared with that determined by the EBM. A criterion requiring a cautious decision, based for instance on a large difference between the yield of the individual's own sampled population and the yield of the overall sampled population, may lead to a too slow rate of change in the population profile, compared to those required by the EBM. Clearly, the EBM gives no direct hint on which decision procedure should be adopted by the individual agents of the ABM.

5 Results of Simulations

The initial set up for every simulation was defined with the following set of values: Hawks= 90%, Doves = 5%, LawAbiders = 5% of the overall population.

From such initial set up, a general picture of the behavior of the population can be given [1]. Initially, it is expected that the Hawks mainly conflict with each other, leading to high-costs to the decision to keep adopting the Hawks strategy. So, the Hawks gradually decide to adopt either the LawAbiders strategy or the Doves strategy. As the number of the doves stay small at the beginning, even with some Hawks becoming Doves, their sub-population tend to grow, since their is the best yields in such situation. But as the Doves population grows, their yield reduces, and Doves tend to become LawAbiders. As the number of LawAbiders increase, their yield increasingly becomes even better, producing a permanent tendency towards the spread of such strategy, the LawAbiders finally winning the competition.

We note, however, that the simulations were performed for three different simulation models: the EBM, the pure ABM (with agents strictly limited to a local view of the environment, as usually assumed in ABMs), and a so-called *multilevel* model [1] where agents were given access to *external oracles* that numerically account for certain features of the global environment. The following sub-sections compare the results obtained.

5.1. The EBM Simulations

The results of the EBM that we implemented faithfully meet the expectations, as shown in Fig. 1.

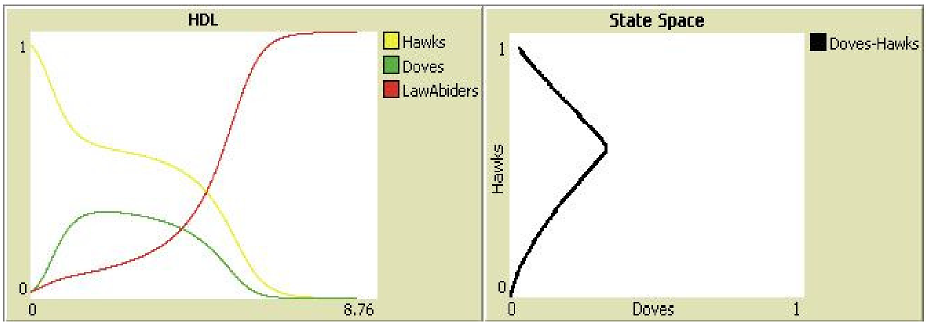


Fig. 1. Result of the EBM simulation.

5.2. The ABM Simulations

For the ABM simulations, the following agent parameters were determined: size of the steps at each simulation time equal to 5 environment cells; a number of 3 conflicts performed and assessed in order to decide about changing or not the strategy; and an environment of 50 to 50 cells, giving an environment area of 2500 cells.

In the case of the ABM simulations, we were specifically interested in the impacts that the density of the population of individuals in the environment would have on the faithfulness of the ABM simulations with respect to the EBM simulations.

The number of steps simulated is determined by the time that the variation of the population reaches some stability. These values were determined after a considerable number of simulations for each of the densities discussed in this work.

To assess such impact, we considered four different population densities, namely, population densities of 0.04, 0.2, 0.4 and 0.6 individuals / environment cell.

The simulations with the population density of 0.04 were considered qualitatively satisfactory after running about 500000 steps, resulting in population sizes of 88% of Law-Abiders, 7% of Hawks and 6% of Doves, compared to the combination of 100%, 0% and 0% required by the EBM. The average graph of the performed simulations can be seen in Fig. 2.

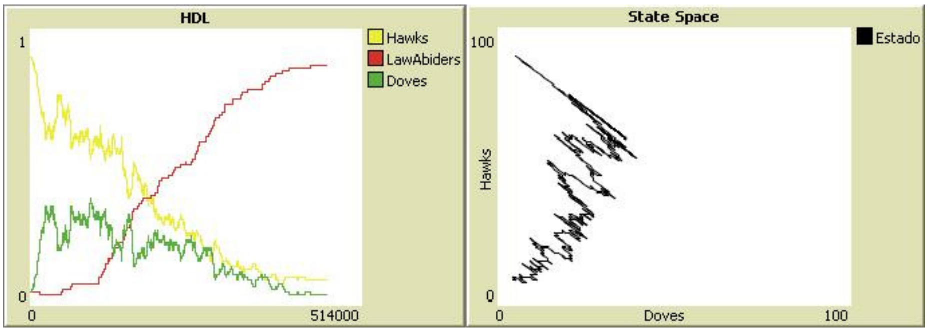


Fig. 2. ABM simulation with population-density = 0.04.

The simulations with population density of 0.4 were less satisfactory. The population sizes obtained were of 74% of LawAbiders, 18% of Hawks and 8% of Doves on the average, as shown in Fig. 3.

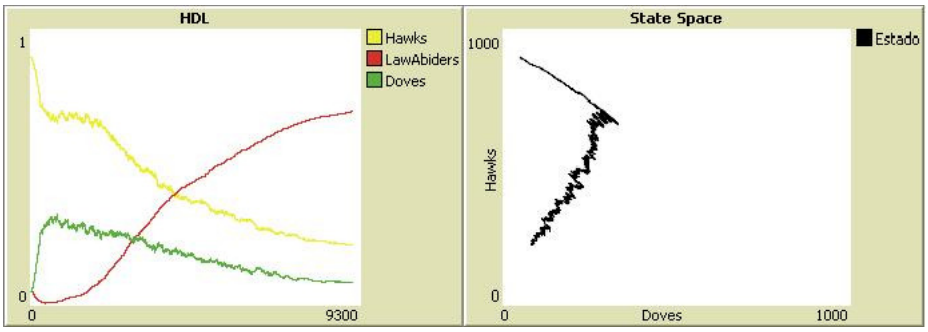


Fig. 3. ABM simulation with population-density = 0.4

The simulations with population density of 0.2 got worse results than simulations with population density of 0.04 and better than simulations with population density of 0.4. The profile obtained was 87,35% of LawAbiders, 8,55% of Hawks and 4,1% of Doves, as shown in Fig. 4.

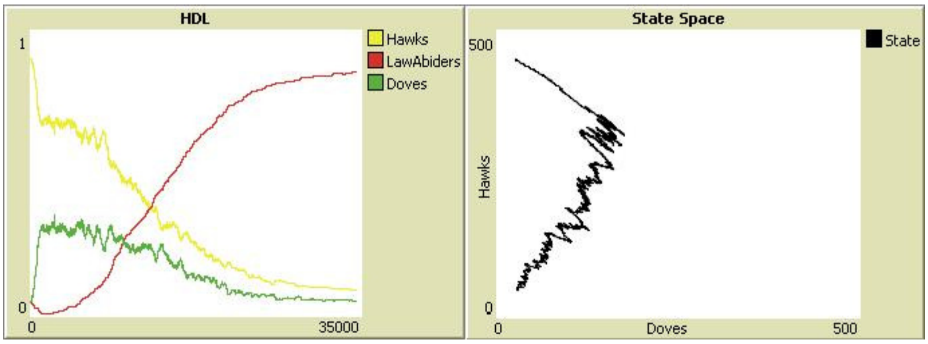


Fig. 4. ABM simulation with population-density = 0.2

The simulations with population density of 0.6 were the less satisfactory. The population sizes obtained were of 61,5% of LawAbiders, 26,5% of Hawks and 12% of Doves on the average, as shown in Fig. 5.

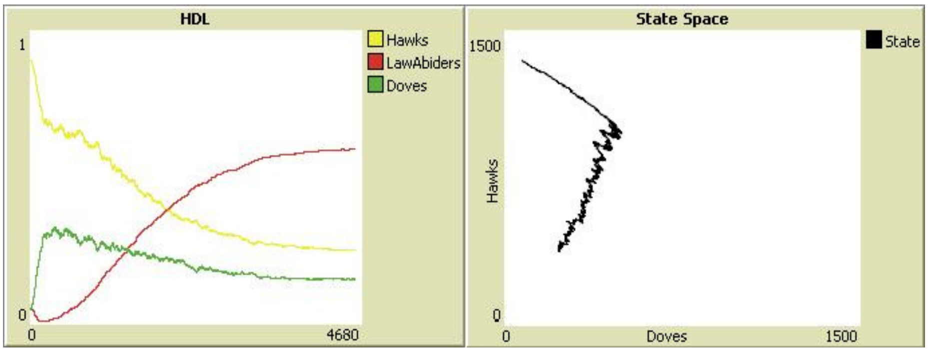


Fig. 5. ABM simulation with population-density = 0.6

We have found, then, that the ABM with the restriction that agents do not occupy cells with more then two individuals leads to the effect that the *higher the population density*, the *worse the faithfulness* of the ABM to the EBM.

This is seems to be so because with high population densities, the individuals have difficulty in finding places where to move to, thus tending to reduce the scope of their wandering and, so, tending to restrict their conflicts to their immediate neighbors, with the consequence that the quality of their assessments of the yields of the different sampled populations is reduced.

Similarly, in a population less dense, with fewer individuals per cell, the agents may take more time to find another agent to fight with, so retarding the decision on whether to change or not the strategy. As these individual decisions affect the total variation of the population at each moment, the final result takes longer to appear, thus requiring a larger number of steps to stabilize the number of individuals in each population. Also, retarding the decisions may lead them to be taken when they may be not suitable anymore, as the environment may have changed significantly during the sampling period.

Again, we think that this analysis shows clearly a strong influence on the result of the ABM simulation of an *unexpected interference* (from the sole EBM point of view) between a *neatly macro-level ABM parameter* (the population density) and a *neatly micro-level ABM parameter* (the decision criterion about where to move to at each instant).

The unexpectedness of the interference is even more strong as the programming of the criterion about how to wander in the environment would usually be treated as a "mere" programming implementation detail, assumed to have no impact on the social behaviors of the agents and on the overall functioning of the society.

5.3. The Simulations of the Multilevel Model

In a multilevel model [1], the simulations of individual behaviors are allowed to make use of information that has a global nature with respect to the society. The main advantage of multilevel models is the decrease of the need to assign to the very behaviors of the individual agents (forced to behave under restricted local views of the society) the responsibility of assessing the whole simulated society. Thus, the programming of the agents may concentrate on the particular behavioral issues that the simulation is designed to study.

Multilevel models could be imagined that make use of agent technologies in different ways, from the simple simulation of agent actions on the basis of the evolution of individualized numerical variables [1], to the full-fledge use of agent oriented programming techniques, where external oracles responsible for the global observation of the society communicate their findings to the agents through either messages or globally shared variables, an alternative that we have called *agent-based multilevel simulation* in [5].

The multilevel model we used in this study is of that agent-based kind. The individuals change strategy according to probabilities that depend of the values of the yields, which are calculated by external oracles to match the $1+F_i(t)$ factors in the EBM model.

The result of a simulation for a population density of 0.04 is shown in Fig. 6, where the steady point of the simulation is seen to have been achieved around 15000 simulation steps.

Clearly, the agent-based multilevel simulation performed better than any of the ABMs, due to the availability of better assessments of the yields of the sub-populations of the society.

For the sake of space, we let for a further paper the analysis of the interference of the variation of the population density on the results of this agent-base multilevel model.

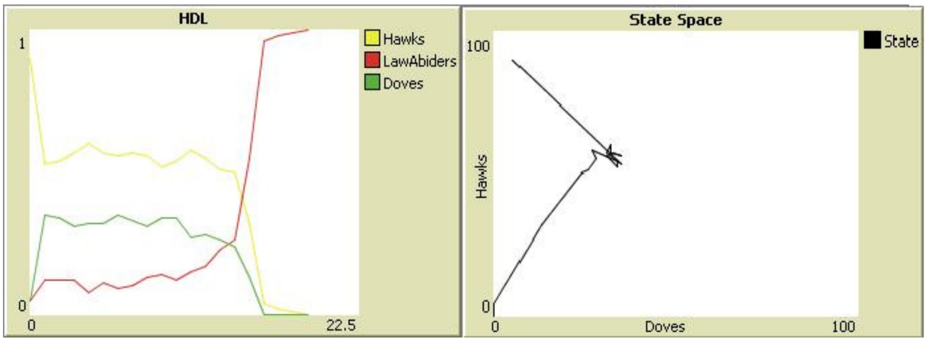


Fig. 6. Agent-base Multilevel simulation with population density = 0.04.

6 Conclusion

We initially targeted this study to the elucidation of the interference that the population-density could have on the faithfulness of the ABM with respect to the EBM.

We have found that the population-density, which is a macro-level parameter, has a strong connection to the micro-level rule of spatial displacement of the individuals in the ABM, to the point that the population-density parameter is able to jeopardize the quality of the results of the ABM simulations, if the displacement rule presents certain restrictions.

The restriction that agents do not occupy cells with more then two individuals, in the ABM simulations, leads to the effect that the higher the population density, the worse the faithfulness of the ABM to the EBM. This seems to be so because, in high population densities, the individuals have difficulty in finding places where to move to, and tend to reduce the scope of their wandering and to restrict their conflicts to their immediate neighbors, thus reducing the quality of the assessments they make of the yields of the different populations of the society.

On the other hand, in a population with low density, the agents may take long times to find other agent to fight with, so retarding the decision about whether to change or not their strategy. As these individual decisions affect the total variation of the population at each moment, the global result of such changes takes longer to appear, thus requiring a larger simulation time to stabilize the number of individuals in each population.

Also, the development of the agent-based multilevel simulation model with external oracles able to globally assess the yields of the sub-populations allowed the simulations to provide more faithful results than the ABM that operates without such oracles.

The two final conclusions that we reach are: first, that this work showed the possibility of the development of a systematic study of the interdependencies of ABMs and EBMs when they are used complementarily; second, that the idea of taking EBMs as formal specification of ABMs seems fruitful, as EBMs may then function as yardsticks for the assessment of the quality of the ABMs results; and third, that a simulation model, like the agent-based multilevel model, that adequately combines resources taken from both EBMs and ABMs may provide better simulation results than simulations based on just ABMs.

Acknowledgements: This work was partially supported by CAPES and CNPq.

References

1. Gilbert, N. and Troitzsch, K. G. Simulation for the Social Scientist. Open University Press, 2005.
2. Parunak, H. v. D. and Savit. R. and Riolo, R. L. Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. In: Multi-Agent Systems and Agent-Based Simulation First International Workshop, MABS'98, Paris, France, July 4-6, 1998. Proceedings. Springer, Berlin, 1998. p. 10–25.
3. Moss, S. Alternative Approaches to the Empirical Validation of Agent-Based Models. Journal of Artificial Societies and Social Simulation vol. 11, no. 1, January 2008. (available online at http://jasss.soc.surrey.ac.uk/11/1/5.html).
4. Coll, J. C. M. A Bioeconomic Model of Hobbes' "State of Nature". Social Science Information, 25: 493–505, 1986.
5. Costa, A. C. R.; Jeannes F. M.; Cava U. A. Equation-based Models as Formal Specifications of Agent-based Models for Social Simulation: preliminary issues. Proceedings of 1st Brazilian Workshop on Social Simulation, Salvador, Brazil, October 26-30, 2008.

Chapter 11

TEMA - Text Mining and Applications

Combining Unigrams and Bigrams in Semi-Supervised Text Classification

Igor Assis Braga, Maria Carolina Monard, Edson Takashi Matsubara

Mathematics and Computer Science Institute
University of Sao Paulo
Sao Carlos, SP 13566-590
Brazil
{igorab,mcmonard,edsontm}@icmc.usp.br

Abstract. Unlabeled documents vastly outnumber labeled documents in text classification. For this reason, semi-supervised learning is well suited to the task. Representing text as a combination of unigrams and bigrams has not shown consistent improvements compared to using unigrams in supervised text classification. Therefore, a natural question is whether this finding extends to semi-supervised learning, which provides a different way of combining multiple representations of data. In this paper, we investigate this question experimentally running two semi-supervised algorithms, Co-Training and Self-Training, on several text datasets. Our results do not indicate improvements by combining unigrams and bigrams in semi-supervised text classification. In addition, they suggest that this fact may stem from a strong "correlation" between unigrams and bigrams.

1 Introduction

Text Classification (TC) has received a great deal of attention from the machine learning community in the last ten years. The focus has primarily been on supervised classification, although there has also been interest in Semi-Supervised Learning (SSL). In SSL, one can derive a classifier from labeled and unlabeled data. This is particularly suited for many text classification tasks, since it is often the case that many unlabeled documents are available although labeling them is expensive.

A recurrent issue in TC is text representation. Extracting words, or *unigrams*, is by far the most common way to represent raw text in TC [1]. While this feature extraction technique is applicable to almost any kind of text, it has been found to be poor for certain classification tasks [2]. For dealing with this deficiency, word n-grams, *i.e.*, sequences of n consecutive words, were proposed by various researchers as a way to expand the standard unigram representation model [3].

Combinations of unigrams and n-grams, particularly *bigrams* ($n = 2$), have not shown consistent improvement in supervised text classification [1, 3]. Thus, a natural question arises: *does this behavior extend to semi-supervised learning*? In this paper, we investigate this question, which, to the best of our knowledge is yet unanswered. We conducted experiments on traditional text datasets using both the Co-Training and the Self-Training semi-supervised algorithms. The results do not indicate a major performance improvement by combining unigrams and bigrams in SSL. Based on these results and on other measurements taken during the execution of the Co-Training algorithm, we have found a possible explanation of why combining unigrams and bigrams does not help boost classification performance in SSL and even in the supervised case.

The remainder of this paper is organized as follows. In Section 2, we describe the Co-Training and the Self-Training algorithms. In Section 3, we review some related work and explain two different ways of combining unigrams and bigrams in semi-supervised learning. In Section 4, we present the experimental design and results. Finally, we discuss our findings and future work in Section 5.

2 Semi-Supervised Learning

Semi-Supervised Learning (SSL) has been applied to classification tasks in cases where labeled examples are scarce in comparison to available unlabeled examples. More precisely, we want to induce a classifier $h \rightarrow X \times Y$ using a set of j labeled examples $\{(\mathbf{x}_i, y_i)_{i=1}^{j}\}$ and a set of k unlabeled examples $\{(\mathbf{x}_i)_{i=j+1}^{j+k}\}$, with $k >> j$. Algorithms in SSL can be divided into *single-view* and *multi-view* algorithms. The latter requires that examples be represented by at least two different feature sets that are each nearly sufficient for classification on their own.

In the next section, we show two different ways of combining unigrams and bigrams in SSL. One of them uses a single-view algorithm, whereas the other requires a multi-view algorithm. In the remainder of this section, we present a single-view algorithm, Self-Training, and a multi-view one, Co-Training. Both of them were used in the experiments we describe later.

2.1 Self-Training

Self-Training is a wrapper semi-supervised algorithm [4] that makes use of a supervised learning algorithm that "trains" itself. Starting from a (possibly) small labeled set, the supervised learning algorithm derives a classifier, which is used to classify unlabeled examples that are later inserted into the labeled set. This expanded labeled set forms the basis of another induction step, and the process is repeated.

Algorithm 1 describes Self-Training. It takes as input a set L of labeled examples and a set U of unlabeled examples. Initially, a classifier h is derived using L and a supervised learning algorithm, which needs to produce classifiers that output a confidence score for its predictions. After h is derived, it is used to classify all examples in a randomly chosen $U' \in U$. The examples classified with the greatest confidence scores are selected by the function *bestExamples* and later inserted in L. If binary classification is all that matters, *bestExamples*

Algorithm 1: SELF-TRAINING

Input:
- a set L of labeled examples
- a set U of unlabeled examples

repeat
  use the examples in L to derive a classifier h;
  randomly choose some examples from U, and insert them in U';
  L' = all examples in U' labeled by h;
  $L'' = bestExamples(L')$;
  $L = L \cup L''$;
  remove from U' all the examples in L'';
until $L'' = \emptyset$;
Output: h

can select the p most confident positive examples and the n most confident negative examples from L'. The algorithm stops when there are no more unlabeled examples available or other stopping criteria are met.

2.2 Co-Training

The CO-TRAINING algorithm [5] is the multi-view counterpart of SELF-TRAINING. Interest in multi-view algorithms comes from the fact that they take advantage of multiple representations (views) of the data to produce better classifiers. CO-TRAINING acts as if two SELF-TRAINING routines were executed simultaneously over the same sets L and U, the differences being that 1) multiple classifiers are derived, each one in a different view of the examples in L; and 2) the examples labeled by each classifier are also available to derive the other classifiers in the next iteration.

Hereafter, we assume that two views of the examples are available, so that an example $\mathbf{x}$ is a pair $(\mathbf{x}_1, \mathbf{x}_2)$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ are the values of the features that describes the example $\mathbf{x}$ in views 1 and 2 respectively. The general conditions that enable CO-TRAINING to be successful are that $\mathbf{x}_1$ and $\mathbf{x}_2$ should not be too correlated and that the classifiers we want to learn in each view give the same label to $\mathbf{x}_1$ and $\mathbf{x}_2$ [6].

Algorithm 2 describes CO-TRAINING. It is similar to SELF-TRAINING, although now the function *bestExamples* has to select the most confident examples classified by h_1 and h_2. In the original CO-TRAINING algorithm [5], *bestExamples* selects the p most confident positive examples and the n most confident negative examples in each L'_1 and L'_2. With this version of *bestExamples* in mind, at most $2p + 2n$ examples are labeled at each iteration, since the same example can be selected from both L'_1 and L'_2.

Algorithm 2: Co-Training

Input:
- a set L of labeled examples
- a set U of unlabeled examples

repeat
  use view $\mathbf{x}_1$ of the examples in L to derive a classifier h_1;
  use view $\mathbf{x}_2$ of the examples in L to derive a classifier h_2;
  randomly choose some examples from U, and insert them in U';
  L'_1 = all examples in U' labeled by h_1;
  L'_2 = all examples in U' labeled by h_2;
  $L'' = bestExamples(L'_1, L'_2)$;
  $L = L \cup L''$;
  remove from U' all the examples in L'';
until $L'' = \emptyset$;
Output: h_1, h_2

3 Combining Unigrams and Bigrams in Text Classification

The study of unigram and bigram combinations started in supervised text classification. Next, we briefly discuss some previous work along this line. Following that, we explain two different ways of combining unigrams and bigrams in semi-supervised learning.

Supervised TC There are positive and negative results reported in the text classification literature on the subject of combining unigrams and bigrams. We review recent work, as good reviews can be found elsewhere [3, 7].

Bekkerman et al. [3] present a critical summary of both positive and negative results. They state that the majority of positive results are not significantly better than the baseline results of the datasets used, and, when they are, the baseline results tend to be very low. They also propose to combine unigrams and bigrams in supervised text classification using a feature generation method based on unigram and bigram clustering. However, the improvements achieved on the well known 20 Newsgroup dataset are not statistically significant.

Caropreso et al. [8] evaluate the usefulness of unigrams and bigrams in a learner-independent manner. They use some feature evaluation metrics to analyze the discrimination power of bigrams compared to that of unigrams. In the Reuters-21578 dataset, they find that there are bigrams that are more discriminative than unigrams. However, when more and more bigrams are selected at the expense of letting unigrams out of the feature set, the classification performance in the same dataset tends to degrade.

Pang et al. [2] analyze the use of machine learning techniques for sentiment classification in textual data. They represent text documents using, among others, unigrams, bigrams, and a single feature set that joins together unigrams and bigrams. This expanded feature set is then compared to what is achievable by using unigrams or bigrams alone on the Movie Review dataset. Their results show that the feature set composed of unigrams and bigrams does not reach a classification performance better than unigrams.

Semi-supervised TC As far as we know, there was no attempt to study unigram and bigram combinations in semi-supervised text classification. Previous work in semi-supervised TC has only considered the use of unigrams for text representation [9–11]. Therefore, it would be interesting to verify if this combination can improve classification performance in semi-supervised TC.

As a first approach, we can combine unigrams and bigrams as it has been done in supervised learning: joining the two representations together and using a single-view semi-supervised algorithm to learn a classifier from it. However, it should be observed that the way unigrams and bigrams will interact with each other is fully determined by the learning algorithm used.

As we described in the last section, a multi-view semi-supervised algorithm can be used when examples can be represented by at least two different feature sets that are each nearly sufficient for classification on their own. As our experiments in the next section suggest, this happens to be the case of unigrams and bigrams. An interesting characteristic of this combination is that each representation is kept in its own "world", since two classifiers are independently induced in each representation.

4 Experimental Evaluation

We conducted experiments to evaluate unigram and bigram combinations in semi-supervised learning. Particularly, we tried a single-view combination using Self-Training and a multi-view combination using Co-Training. We also ran Self-Training using unigrams and bigrams alone for comparison.

4.1 Datasets

Five text datasets were used in the experiments. One of them consists of the web pages view from the Courses dataset [5]; three are subsets of the UseNet news articles 20 Newsgroups dataset[1]; and the last one is the `polarity_dataset v2.0` from the Movie Review Data[2] — Table 1. All datasets except Courses are balanced.

The datasets were decomposed into the attribute-value format, where unigrams (1-gram) were used as one view and bigrams (2-gram) as the second view of the datasets. To this end, we used PreTexT II[3], a locally-developed text preprocessing tool. Stop-word removal and stemming [12] were carried out for each

[1] `http://people.csail.mit.edu/jrennie/20Newsgroups/`
[2] `http://www.cs.cornell.edu/People/pabo/movie-review-data/`
[3] `http://www.icmc.usp.br/~caneca/pretext.htm`

Table 1. Datasets description and attributes

Dataset	#Doc	View	#Attr	#Attr. P	Class	%Class
Courses	1050	1-gram	12254	3313	course	22%
					non-course	78%
		2-gram	46421	2073	course	22%
					non-course	78%
Hardware	1943	1-gram	13398	3958	pc	50%
					mac	50%
		2-gram	47331	2846	pc	50%
					mac	50%
Vehicles	1984	1-gram	14048	5362	car	50%
					motorcycles	50%
		2-gram	51404	3605	car	50%
					motorcycles	50%
Sports	1993	1-gram	14254	5741	baseball	50%
					hockey	50%
		2-gram	60114	4548	baseball	50%
					hockey	50%
Movie	2000	1-gram	25302	10669	pos. review	50%
					neg. review	50%
		2-gram	299423	9186	pos. review	50%
					neg. review	50%

dataset. After that, unigrams and bigrams that appeared in 3 or less documents were removed. The result of this pre-processing step is tabulated in Table 1, which shows the dataset name (Dataset); number of documents in the dataset (#Doc); number of generated attributes (#Attr); number of attributes left after pre-processing (#Attr. P); and class distribution (%Class).

4.2 Methodology

The supervised learning algorithm we used in Self-Training and Co-Training was the *Multinomial Naive Bayes* (MNB) [13]. In order to obtain a lower bound of the error that the algorithms can reach in these datasets, we measured the error rate of MNB using the full datasets and 10-fold cross-validation. Results (AUC and mean error rate) are shown in Table 2. It can be observed that, although there are no significant differences related to the AUC values for all the datasets, the error rate (%Error) is a little higher using bigrams (2-gram). It is also possible to observe that except for the Movie dataset and for the Hardware dataset using bigrams, the error rate of the classifiers is low. Moreover, using unigrams and bigrams together as a single feature set (1+2-gram) does not always improve classification performance compared to unigrams (1-gram), and, when it does, it is by a small margin. These results are in accordance with our analysis of previous work.

As the datasets we use contain only labeled examples, we ran Self-Training and Co-Training in a simulation mode, in which the true labels of an expressive

Table 2. *Multinomial Naive Bayes* results

Dataset	View	*Multinomial Naive Bayes* AUC	%Error
Courses	1-gram	0.97 (0.01)	4.57 (1.61)
	2-gram	0.98 (0.01)	5.24 (1.03)
	1+2-gram	0.97 (0.01)	4.86 (1.45)
Hardware	1-gram	0.98 (0.01)	6.23 (1.37)
	2-gram	0.96 (0.01)	12.15 (2.23)
	1+2-gram	0.98 (0.01)	6.38 (1.69)
Vehicles	1-gram	0.99 (0.01)	2.27 (1.33)
	2-gram	0.99 (0.01)	4.84 (2.13)
	1+2-gram	0.99 (0.01)	1.82 (1.19)
Sports	1-gram	0.99 (0.03)	1.15 (0.79)
	2-gram	0.99 (0.01)	3.01 (0.67)
	1+2-gram	0.99 (0.01)	0.95 (0.65)
Movie	1-gram	0.87 (0.03)	19.00 (2.99)
	2-gram	0.84 (0.04)	23.55 (3.32)
	1+2-gram	0.87 (0.03)	18.60 (3.25)

number of examples are hidden from the algorithms to create the unlabeled set. Furthermore, to assess the behavior of both algorithms using 10 cross-validation, the sampling method was adapted as follows: first, the examples in both views are paired and marked with an ID. Then, the folds are sampled so that both training and test samples are compatible, *i.e.*, an example identified with a given ID appears only in the training or in the test sample in both views.

All experiments were carried out using the same number of initial labeled examples L, distributed accordingly to the class distribution of the complete dataset. As in SSL we are concerned with small labeled sets, we fixed this number as 30, which corresponds to 1.5% of the number of examples in the largest dataset. It is possible to use the class distribution of the complete dataset because we are executing the algorithms in a simulation mode, where the true labels of the datasets are known. Previous experiments have shown that the best results are obtained if the true class distribution of the examples is known [14]. However, in a real case, it would be unwise to estimate this distribution using the few labeled examples in L.

For the sake of simplicity, consider that examples belong to two possible classes $\{\oplus, \ominus\}$. In each iteration of Self-Training, p examples most confidently classified as $\oplus$ and n examples most confidently classified as $\ominus$ are selected. The confidence score is measured by the MNB estimative of $P(\oplus|\mathbf{x})$. As for Co-Training, the same procedure is carried out for each view, and the confidence score is measured by the estimative of $P(\oplus|\mathbf{x}_1)$ in the first view and $P(\oplus|\mathbf{x}_2)$ in the second view. Moreover, if an example is selected in both views and $h_1(\mathbf{x}_1) \neq h_2(\mathbf{x}_2)$, the chosen label is the one given with higher confidence (ties broken randomly). For the unbalanced dataset Courses $(p, n) = (2, 8)$, and for the remaining datasets $(p, n) = (5, 5)$. Considering the different class distribution of the datasets, $p+n = 10$ is the minimum number that covers the class distribution of all the datasets. This means that in each iteration, 10 examples from U' are labeled by Self-Training while a minimum of 10 and a maximum of 20 examples can be labeled by Co-Training.

4.3 Results

In what follows, 1-gram represents the unigram view representation, 2-gram the bigram representation, and 1+2-gram represents the two views joined together in Self-Training. Tables 3 and 4 summarize the experimental results. We use the AUC, mean error rate (%Error) and the number of incorrectly labeled examples (#Errors) to assess the experiments. Values between brackets refer to the standard deviation. The AUC and the mean error rate refer to the classification on the test set, *i.e.* to the final classifiers generated by Self-Training or Co-Training. On the other hand, #Errors refers to the mean number of incorrectly labeled examples during the training phase, where the set L is incremented with new labeled examples.

Table 3. Self-Training results

Dataset	View	Self-Training		
		AUC	%Error	#Errors
Courses	1-gram	0.96 (0.01)	5.43 (1.68)	45.20 (3.80)
	2-gram	0.95 (0.02)	6.29 (1.50)	62.60 (8.11)
	1+2-gram	0.96 (0.01)	5.24 (1.97)	40.40 (2.07)
Hardware	1-gram	0.81 (0.12)	24.61 (10.80)	424.00 (130.93)
	2-gram	0.76 (0.15)	28.94 (10.51)	474.25 (198.17)
	1+2-gram	0.86 (0.06)	18.76 (6.27)	341.89 (105.52)
Vehicles	1-gram	0.99 (0.01)	3.88 (1.57)	74.50 (15.96)
	2-gram	0.98 (0.01)	5.44 (1.03)	137.60 (31.96)
	1+2-gram	0.99 (0.01)	3.28 (1.35)	62.50 (4.88)
Sports	1-gram	0.99 (0.01)	2.16 (0.98)	39.40 (9.74)
	2-gram	0.99 (0.01)	4.07 (0.87)	76.20 (22.72)
	1+2-gram	0.99 (0.01)	1.91 (0.82)	36.90 (8.29)
Movie	1-gram	0.66 (0.07)	36.30 (5.86)	650.40 (85.46)
	2-gram	0.60 (0.05)	42.90 (3.84)	759.40 (65.32)
	1+2-gram	0.65 (0.07)	37.45 (6.03)	643.80 (123.36)

As can be observed in Table 3, Self-Training results do not show a consistent improvement by joining together unigrams and bigrams (1+2-gram). Although the worst results are the ones where only bigrams (2-gram) are used, results using only unigrams (1-gram) are compatible with the ones obtained using 1+2-gram, except for the Hardware dataset.

In Co-Training, if we do not allow the classifiers h_1 and h_2 to label examples for each other, we are left with Self-Training using unigrams and bigrams

Table 4. Co-Training results

Dataset	View	Co-Training		
		AUC	%Error	#Errors
Courses	1-gram	0.97 (0.01)	5.52 (2.19)	
	2-gram	0.96 (0.02)	6.00 (1.49)	44.50 (1.96)
Hardware	1-gram	0.87 (0.06)	20.28 (7.76)	
	2-gram	0.84 (0.07)	23.88 (6.96)	378.33 (39.80)
Vehicles	1-gram	0.99 (0.01)	4.36 (1.45)	
	2-gram	0.98 (0.01)	6.38 (0.30)	102.30 (27.65)
Sports	1-gram	0.99 (0.01)	1.96 (1.10)	
	2-gram	0.99 (0.01)	3.96 (1.19)	42.80 (15.20)
Movie	1-gram	0.67 (0.09)	37.55 (7.42)	
	2-gram	0.65 (0.09)	38.70 (7.36)	681.10 (128.16)

alone (remember from Section 2 that Co-Training is the multi-view counterpart of Self-Training). For this reason, combining unigrams and bigrams with Co-Training is worthwhile only when the classifiers obtained by Co-Training achieve better classification performance than the classifiers obtained by Self-Training using unigrams or bigrams alone. Observing Tables 3 and 4, it is possible to verify that Co-Training is not better than Self-Training using unigrams (1-gram) or bigrams (2-gram) on the Vehicles and Movie datasets. On the Hardware dataset, the inverse situation occurrs, and on the other datasets, the results of both are similar.

5 Discussion and Future Work

The results reported in the previous section do not show an improvement in classification performance when we combine unigrams and bigrams in a single-view and in a multi-view semi-supervised setting. Pursuing a possible explanation for these results, we plotted, for each dataset, a histogram of the mean number of examples correctly and incorrectly labeled by Co-Training versus $p_1 + p_2$, where $p_1 = P(\oplus|\mathbf{x}_1)$ and $p_2 = P(\oplus|\mathbf{x}_2)$. The value of p_1+p_2 for a given example $\mathbf{x}$ will be close to 0 if both h_1 and h_2 classify $\mathbf{x}$ as $\ominus$ with high confidence. On the other hand, the value of $p_1 + p_2$ will be close to 2 if both h_1 and h_2 classify $\mathbf{x}$ as $\oplus$ with high confidence. Figures 1, 2, and 3 show some histograms for the Sports, Hardware, and Movie datasets. All other histograms can be found in `http://www.icmc.usp.br/~igorab/hist/`.

It is possible to observe in Figure 1 that most of the correctly labeled examples in the Sports dataset fall in the bins where $p_1 + p_2$ equals or is next to 0 or 2. In other words, for most of the examples labeled by Co-Training in this dataset, the classifiers h_1 and h_2 agree in their classification with high confidence. Simply put, this means that one classifier did not help the other, what causes the results in this dataset to be very close to the ones of Self-Training using unigrams or bigrams alone.

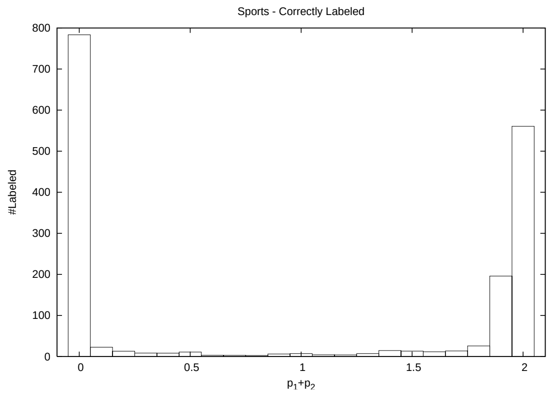


Fig. 1. Number of correctly labeled examples (#*Labeled*) versus $p_1 + p_2$ in the SPORTS dataset

The same pattern was observed in the histograms of the other datasets. In the HARDWARE dataset, for which the results are favorable to combining unigrams and bigrams, most of the correctly labeled examples still fall near 0 or 2, although the histogram in Figure 2 shows relatively more correctly labeled examples in other bins. Furthermore, in the MOVIE dataset, for which both CO-TRAINING and SELF-TRAINING perform equally bad, the histogram in Figure 3 shows that most of the incorrectly labeled examples are next to bins 0 and 2. In other words, both classifiers h_1 and h_2 agree with high confidence in incorrectly labeling these examples.

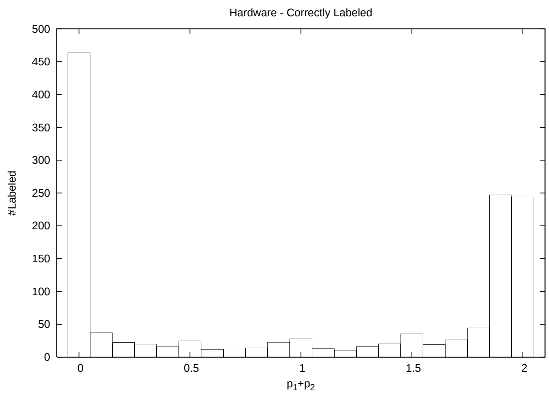


Fig. 2. Number of correctly labeled examples (#*Labeled*) versus $p_1 + p_2$ in the HARDWARE dataset

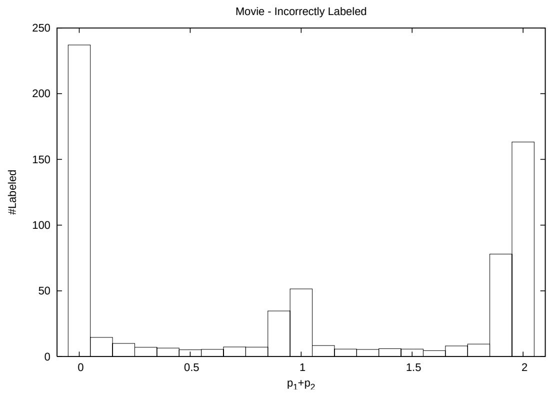


Fig. 3. Number of incorrectly labeled examples (*#Labeled*) versus p_1+p_2 in the MOVIE dataset

In summary, these histograms show a case of strong *view correlation* for unigrams and bigrams, since the classifiers derived on each of them predicts equally with high confidence in most of the examples. However, one of the requirements for CO-TRAINING to succeed is that the views should not be too correlated [6]. This seems a plausible explanation for why combining unigrams and bigrams in CO-TRAINING does not give better results than SELF-TRAINING using unigrams and bigrams alone. We conjecture that this view correlation is also responsible for the small improvements observed when unigrams and bigrams are joined together in SELF-TRAINING and in plain supervised learning.

As future work, we plan to extend the experimental evaluation to include more datasets. We also want to explore a text representation scheme based on carefully selected bigrams (or even trigrams). We would like to combine unigrams only to n-grams that, in a given domain, are more meaningful together than apart. This selection of bigrams can start by using methods for term extraction, which is a subject widely studied in the Natural Languague Processing community [15, 16].

Acknowledgments. The authors wish to thank research fund agencies FAPESP and CNPq for their continuous support.

References

1. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34**(1) (2002) 1–47
2. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: EMNLP '02: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing. (2002) 79–86

3. Bekkerman, R., Allan, J.: Using bigrams in text categorization. Technical report, University of Massachusetts (2003) www.cs.umass.edu/~ronb/papers/bigrams.pdf.
4. Chapelle, O., Schölkopf, B., Zien, A.: Introduction to semi-supervised learning. In: Semi-Supervised Learning (Adaptive Computation and Machine Learning). (2006) 1–12
5. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with Co-Training. In: COLT '98: Proceedings of the 11th Annual Conference on Computational Learning Theory. (1998) 92–100
6. Balcan, M.F., Blum, A., Yang, K.: Co-Training and expansion: Towards bridging theory and practice. In: NIPS '04: Advances in Neural Information Processing Systems 17. (2005) 89–96
7. Tan, C.M., Wang, Y.F., Lee, C.D.: The use of bigrams to enhance text categorization. Information Processing and Management **38**(4) (2002) 529–546
8. Caropreso, M.F., Matwin, S., Sebastiani, F.: A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In: Text databases & document management: theory & practice. (2001) 78–102
9. Nigam, K., McCallum, A., Mitchell, T.: Semi-supervised text classification using em. In: Semi-Supervised Learning (Adaptive Computation and Machine Learning). (2006) 33–53
10. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of Co-Training. In: CIKM '00: Proceedings of the 9th International Conference on Information and Knowledge Management. (2000) 86–93
11. Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML '99: Proceedings of the 16th International Conference on Machine Learning. (1999) 200–209
12. Porter, M.F.: An algorithm for suffix stripping. Program: electronic library and information systems **40**(3) (2006) 211–218
13. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. Technical Report WS-98-05, Association for the Advancement of Artificial Intelligence (1998) http://www.aaai.org/Papers/Workshops/1998/WS-98-05/WS98-05-007.pdf.
14. Matsubara, E.T., Monard, M.C., Prati, R.C.: On the class distribution labelling step sensitivity of Co-Training. In: IFIP AI '06: Artificial Intelligence in Theory and Practice. (2006) 199–208
15. Dias, G., Guillore, S., Bassano, J.C., Lopes, J.G.P.: Combining linguistics with statistics for multiword term extraction: A fruitful association? (2000) 1473–1491
16. Pantel, P., Lin, D.: A statistical corpus-based term extractor. In: AI '01: Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence. (2001) 36–46

Extraction of Definitions in Portuguese: An Imbalanced Data Set Problem

Rosa Del Gaudio and António Branco

University of Lisbon
Faculdade de Ciências, Departamento de Informática
NLX - Natural Language and Speech Group
Campo Grande, 1749-016 Lisbon, Portugal
rosa@di.fc.ul.pt antonio.branco@di.fc.ul.pt

Abstract. Definition extraction is an important task in NLP and IR fields in the context of e.g. question answering, ontology learning, dictionary and glossary construction. When addressed with learning algorithms, it turns out to be a challenging task due to the structure of the data set, the reason being that the definition-bearing sentences are much fewer than the sentences that are non definitions. In this paper, we present results from experiments that seek to obtain optimal solutions for this problem by using a corpus written in the Portuguese language. Our results show an improvement of 29 points regarding AUC metric and more than 60 points when considering the F-measure.

Key words: automatic definition extraction, machine learning, imbalanced data set.

1 Introduction

Definition Extraction is an important task in Natural Language Processing (NLP) and Information Retrieval (IR), in the context of e.g. Question Answering (QA), ontology learning, dictionary and glossary construction, etc.

The interest on definitions dates back to Antiquity. According to Aristotle, the formal structure of a definition should resemble an equation with the *definiendum* (what is to be defined) on the left hand side and the *definiens* (the part which is doing the defining) on the right hand side. The *definiens* should consist of two parts: the *genus* (the nearest superior concept) and the *differentiae specificae* (the distinguishing characteristics). In this way, definitions would adequately capture the concept to be defined.

A thorough study[1] on dictionary entries and definitions automatically extracted presents a description of the linguistic structure of definition sentences, identifying 16 different types of definitions. Nevertheless, in fields such as QA most of the research is focused on the extraction of a definition in a sentence composed by a subject, a copular verb and a predicative phrase.

In this field, the particular type of question, termed as definition question or "what" question, presents characteristics that differentiate it from other questions. All other types of questions, introduced by "who", "when", etc., give some clues on the type of answer which is supposed to be obtained. For example, which semantic type of named entity would provide a better answer to the question. In the case of definition questions, the space of answers is open and this implies that this class of questions needs specific techniques to be dealt with. In particular, when learning algorithms are used, this broadness gives rise to an imbalanced data set, which, depending on the corpus and the techniques used, may present different degrees of imbalance. For example, using a corpus consisting of encyclopedic text and web documents, [2] report that only 18% of the sentences were definitions. On the other hand, using only encyclopedic documents, [3] had a balanced corpus where the definition-bearing sentences represent 59% of the whole corpus.

Similarly to other works in this field, in this paper a definition is considered to be a sentence containing an expression (the *definiendum*) and its definition (the *definiens*), connected by the verb "to be", as in the example "FTP is a protocol that allows the transfer of archives from a place to another through the Internet." The corpus used in the work reported in this paper is composed mostly of tutorials and scientific papers in the Information Technology field, where the definition-bearing sentences were manually annotated and represent the 9% of all sentences.

The definition extraction problem can thus be represented as a binary classification task, where for each sentence in the corpus it is possible to assign the correct class: "definition" or "no definition".

In this paper, we present results obtained with the application of several sampling approaches to our imbalanced data set in order to build more effective classifiers for the definition extraction problem. We try to keep our model as general as possible in order for it to be applicable in different domains. For this reason, we only use information on part of speech (POS) as a feature. This makes the present approach viable for all those languages that are not equipped with rich lexical resources as learning data or in a situation where the domain is too specific to benefit from such resources.

Our task handles several aspects that are common to different machine learning tasks in NLP applications: small amounts of data, inherent ambiguity, noisy data (human annotators make mistakes), imbalanced class distribution, this last aspect being the main issue addressed in this paper.

The rest of the paper is organized in the following way: Section 2 reports on work in the area of definition extraction. Section 3 gives a brief description of the corpus used. Sections 4 and 5 report respectively on the algorithms and on the sampling techniques used. Section 5 is devoted to discussing possible evaluation metrics to be used in the case of imbalanced data sets. Finally, Section 6 presents the results and in Section 7 conclusions are drawn and future directions of our investigation are put forward.

2 Related Work

Previous research on automatic extraction of definitions explored the lexico-syntactic patterns in texts taking into consideration mainly POS or lemmas as main linguistic features. Since the 90's, several authors have proposed methods to identify lexico-syntactic patterns [4, 5].

DEFINDER [6] is an automatic definition extraction system that combines simple cue-phrases and structural indicators introducing the definitions and the defined term. It was developed with a well-structured medical corpus, where 60% of the definitions are introduced by a set of limited text markers. The nature of the corpus used can explain the high performance obtained by this system (87% precision and 75% recall).

Malaise and colleagues [7] developed a system for the extraction of definitory expressions containing hyperonym and synonym relations from French corpora, using corpora from different domains for training and testing. These authors used lexical-syntactic markers and patterns to detect at the same time definitions and relations. For the two different relations (hyponym and synonym), they obtained, respectively, 4% and 36% of recall, and 61% and 66% of precision.

More recently, machine learning techniques were combined with pattern recognition in order to improve the general results. In particular, [3] used a maximum entropy classifier to extract definitions in order to distinguish actual definitions from other sentences. They propose several attributes to classify definition sentences, namely text properties (such as n-gram and bag-of-words), sentence position, syntactic properties and named entity classes. The corpus used was derived from medical pages of the Dutch Wikipedia, from which they extracted sentences based on syntactic features, ending up with 2,299 senteces of which 1,366 are actual definitions. This gives an initial accuracy of 59%, that was improved with machine learning algorithms up to 92.21%

In [8], a system to extract definitions from off-line documents is presented. They experimented with three different algorithms, namely Naïve Bayes, Decision Tree and Support Vector Machine (SVM), obtaining the best score with SVM with a F-measure of 0.83 with a balanced data set.

Westerhout and Monachesi [9] combine syntactic patterns with a Naïve Bayes classification algorithm with the aim of extracting glossaries from tutorial documents in Dutch. They used several properties and several combinations of them, obtaining an improvement in precision of 51.9%, but a decline in the recall of 19.1% in comparison with the syntactic pattern system developed previously by these authors, using the same corpus.

In spite of the increasing attention the imbalanced data set issue has attracted in the machine learning community, shown by two different workshops held in 2000 [1] and 2003 [2], little attention has beeen payed to the data set structure of the definition detection task when addressed with machine learning techniques.

[1] Japkowicz, editor, Proceedings of the AAAI2000 Workshop on Learning from Imbalanced Data Sets, AAAI Tech Report WS-00-05.

[2] N. V. Chawla, N. Japkowicz, and A. Kolcz, editors, Proceedings of the ICML2003 Workshop on Learning from Imbalanced Data Sets. 2003.

Recently, some authors have started to look at this problem of imbalanced data sets in the context of definition extraction. In particular, [10] down-sampled their corpus using different ratios (1:1, 1:5, 1:10) in order to seek for best results. The corpus they used presented an original ratio of non-definitions to definitions of about 19. Although they obtained some improvement in terms of the F-measure, in particular with the ratio 1 to 5, they couldn't improve results obtained with a rule based grammar previously developed using the same corpus. These authors also investigated the use of Balanced Random Forest algorithm in order to deal with this imbalance, succeeding in outperforming the rule based grammar previously developed by 5 percentage points [11].

3 Data Set Description

The corpus used for the experiments was collected in the context of the LT4eL project [12]. It was used to develop different tools, such a keyword extractor, a glossary candidate detector and an ontology, in order to support e-learning activities[13, 14]. The corpus is composed of several tutorials and scientific papers in the field of Information Technology and has a size of 274,000 tokens. It was automatically annotated with morpho-syntactic information using the LX-Suite [15, 16].

Definition-bearing sentences were manually annotated. In each sentence, the term defined, the definition and the connection verb were annotated using a different XML tag. As the focus of this work are definitions conveyed by the verb "to be", a simple grammar was developed in order to extract all the sentences with this verb as main verb. A sub-corpus was obtained, composed by 1,360 sentences, 121 of which are definitions, with a ratio of about 10:1.

As for feature selection, works in similar areas tend to use different types of properties (text, document, syntactic, etc.). Examples of text properties are bag-of-words and n-grams [17] either of part-of-speech or of base forms. Regarding document properties, the position of the definition inside the document is often used as a property [18], as well as the presence of determiners in the *definiens* and in the *definiendum* [3]. Other relevant properties can be the presence of named entities [3] or data from an external source such as encyclopedic data, wordnets, etc. [19].

Most of these features are strictly related to the corpus used, rendering generalizations for other corpus very difficult. For example, in [3] the use of the position of a definition-bearing sentence as a feature is based on the observation that definitions tend to occur at the beginning of a document, but the corpus used in their work was based on wikipedia articles, and this is just a characteristic of this public encyclopedia. Similar problems arise when information other than part of speech is used as a feature. In this case, the results obtained are typically hard to generalize to other text domains.

For all these reasons, in the present work, instances were represented as n-grams of POS. Different configurations were tested with n ranging from 1 to 4. From all POS n-grams extracted from the set of 1,360 definitions, the 100 most frequent were used as features. Each sentence was represented as an array where cells record the number of occurrences of these n-grams. In this paper, due to the limited number of pages available, only results obtained with the best representation are shown, that is with bi-grams.

4 Machine Learning Algorithms

When selecting learning algorithms, two different considerations were taken into account. First, we want to use those algorithms that in literature represent the state of the art for definition extraction and also for imbalanced data sets problems. Second, we want to cover different classes of algorithms, having at least a representative algorithm for different classes. In this way, results obtained with different sampling techniques may be generalized to a larger range of algorithms. Five different algorithms were selected: Naïve Bayes, C4.5, Random Forest, k-NN, SVM.

Naïve Bayes is a simple probabilistic classifier that is very popular in Natural Language applications. In spite of its simplicity, it permits to obtain results quite similar to those obtained with more complex algorithms.

C4.5 and Random Forest are two decision tree algorithms. The first is a relatively simple algorithm that splits the data into smaller subsets using the information gain in order to chose the attribute for splitting the data. The second is a classifier consisting of a collection of decision trees. For each tree, a random sample of the data set is selected (the remaining is used for error estimation), and for each node of the tree, the decision at that node is based on a restricted number of variables.

The k-NN algorithm is a type of instance-based learning, also called lazy learning because, unlike the algorithms above, the training phase of the algorithm consists only in storing the feature vectors and class labels of the training samples and all computation is deferred for the classification phase. In this phase, the algorithm computes the distance between the target sample and n samples in the data set, assining the most frequent class. Two different K nearest neighbors classifiers were constructed, with k equal to 1 and to 3.

SVM is a classifier that tries to find an optimal hyperplane that correctly classifies data points as much as possible and separates the point of two classes as far as possible.

All the classifiers were implemented using the Weka workbench [20].

5 Sampling Techniques

In many real-world classification applications, most of the examples are from one of the classes, while the minority class is the interesting one. As most of the learning algorithms are designed to maximize accuracy, the imbalance in the class distribution leads to a poor performance of these algorithms. The issue is therefore how to improve the classification of the minority class examples.

A common solution is to sample the data, either randomly or intelligently, to obtain an altered class distribution.

Random over-sampling consists of random replication of minority class examples, while in random down-sampling majority class examples are randomly discarded until the desired amount is reached. These two very simple methods are often criticized due to their drawbacks. Several authors pointed out that the problem with under-sampling is that this method can discard potentially useful data that could be important for the induction process. On the other hand, random over-sampling can increase the likelihood of overfitting, since it makes exact copies of the minority class examples.

When speaking about negative and positive examples in a data set, it is important to have in mind that not all the examples have the same value. There are examples that are more prototypical than others and represent better the class to which they belong, while others are too similar to be useful, and others are just noise.

Building on these considerations, several methods were proposed in order to retain safe examples in the re-balanced data set. We consider here two of such methods, namely the Condensed Nearest Neighbour Rule and Tomek Links algorithm.

Condensed Nearest Neighbor Rule (CNN) [21] finds a consistent subset of examples in order to eliminate the examples from the majority class that are distant from the decision border, since these examples might be considered less relevant for learning. The CNN is sensitive to noise and noisy examples are likely to be misclassified as many of them will be added to the training set.

Tomek Links [22] removes both noise and borderline examples. Tomek Links are pairs of instances of different classes that have each other as their nearest neighbors. As an under-sampling method, only examples belonging to the majority class are eliminated. The major drawback of Tomek Links under-sampling is that this method can discard potentially useful data that could be important for the induction process. This method has an higher order computational complexity and will run slower than other algorithms.

While the previous methods are intelligent down sampling techniques, SMOTE is an over-sampling method that produces new synthetic minority class examples. SMOTE [23] forms new minority class examples by interpolating between several minority class examples that lie together in "feature space" rather than "data space". For each minority class example, this algorithm introduces synthetic examples along the line segments joining any/all of the k minority class nearest neighbors (in this work k is equal to 3).

6 Evaluation Issues

Using the confusion matrix in Table 1 as a starting point, we discuss the possible metrics for the evaluation of the classifiers investigated in this work. One of the most used metrics is the Error Rate, defined as 1.0-(TP+TN)/(TP-FP+FN+TN). However, using this metric implies that the class distribution is known and fixed, an assumption that does not hold in real world applications as the one proposed here. Moreover, Error Rate is biased to favor the majority class, making it a bad choice when evaluating the effects of class distribution. Another aspect against the use of Error Rate is that it considers different classification errors as equally important, and in domains such medical diagnosis, the error of diagnosing a sick patience as healthy is a fatal error while the contrary is considered a much less serious error. In general, any performance metric, such as accuracy and Error Rate, that uses values from both columns will be sensitive to class imbalance.

	Positive Prediction	Negative Prediction
Positive Class	True Positive (TP)	False Negative (FN)
Negative Class	False Positive (FP)	True Negative (TN)

Table 1. Confusion Matrix for a binary classification problem

Starting from the confusion matrix it is possible to derive metrics that are not sensitive to the skew of the data. In particular, four metrics are proposed in [24]:

– False Negative Rate: FN/(TP+FN) - the percentage of positive examples misclassified as belonging to the negative class
– False Positive Rate: FP/(FP+TN) - the percentage of negative examples misclassified as belonging to the positive class
– True Negative Rate: TN/(FP+TN) - the percentage of negative examples correctly classified as belonging to the negative class
– False Positive Rate: TP/(TP+FN) - the percentage of positive examples correctly classified as belonging to the positive class

A good classifier should try to minimize FN and FP rates, and maximize TN and TP rates.

Unfortunately, there is a tradeoff between these two metrics, and in order to analyze this relationship ROC graphs are used. ROC graphs are two-dimensional graphs where the TP rate is plotted on the Y axis and the FP rate is plotted on the X axis. ROC graphs are consistent for a given problem even if the distribution of positive and negative instances is highly skewed.

In order to compare classifiers, it is possible to reduce a ROC curve to a scalar value representing the performance of the classifier. The area Under the ROC (AUC) is a portion of the area of the unit square. Its value will always be between 0 and 1. However, because random guessing produces the diagonal ine between (0,0) and (1,1), which has an area of 0.5, no realistic classier should have an AUC less than 0.5.

In this work, we will use the AUC measure in order to assess the performance of classifiers. Furthermore, for each classifier, we present also the F-measure[3] in order to compare our results with the results of previous works in this area.

7 Results and Discussion

In this section, we present the results for the different learning algorithms used, namely Naïve Bayes, C4.5, Random Forest, k-NN and SVM. For each classifier, results regarding the different sampling techniques discussed above in Section 5, that is, random over and down sampling, SMOTE, CNN and Tomek Links are shown. We also present results obtained using the original data set, which is the data set with the original imbalance. This result represents our base line against which results obtained with sampled data sets are to be compared with. Values in bold represent the best score for each classifier.

Since the data set size does not allow us to split the corpus into two samples, a training set and a test set, 10-fold cross validation was used.

Tables 2 and 3 display the performance of the two classifiers using k-NN algorithm. In particular, Table 2 reports on the results of the most basic implementation of k-NN, that is with k equal to 1 (1-NN). In this case, a test example is simply assigned to the class of its nearest neighbor. Table 3 displays results obtained by a classifier using k-NN algorithm with k equal to 3 (3-NN).

Regarding the results in Table 2, it is possible to notice that, for the AUC metric, only the SMOTE sampling technique is able to significantly improve the base line, obtaining a score of 0.66 with an improvement of 10 points. If we focus on the F-measure, there is a substantial improvement with the different techniques, namely SMOTE and random down-sampling. As for AUC metric, also when considering the F-measure, the SMOTE presents the best score, namely 0.63 with an improvement on base line of 42 points.

Regarding results in Table 3, there are 4 sampling techniques that outperform the base line for F-measure: SMOTE (with the best score), followed by CNN, Tomek Links and random down-sampling. As to the AUC metric, the best performance is achieved by SMOTE and Tomek Links, with an improvement of 13 and 9 points respectively in comparison with the base line. Although the base lines for the classifiers above are very similar, they differ in the way they respond to the sampling techniques. In particular, the 3-NN algorithm seems to take more advantage from the use of sampling, since it obtains better results in all the experiments.

Tables 4 and 5 show the performance of the two classifiers based on decision tree algorithms, namely the C4.5 and Random Forest. The results displayed in Table 4 refer to the best setting for the C4.5 classifier, where the tree was pruned using the C4.5 standard pruning procedure with no Laplace correction. Regarding Table 5, the classifier was built using 10 different trees.

Similarly to previous classifiers, SMOTE sampling method presents the best results in terms of AUC for both classifiers, with a rise of 23 and 29 points for

[3] $F - measure = \frac{2*Precision*Recall}{(Precision+Recall)}$

1-NN

Sampling	F-m	AUC
Original	0.19	0.56
Downsampling	0.62	0.57
Oversampling	0.36	0.55
SMOTE	**0.63**	**0.66**
CNN	0.23	0.52
Tomek	0.57	0.59

Table 2. Results obtained for the classifier using **k-NN** algorithm with **k=1**

3-NN

Sampling	F-m	AUC
Original	0.17	0.57
Downsampling	0.62	0.59
Oversampling	0.51	0.58
SMOTE	**0.66**	**0.70**
CNN	0.65	0.61
Tomek	0.64	0.66

Table 3. Results obtained for the classifier using **k-NN** algorithm with **k=3**

C4.5 and Random Forest respectively. The same observation holds for the F-measure, with an improvement of 60 and 63 points respectively. For this metric, good results are also achieved by Tomek Links and CNN.

C4.5

Sampling	F-m	AUC
Original	0.17	0.65
Downsampling	0.58	0.59
Oversampling	0.37	0.67
SMOTE	**0.77**	**0.87**
CNN	0.62	0.61
Tomek	0.63	0.60

Table 4. Results obtained for the classifier using **C4.5** algorithm

Random Forest

Sampling	F-m	AUC
Original	0.13	0.65
Downsampling	0.57	0.65
Oversampling	0.21	0.64
SMOTE	**0.75**	**0.94**
CNN	0.59	0.66
Tomek	0.65	0.59

Table 5. Results obtained for the classifier using **Random Forest** algorithm

Table 6 displays results obtained with a SVM classifier using a sigmoid kernel. The AUC base line for this classifier is very low, with a value below 0.5. Using sampling techniques the performance of this classifier is comparable to the 1-NN, reaching an AUC of 0.68 with random down-sampling. It is interesting to observe that although SVM is a complex algorithm, it achieves a performance similar to the simplest algorithm used in this work, namely 1-NN. Furthermore it is the only classifier where the SMOTE does not show the best result, considering either AUC or F-measure. Only the classifier based on SVM presents the best result when coped with the random down-sample method.

The results in Table 7 refer to a Naïve Bayes classifier using normal distribution. The base line for this classifier is higher than for the other classifiers in terms of both metrics taken in consideration. Nevertheless, the improvements achieved with the use of sampling do reach the performance of other classifiers, namely C4.5 and Random Forest.

In general, the SMOTE sampling technique shows the best results in terms of AUC, followed by Tomek Links and random over-sampling. The best score for SMOTE is achieved by Random Forest with 0.94 followed by C4.5 with 0.87.

SVM		
Sampling	F-m	AUC
Original	0.12	0.48
Downsampling	**0.67**	**0.68**
Oversampling	0.61	0.59
SMOTE	0.60	0.60
CNN	0.59	0.57
Tomek	0.64	0.49

Table 6. Results obtained for the classifier using **SVM** algorithm

Naïve Bayes		
Sampling	F-m	AUC
Original	0.24	0.66
Downsampling	0.62	0.62
Oversampling	0.67	0.68
SMOTE	**0.72**	**0.76**
CNN	0.64	0.63
Tomek	0.69	0.72

Table 7. Results obtained for the classifier using **Naïve Bayes**

These results are comparable with those reported in the literature on imbalanced data sets in general. In a comprehensive study on the behavior of several methods for balancing training data, using the 11 UCI data set [4], Batista and colleagues [24] show that in most of the cases and with several data sets in different domains SMOTE and random over-sampling are the most effective methods. In general, they obtain a rise in the AUC metric of few percentage points (1 to 4), when the base line was already high (more than 0.65); when the base line was under this value the improvement was comparable to the one obtained in our work. In particular for the flag data set, they obtained an improvement of 34 percentage points.

Focusing on Natural Language applications, [25] apply these methods to sentence boundary detection in speech, showing that SMOTE and random down-sampling get the best results with an AUC of 0.89 (the base line being 0.80). However, they did not experiment intelligent down-sampling methods such as CNN or Tomek Links. Batista [26], in a case study on automated annotation of keywords , gets the best results in terms of AUC with an improvement of 4 percentage points on the original data set using a combination of SMOTE with Tomek Links, followed by simple SMOTE.

In our case the improvement regarding the original data set is between 10 and 29 points, demonstrating how these methods can be effective in this application.

Regarding the comparison with other work in definition extraction, the improvement obtained on the F-measure, with the best result of 0.77 with C4.5 classifier, outperforms most of the systems presented in Section 2, confirming the importance of sampling techniques in supporting definition extraction tasks. For instance, [9] reports on a F-measure of 0.73, obtained with a combination of syntactic rules and a Naïve Bayes classifiers for the Dutch language, in turn, [10], with a similar approach, but for the Polish, obtain a F-measure of 0.35.

Furthermore, in all these works a combination of features is used in order to reach better results, while in this paper we only use bi-grams of POS as feature.

To conclude, our results outperform those systems that represent the state of the art in the area, such as DEFINDER, which shows a F-mesure of 0.80.

[4] http://archive.ics.uci.edu/ml/

8 Conclusions and Future Work

In this work, we presented a study on the better way to deal with imbalanced data sets in the context of definition extraction. We reported results for five classifiers and five different sampling techniques. Our results are comparable to the results obtained in previous work in the area, confirming the SMOTE sampling method as one of the most effective in dealing with imbalanced data sets.

Furthermore, this work empirically demonstrates the effectiveness of sampling methods in the definition extraction field. This finding is supported by the magnitude of the improvement obtained in comparison with the original data sets for both the metrics used. In particular, our results show an improvement of 29 points regarding the AUC metric and more than 60 points when considering the F-measure.

In future work we are planning to experiment with more sampling techniques as well combining them in different ways. Additionally, we want to use different data sets in different languages in order to validate our findings.

References

1. Barnbrook, G.: Defining Language: a local grammar of definition sentences. John Benjamins Publishing Company (2002)
2. Tjong, E., Sang, K., Bouma, G., de Rijke, M.: Developing offline strategies for answering medical questions. In: Proceedings of the AAAI-05 workshop on Question Answering in restricted domains. (2005) 41–45
3. Fahmi, I., Bouma, G.: Learning to identify definitions using syntactic feature. In Basili, R., Moschitti, A., eds.: Proceedings of the EACL workshop on Learning Structured Information in Natural Language Applications, Trento, Italy (2006)
4. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1992) 539–545
5. Person, J.: The expression of definitions in specialised text: a corpus-based analysis. In Gellerstam, M., Jaborg, J., Malgren, S.G., Noren, K., Rogstrom, L., Papmehl, C., eds.: 7th Internation Congress on Lexicography (EURALEX 96), Goteborg, Sweden (1996) 817–824
6. Klavans, J., Muresan, S.: Evaluation of the DEFINDER system for fully automatic glossary construction. In: Proceedings of the American Medical Informatics Association Symposium (AMIA 2001). (2001)
7. Malais, V., Zweigenbaum, P., Bachimont, B.: Detecting semantic relations between terms in definitions. In: the 3rd edition of CompuTerm Workshop (CompuTerm 2004) at Coling 2004. (2004) 55–62
8. Chang, X., Zheng, Q.: Offline definition extraction using machine learning for knowledge-oriented question answering. In Huang, D.S., Heutte, L., Loog, M., eds.: ICIC (3). Volume 2 of Communications in Computer and Information Science., Springer (2007) 1286–1294
9. Westerhout, E., Monachesi, P.: Extraction of Dutch definitory contexts for elearning purposes. In: CLIN proceedings 2007. (2007)

10. Przepiorkowski, A., Marcinczuk, M., Degorski, L.: Noisy and imbalanced data: Machine learning or manual grammars? In: Text, Speech and Dialogue: 9th International Conference, TSD 2008, Brno, Czech Republic, Lecture Notes in Artificial Intelligence, Berlin, Springer-Verlag (2008)
11. Kobylinski, L., Przepiorkowski, A.: Definition extraction with balanced random forests. In Ranta, A., ed.: GoTAL 2008, Gothenburg, Springer-Verlag Berlin Heidelberg (2008) 237–247
12. Monachesi, P., Lemnitzer, L., Simov, K.: Language technology for elearning. In: Proceedings of EC-TEL, Springer LNCS (2006)
13. Avelãs, M., Branco, A., Gaudio, R.D., Martins, P.: Supporting e-learning with language technology for portuguese. In: Proceedings of the International Conference on the Computational Processing of Portuguese (PROPOR2008), Springer (2008)
14. Gaudio, R.D., Branco, A.: Learning to identify definitions using syntactic feature. In: Progress in Artificial Intelligence, 13th Portuguese Conference on Aritficial Intelligence, EPIA 2007, Workshops: GAIW, AIASTS, ALEA, AMITA, BAOSW, BI, CMBSB, IROBOT, MASTA, STCS, and TEMA, Guimarẽs, Portugal, Springer Berlin (2007) 659–670
15. Silva, J.R.: Shallow processing of Portuguese: From sentence chunking to nominal lemmatization. Master's thesis, Universidade de Lisboa, Faculdade de Ciências (2007)
16. Branco, A., Silva, J.R.: Lx-suite: Shallow processing tools for portuguese. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06). (2006) 179–183
17. Miliaraki, S., Androutsopoulos, I.: Learning to identify single-snippet answer to definition questions. In: Proceeding of the 20th International Conference on Computational Linguistic (COLING 2004), Geneva, Switzerland (2004) 1360–1366
18. Joho, H., Sanderson, M.: Retrieving descriptive phrases from large amounts of free text. In: Proceeding of hte 9th international conference on Information and knowledge management. (2000) 180–186
19. Saggion, H.: Identifying definitions in text collections for question answering. In: LREC 2004. (2004)
20. Witten, I.H., Frank, E.: Data Mining: Pratical Machine Learning Tools and Techniques (Second Edition). Morgan Kaufmann (2005)
21. Hart, P.E.: The condensed nearest neighbor rule (corresp.). Information Theory, IEEE Transactions on **14**(3) (1968) 515–516
22. Tomek, I.: Two modifications of cnn. Systems, Man and Cybernetics, IEEE Transactions on **6**(11) (1976) 769–772
23. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16** (2002) 321–357
24. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. Newsl. **6**(1) (2004) 20–29
25. Liu, Y., Chawla, N.V., Harper, M.P., Shriberg, E., Stolcke, A.: A study in machine learning from imbalanced data for sentence boundary detection in speech. Computer Speech & Language **20**(4) (2006) 468–494
26. Batista, G.E.A.P.A., Bazzan, A.L.C., Monard, M.C.: Balancing training data for automated annotation of keywords: a case study (2003)

Parallel Texts Alignment

Luís Gomes, José Aires, and Gabriel Pereira Lopes

CITI, Departamento de Informática,
Faculdade de Ciências e Tecnologia,
Universidade Nova de Lisboa,
2829-516 Caparica, Portugal
luismsgomes@gmail.com, {aires,gpl}@di.fct.unl.pt

Abstract. Alignment of parallel texts (texts that are a translation of each other) is a step required by many applications that use parallel texts, including statistical machine translation, automatic extraction of translation equivalents, automatic creation of concordances, etc.
Most of existing methods for parallel texts alignment try to infer simultaneously a bilingual word lexicon and a set of correspondences between the occurrences of those words in the texts. Some authors suggest that an external lexicon can be used to complement the inferred one, but they tend to consider it secondary/optional. We defend that lexicon inference should not be embedded in the alignment process, and present LEXIC-AL, a new alignment method that relies exclusively on externally managed lexicons. In our experiments with the European Constitution corpus, LEXIC-AL achieves 84.45% precision and 84.55% recall.

Key words: parallel texts alignment, parallel corpora, extraction of translation equivalents

1 Introduction

Simard et al [17], Davis et al [5], Melamed [11, 12], Ribeiro et al [16, 8] and Ribeiro [15] use cognates as lexical cues for alignment. However, the number of cognates and loan words is highly dependent on the languages of the texts being aligned, as noted by Melamed [12] and confirmed by the results of the evaluation carried by Bilbao et al [2] on the impact of cognates on alignment. Melamed [12] suggests using an external bilingual lexicon in addition to cognate matching to increase the number of correspondences. However, further ahead we present several arguments supporting that we should go one step further and use *only* an external lexicon; the identification of cognate words shouldn't be done within the alignment process.

Kay and Röscheisen [9], Chen [3], Fung and Church [6] and Fung and McKeown [7] infer (by different methods) a bilingual lexicon as part of the alignment process and use that lexicon to establish correspondences between words in the two parallel texts. Like for the identification of cognates mentioned above, we defend that alignment should not be entangled with lexicon inference and instead it should rely solely on an externally managed bilingual lexicon. This separation of alignment and lexicon extraction is supported by the following arguments:

1. Using exclusively an externally managed lexicon gives us *full control over the lexicon that is used to align.* By contrast, it is not possible to avoid eventual bogus entries in an inferred lexicon because the whole method is automatic.
2. Because the alignment depends on the size of the lexicon, an external lexicon has greater potential to obtain better alignments because we can enrich the lexicon over time. By contrast, the alignment methods that infer the lexicon automatically do not improve over time, because every time they align a pair of texts, they infer a new lexicon from scratch.
3. We can combine several sophisticated extraction methods to enrich the alignment lexicon. For example, we can use methods that specialize on identification of possible cognates, extraction of multi-word translations, extraction of single word translations, extraction of named entities, etc.

We have developed an extractor of phrase translations from aligned parallel corpora (Aires et al [1]) that has been used to periodically augment our English-Portuguese lexicon. The extracted equivalents are evaluated (marked as accepted or rejected) by a linguist before they are used for alignment.

2 Method outline

Our alignment method is divided in two stages. The first stage is performed independently for each text and is executed in parallel, taking advantage of multiple cores or processors in the computer. In this stage we obtain the list of occurrence offsets for each term of the lexicon occurring in the texts. This is implemented by a procedure named LOOKUP that takes as input a sorted list of terms and a text, and produces a list of occurrence offsets for each term found in the given text. This procedure is further discussed in section 3.

The second stage iterates over two steps: identify corresponding occurrences and select a list of non-crossing occurrences to be used as alignment anchors (see figure 1). Each iteration produces a more precise alignment (with more anchors) and the loop terminates when two consecutive iterations produce the same anchors.

English	Portuguese
the united kingdom	**o reino unido**
shall not be obliged	não ficará obrigado
or	**ou**
committed	comprometido
to adopt	**a adoptar**
the euro	**o euro**
without	**sem**
a	**uma**
separate	
decision	**decisão**
to do so by its	distinta em esse sentido de o seu
government	**governo**
and	**e**
	de o seu
parliament	**parlamento**

Fig. 1: Alignment of a passage of the European Constitution. The segments in bold are alignment anchors, i.e., non-crossing correspondences between occurrences of equivalent terms that are in the lexicon. The other segments contain whatever text exists between each two consecutive anchors.

The main alignment procedure is:

procedure ALIGN(TextX, TextY, ListOfTermsX, ListOfTermsY)
(first stage)
OccursX ← LOOKUP(ListOfTermsX, TextX)
OccursY ← LOOKUP(ListOfTermsY, TextY)
(second stage)
Anchors ← {Origin, Terminus}
repeat
Correspondences ← CORRESPOND(OccursX, OccursY, Anchors)
Anchors ← SELECT(Correspondences)
until two consecutive iterations produce the same Anchors
OUTPUT(Anchors)
end procedure

The CORRESPOND procedure tries to find out which occurrences of a given term (in one text) correspond to which occurrences of an equivalent term (in the other text). It takes three lists as input: two lists of term occurrences (one of each text) and the list of alignment anchors to be used as guide (which consists of a list of non-crossing correspondences). The criteria to decide which occurrences correspond to each other is presented in section 4.

The SELECT procedure selects a list of non-crossing correspondences to be used as alignment anchors according to the criteria given in section 5.

The CORRESPOND and SELECT procedures are executed one after the other in an iterative refinement loop. The output of CORRESPOND is passed to SELECT, and the output of SELECT is passed to CORRESPOND in the next iteration, and so on. To bootstrap this cycle we give two alignment anchors, the origin and the terminus of the texts, to the first invocation of CORRESPOND, which is equivalent to say that we use the *golden diagonal* as an initial alignment approximation. As it turns out, the criteria used to find correspondences is very robust and the crudeness of the alignment given as guide impacts mostly the number of correspondences produced (not their precision), meaning that more iterations are needed for texts that are less parallel. In our experiments, further discussed in section 6, LEXIC-AL needs on average 3 iterations.

3 Lookup lexicon terms in the texts

The LOOKUP procedure is performed independently for each text and may be executed in parallel on multi-core/multiprocessor machines. It takes as input a sorted list of terms and a text, and produces a list of occurrence offsets for each term that occurs in the text. Figure 2 is an excerpt of the output of LOOKUP.

Id	Term	Occurrence offsets
...	...	...
13292	comercialização	26442, 118619
59480	cominação	36238
129400	cominação de multas	36238
129401	cominação de multas e sanções pecuniárias compulsórias	36238
20649	comissão	6938, 8237, 8462, ...
...	...	...

Fig. 2: Excerpt of the output of the LOOKUP procedure executed on the same Portuguese text that was used to create the suffix array in figure 3, where term "comercialização" occurs at offsets 26442 and 118619.

First we construct the suffix array (Manber and Myers [10]) for the text (see figure 3) and then we perform a simultaneous scan through the list of terms and the suffix array, checking *which terms are prefix of which suffixes.*

The pseudocode for LOOKUP is presented below. It produces the output presented in figure 2.

Offset	Suffix
...	...
223627	comercial comum é conduzida de acordo com os princípios e ...
26442	comercialização . ¶ o presente artigo é aplicável a ...
118619	comercialização de os diversos produtos ; medidas de ...
74580	cometidas a o sistema europeu de bancos centrais , o banco ...
75263	cometidas a o sistema europeu de bancos centrais a o abrigo ...
69327	cometidas a o sistema europeu de bancos centrais são : ¶ ...
247838	cometido falta grave . ¶ 3 . ¶ o provedor de justiça europeu ...
255647	cometido falta grave pode ser demitido por o tribunal de justiça ...
...	...

Fig. 3: Part of a suffix array of a Portuguese text. The suffixes in the rightmost column are obtained by printing the text from the offset presented in the first column of the table; besides the text being aligned we only have the column on the left loaded into main memory.

```
procedure LOOKUP(ListOfTerms, Text)
    SuffixArray ← MAKE_SUFFIX_ARRAY(Text)
    Term ← first term in ListOfTerms
    Suffix ← first suffix in SuffixArray
    repeat
        if Term is prefix of Suffix then
            OUTPUT(id of Term)
            OUTPUT(Term)
            OUTPUT(offset of Suffix in the text)
            for all suffixes S next to Suffix having Term as prefix do
                OUTPUT(offset of S in the text)
            end for
            OUTPUT(newline)
            Term ← the next term from ListOfTerms
        else if Term is lexicographically lower than Suffix then
            Term ← the next term from ListOfTerms
        else
            Suffix ← the next suffix in SuffixArray
        end if
    until we have run through all terms or through all suffixes
end procedure
```

This algorithm runs in linear time and it does not impose any limitation to the length of the terms. The list of terms is read sequentially from disk, one term at a time, making this algorithm very economic in terms of memory usage. It is also very fast because data is read sequentially.

4 Finding corresponding occurrences of equivalent terms

This section describes a criteria to decide which occurrences $o_{x:i}$ of a term in text X correspond to which occurrences $o_{y:k}$ of an equivalent term in the other text Y. In Figure 4 we represent the parallel texts as two parallel line segments. The small black rectangles represent occurrences of the word "Commission" in the X (English) text and it's translation, "Comissão", in the Y (Portuguese) text. In the English text the word "Commission" was replaced by the pronoun "it" in one place but not so in the Portuguese text. Thus the number of occurrences in both texts is different. Looking at this representation, which occurrences in X and Y can we assume to correspond?

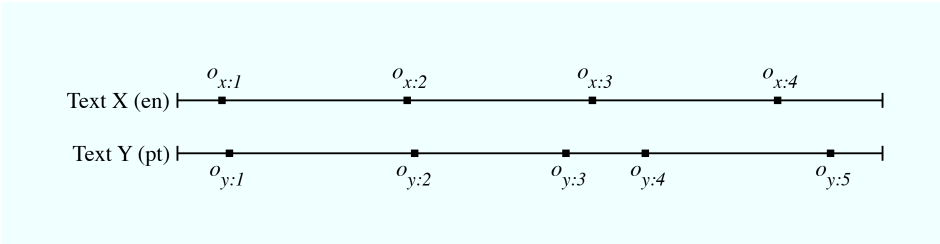


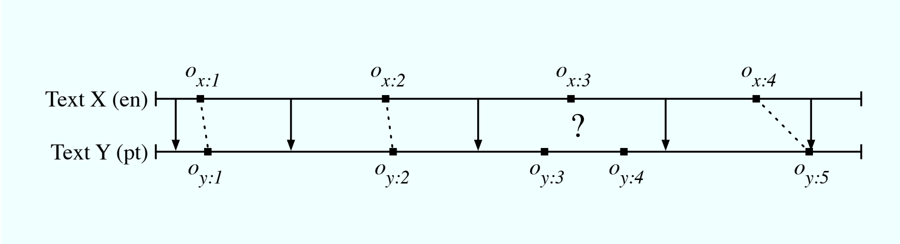
Fig. 4: Correspondences between occurrences of the word "Commission" in the English text (X) and occurrences of the word "Comissão" in the Portuguese text (Y).

We assume a correspondence between occurrences that are roughly at the same position in both texts. But, even if the occurrences are slightly apart, we can still match them if that pair is *isolated* from the other occurrences. In figure 4 the occurrences $o_{x:1}$ and $o_{y:1}$ are close to each other and distant from $o_{x:2}$ and $o_{y:2}$, thus we have no problem assuming that they correspond. The same holds for $o_{x:2}$ and $o_{y:2}$. The occurrences $o_{x:4}$ and $o_{y:5}$ are not so close to each other, but we can, arguably, match them as well because they are distant enough from other occurrences. However, we could not decide which occurrence of Y corresponds to $o_{x:3}$, even though $o_{x:3}$ is closer to $o_{y:3}$ than $o_{x:4}$ is to $o_{y:5}$.

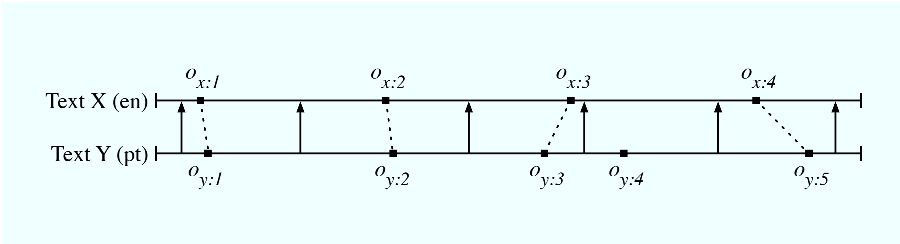
Our criteria to find pairs of occurrences that correspond is based on the distance between the pair and their isolation from other occurrences.

We compute the midpoints between consecutive occurrences in one text and we map them to positions in the other text as described further ahead. The midpoints and their mapped positions are used to divide both texts into a number of corresponding segments as shown in figure 5.

We assume that two occurrences correspond if they are within segments that correspond and if there are no other occurrences in those segments. According to this criteria we find that $o_{x:1}$ corresponds to $o_{y:1}$, $o_{x:2}$ corresponds to $o_{y:2}$ and $o_{x:4}$ corresponds to $o_{y:5}$. The occurrences $o_{x:3}$ and $o_{y:3}$ meet our correspondence criteria for the segments in figure 5b but not for the segments in figure 5a. In our



(a) Segmentation of both texts using midpoints between occurrences in X and their mapped positions in Y.



(b) Segmentation of both texts using midpoints between occurrences in Y and their mapped positions in X.

Fig. 5: Two occurrences are assumed to correspond if they are within segments that correspond and if there are no other occurrences in those segments. Correspondences are represented by dotted lines.

experiments we have considered only correspondences that are confirmed both ways. Therefore, we leave $o_{x:3}$, $o_{y:3}$ and $o_{y:4}$ without correspondence.

Figure 6 presents a polygonal chain that is obtained by connecting the origin of the texts to the lower bounds of the first alignment anchor, the upper bounds of that anchor, the lower bounds of the second anchor, and so on. The polygonal chain ends at the terminus the texts. The sequence of points of the polygonal chain are:

$$((0,0),(l_{x:1},l_{y:1}),(u_{x:1},u_{y:1}),(l_{x:2},l_{y:2}),\cdots,(l_{x:n},l_{y:n}),(u_{x:n},u_{y:n}),(L_x,L_y))$$

To map a position in text X to a position in Y we compute the ordinate of the point in the line having the given abscissa. Conversely, to map a position in Y to a position in X we compute the abscissa of the point in the line having the given ordinate.

5 Selecting a list of non-crossing correspondences

The Longest Sorted Sequence Algorithm (LSSA) described by Ildefonso and Lopes [8] selects a monotonic sequence of points from a set of corresponding

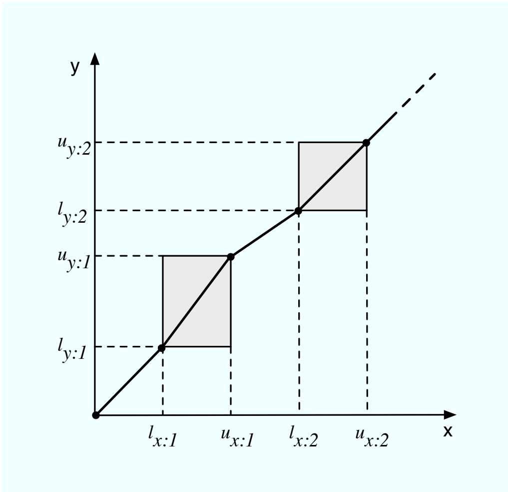


Fig. 6: A monotone polygonal chain obtained from alignment anchors (represented as shaded rectangles).

points under the assumption that the most reliable sequence is the one that includes more correspondence points.

We have observed that correspondences are not all equally reliable, and thus, they should not be all equally weighted when deciding the best alignment. The correspondences between small segments, like correspondences between occurrences of punctuation characters, are usually not as reliable as the occurrences between occurrences of large terms, as for example a correspondence between "rubber products manufacturing" and the Portuguese translation "a indústria transformadora de produtos de borracha". We hypothesize that correspondences between larger segments are more reliable. According to this hypothesis, the list of most reliable anchors is the one that maximizes the sum of the lengths of all anchor segments (for brevity we omit the procedure to obtain this list).

6 Evaluation of alignment results

We conducted two experiments upon the European Constitution corpus, consisting of 47 English texts and the respective Portuguese translations, extracted from pages available at `http://europa.eu/scadplus/constitution/`. The average size of the texts is 21KB (4012 tokens) and the largest text is 331KB (59071 tokens).

In the first experiment we used a lexicon with 62025 English-Portuguese pairs of terms that had been automatically extracted from another parallel corpus (European Legislation) using the method described in [1] and manually verified.

LEXIC-AL took on average 0.3 seconds to align each pair of texts on an Intel Core 2 Duo 2.66GHz. The average number of iterations was 3 and the maximum was 6. The largest pair of texts, 323KB (English) and 331KB (Portuguese), is responsible for both the maximum number of iterations and the longest time, 7.7 seconds.

There are two mainstream methodologies for evaluating the quality of an alignment: (a) comparing the alignment against a golden standard created by hand *apriori* (Melamed [13], Véronis and Langlais [18], Och and Ney [14] and Chiao et al [4]), and (b) manual verification of the correctness of an alignment (Bilbao et al [2]) *aposteriori*. Method (a) is suitable for comparing alignments of the same corpus produced by different programs. However, the creation of the golden standard is very expensive in terms of human resources. Furthermore, because the segmentation produced by our method has variable granularity that depends on the quantity and the nature of the entries present in the lexicon, the comparison of the alignment produced against the golden standard would be very difficult because the segmentations would not match. We opted by a *aposteriori* evaluation of the precision and recall of each segment using the formulas below (proposed by Veronis and Langlais):

$$Precision = \frac{\text{Number of correct words in Portuguese segment}}{\text{Total number of words in Portuguese segment}}$$

$$Recall = \frac{\text{Number of correct words in Portuguese segment}}{\text{Total number of words in Portuguese translation of English segment}}$$

Then we computed the average of precision and recall on all segments evaluated. Below we present five example segments and the respective precision and recall.

EN Seg.	PT Seg.	PT Translation	Precision	Recall
particular situations	situações especiais ,	situações especiais	2/3	2/2
at the time	em o momento em que	em o momento	3/5	3/3
, any	,	, alguns	1/1	1/2
separate		distinta	0	0
decision	decisão	decisão	1	1

We selected 2424 aligned segments from the largest pair of texts, 316KB (English) and 323KB (Portuguese), which we consider to be the most challenging pair of texts in this corpus due to their size and because it contains some passages that were not translated to Portuguese.

According to the measures above, LEXIC-AL obtains an average precision and recall of 78.69% and 79.17%.

A second experiment was done after adding to the lexicon mentioned above 4479 new entries, extracted from the European Constitution corpus (Aires et al [1]) using the alignment produced in the first experiment. With the augmented lexicon the alignment precision and recall raised to 84.45% and 84.55% on the same pair of texts.

Bilbao et al [2] report a maximum[1] alignment precision of 75.46% for the alignment method described by Ildefonso and Lopes [8]. Although the evaluation method is similar, the results of the two evaluations cannot be meaningfully compared because there are too many variables that may have an impact on the result — evaluations were conducted on different corpora, the alignment granularity was different, etc. Nevertheless, the precision of the alignments in these experiments allow us to say that LEXIC-AL can produce better alignments than those obtained by the method of Ildefonso and Lopes [8].

7 Conclusions and Future Work

The results obtained in the evaluation are encouraging and we feel motivated to continue improving the method. One problem that is not yet addressed is how to handle crossed correspondences.

The method is language independent and we plan to do experiments with other languages to compare the performance on several language pairs.

From the two experiments described in previous section we conclude that alignment quality can be improved by iterating on term translation extraction, validation of extracted term translations and realignment using the extended lexicon.

A comparative evaluation of the alignments produced by LEXIC-AL and GIZA++ (Och and Ney [14]) is being prepared.

References

1. José Aires, José Gabriel Pereira Lopes, and Luís Gomes. Phrase translation extraction from aligned parallel corpora using suffix arrays and related structures. In *Progress in Artificial Intelligence: 14th Portuguese Conference on AI, EPIA'09*, Lecture Notes in Artificial Intelligence. Springer-Verlag, October 2009.
2. Víctor Bilbao, Gabriel Pereira Lopes, and Tiago Ildefonso. Measuring the impact of cognates in parallel text alignment. In Amilcar Cardoso Carlos Bento and Gael Dias, editors, *2005: Portuguese Conference on Artificial Intelligence, Proceedings*, pages 338–343. IEEE Computer Society, 12 2005.

[1] the alignment method takes a treshold parameter that must be adjusted for each pair of languages; depending on the threshold parameter the alignment precision varies between 53.37% and 75.46%

3. F. Chen, Stanley. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 9–16, Morristown, NJ, USA, 1993. Association for Computational Linguistics.
4. Yun-Chuang Chiao, Olivier Kraif, Dominique Laurent, Thi Minh Huyen Nguyen, Nasredine Semmar, François Stuck, Jean Véronis, and Wajdi Zaghouani. Evaluation of multilingual text alignment systems: the ARCADE II project. In *5th international Conference on Language Resources and Evaluation - LREC'06*, Genoa/Italy, 05 2006.
5. Mark W. Davis, Ted E. Dunning, and William C. Ogden. Text alignment in the real world: improving alignments of noisy translations using common lexical features, string matching strategies and n-gram comparisons. In *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, pages 67–74, Dublin, Ireland, 1995. Morgan Kaufmann Publishers Inc.
6. Pascale Fung and Ward Church, Kenneth. K-vec: a new approach for aligning parallel texts. In *Proceedings of the 15th conference on Computational linguistics*, pages 1096–1102, Morristown, NJ, USA, 1994. Association for Computational Linguistics.
7. Pascale Fung and Kathleen Mckeown. Aligning noisy parallel corpora across language groups: Word pair feature matching by dynamic time warping. In *In Proceedings of the First Conference of the Association for Machine Translation in the Americas, 81–88*, pages 81–88, 1994.
8. Tiago Ildefonso and Gabriel Lopes. Longest sorted sequence algorithm for parallel text alignment. *Computer Aided Systems Theory –EUROCAST 2005*, pages 81–90, 2005.
9. Martin Kay and Martin Röscheisen. Text-translation alignment. *Computational Linguistics*, 19(1):121–142, 1993.
10. Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal on Computing*, 22(5):935–948, 1993.
11. Dan Melamed, I. Bitext maps and alignment via pattern recognition. *Comput. Linguist.*, 25(1):107–130, 1999.
12. I. Dan Melamed. A portable algorithm for mapping bitext correspondence. In *ACL-35: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 305–312, Morristown, NJ, USA, 1997. Association for Computational Linguistics.
13. I. Dan Melamed. Manual annotation of translational equivalence: The blinker project. Technical Report IRCS-98-07, Institute for Research in Cognitive Science, 1998.
14. Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.
15. António Ribeiro. *Parallel Texts Alignment for Extraction of Translation Equivalents.* PhD thesis, Universidade Nova de Lisboa, Lisboa, 2002.
16. António Ribeiro, Gael Dias, G. P. Lopes, and João Tiago Mexia. Cognates alignment. In Bente Maegaard, editor, *Proceedings of the Machine Translation Summit VIII (MT Summit VIII), Santiago de Compostela, Spain, September 18-22, 2001*, pages 287–292. European Association of Machine Translation, 09 2001.
17. M Simard, G Foster, and P Isabelle. Using cognates to align sentences in parallel corpora. In *In Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 67–81, 1992.

18. Jean Véronis and Philippe Langlais. Evaluation of parallel text alignment systems. In Jean Véronis, editor, *Parallel Text Processing: Alignment and Use of Translation Corpora*, volume 13 of *Text, Speech and Language Technology*, chapter 19, pages 369–388. Kluwer, Dordrecht, 2001.

Recognizing Polarity and Attitude of Words in Text

Laritza Hernández[1], A. López-Lopez[2] and José E. Medina[1],

[1] Advanced Technologies Application Center, La Habana, Cuba
{lhernandez,jmedina}@cenatav.co.cu
[2] Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México.
allopez@inaoep.mx

Abstract. In this work, we present a problem of Sentiment Classification in texts. Sentiment Classification is an Opinion Mining and Sentiment Analysis task that has opened several problems, including a significant variety of applications. We propose a strategy to distinguish words that convey evaluation of an item from the rest, as well as to classify the evaluation polarity (positive or negative). In addition, relying on Appraisal Theory, we intend to classify the evaluation words in affect, judgment and appreciation. Both, polarity and attitude are recognized using a corpus-based approach. We have the purpose of applying this task in Spanish texts; thereby, we have created a corpus of movie reviews in this language that was manually processed. In the experiments, we noticed that our strategy has a good performance, achieving 76.78% and 80.16% of accuracy classifying polarity and attitude, respectively.

Keywords: Opinion Mining, Sentiment Analysis, Sentiment Classification, Appraisal Theory.

1 Introduction

Sentiment Classification is an Opinion Mining and Sentiment Analysis task; these are novel research areas strongly related. Some initial works dates back to the late 1980's and early 1990's [19], [21]; however, today there is a lot of controversy about the boundaries between these two areas. Some authors have defined Opinion Mining as a task "in which text mining methods are used to find interesting and insightful correlations between writer's opinions" [1]. Whereas, Sentiment Analysis is conceived as Sentiment Classification, referring to the task of categorizing texts, or pieces of text, based on their subjectivity and orientation [18]. Others extend it to identify or classify appraisal targets, determining the source of an opinion in a text, and developing interactive and visual opinion mining methods [3].

In this paper, we focus on the Sentiment Classification task; we propose a strategy to determine whether a given word conveys the evaluation of an item, and recognizing the evaluation kind. There have been previous works trying to make a distinction of evaluation kind in text finer than single semantic orientation or polarity (e.g. positive, negative or neutral). Some authors attempt to discern kinds of emotions (affects) [7], [13]. Whereas others, relying on Appraisal Theory, seek out expressions of attitude (affect, judgment, and appreciation) [14], [18]. Our work is one in this latter line. Besides, we classify the polarity of word in positive, negative or no-polarity.

Appraisal Theory studies the evaluative use of language, and is divided in three subsystems. Attitude corresponds to the words that emit an evaluation or that invite to take it. Graduation considers the words that intensify, diminish, sharpen or blur the evaluation. Engagement corresponds to those words that indicate the posture that the issuer adopts with its statement. In this paper, we focus only on Attitude that is subdivided in affect (evaluation of sentiments or emotional states), judgment (evaluation of the human behavior), and appreciation (evaluation of objects, processes, or people when they are valued from an aesthetic viewpoint). Attitude, also, can be positive or negative.

According to the Appraisal Theory, there is an overlap among the affect, judgment, and appreciation categories, since affect is considered as the basic system of Attitude, whereas judgment and appreciation are derivations of this, manifesting institutionalized emotions [8].

This theory has been poorly studied in terms of Computational Linguistics and Natural Language Processing; we have only found research for English. Even from a linguistic viewpoint, this theory has not been researched enough for the Spanish language; we have only found three reported works [8], [9], [20]. Two of them are very interesting studies of Kaplan, which we rely on for this work. In Fig. 1 we show some of the categories of the Appraisal Theory and we provide some examples of Spanish words expressing Attitude.

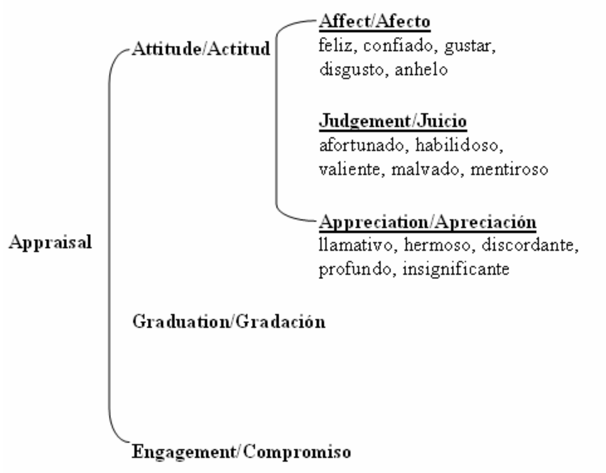


Fig. 1. Categories of Appraisal Theory and example words of Attitude system in Spanish.

Many words of an attitude system, according to the Appraisal Theory, have potential to express affect, judgment and appreciation when we consider them out of context. This situation has motivated us to use a corpus-based approach. This approach allows recognizing the evaluation of words considering the context where these tend to occur.

The problem of evaluative natural language processing has been poorly explored for Spanish [2], [4]. As a result, there is a lack of tools that limits the solutions of Sentiment Classification in this language. In this work, we are taking our first steps toward that goal.

Evaluative words can be found in different kinds of documents. One of the most analyzed types is the online reviews about products, such as movies, computers, phones and others. These are documents carrying out a free style of writing, where a great variety of evaluative expressions can be observed, as well as the three attitude kinds of the Appraisal Theory. But, we should mention that theses documents are not heavily loaded of judgment expressions. However, we can find other document kinds; e.g. editorials, in which a kind of more elaborated discourse can be appreciated. In this state of research, we have manually prepared a corpus, from movie reviews in Spanish taken from the website ciao.es. This is a very useful experimentation resource[1] as we will explain later.

The Sentiment Classification has achieved and opened a significant range of applications from monitoring and automatically summarizing user opinions about commercial products, people, organizations, and so on, up to the evaluation of public relations and marketing firms. For example, in sentence (1) we can notice two subjective words, both with a positive polarity. We could infer that the polarity of the sentence is positive as well, by computing the amount of positive words.

(1) "*Viéndola, me doy cuenta de que si tanto me ha* *gustado*$^{+}$ *es porque la trama es* *comprensible*$^{+}$*.*"

Recognizing attitude in the words (see sentence (2)) allows knowing the evaluation purpose of a sentence (sentiments, objects, or human behavior). Thus, sentences more relevant to a particular interest could be identified. For example, if the concerned item is the human behavior, you could be interested in what anyone says about the capacity, or moral integrity of a given person, more than in its physical appearance. This could be useful for tasks like opinion retrieval, opinion summarization, question answering focused to opinions, and others Opinion Mining tasks.

(2) "*Viéndola, me doy cuenta de que si tanto me ha* [**affect**: *gustado*$^{+}$] *es porque la trama es* [**appreciation:** *comprensible*$^{+}$]."

Thus, the contributions of this paper are to present a strategy for recognizing whether a given word has a polarity, positive or negative (already considered as expressing attitude) or whether it is a word with no polarity (regardless of attitude). Furthermore, according to the Appraisal Theory, if the word expresses attitude, we recognize the attitude kind; i.e., affect, judgment and appreciation. Thereby, we intend to capture the polarity of a word by other words that tend to occur in the same sentences. In a similar way, we try to capture the attitude classes of a given word, but considering the item evaluated in the sentences. On the other hand, we report our initial results toward the sentiment classification on Spanish texts.

[1]We would be glad to share this data set, if interested, contact us by e-mail.

We have structured the present work as follows. In Section 2, we briefly explain some related work to solve the *s*entiment classification problem and some of their drawbacks. In Section 3, we describe the proposed method. In the last section, we present the experimental results by using a textual corpus which we prepared from movie reviews in Spanish, selected from the website ciao.es.

2 Related Work

The detection and classification of subjective words in text using corpus-based approaches rely on syntactic patterns or co-occurrence of words. These approaches allow recognizing the polarity and attitude of the words determined by the context where they tend to appear. Nevertheless, it has the drawback of being dependent on the domain of the corpus used, and in consequence, usually tend to identify an insufficient set of words. Next, we describe some works in the context of our proposal.

Turney and Littman, in 2002 and 2003, proposed a strategy that intends to infer the "semantic orientation" or evaluative character of a word from extremely large corpora, considering its semantic association with other words, which he called "paradigms" [16], [17].

These authors consider that a word has a positive semantic orientation if it conveys the evaluation that the item is desirable; and a negative orientation if it conveys the evaluation that the item is undesirable. They determine the positive/negative semantic orientation of a given word *w* from the strength of its semantic association with the positive/negative paradigms. Then, they compute the semantic orientation of *w* as the difference between both positive and negative strengths. The words (good, nice, excellent, positive, fortunate, correct, and superior) and (bad, nasty, poor, negative, unfortunate, wrong, and inferior) were taken as positive and negative paradigms (*pp* and *np*), respectively.

Turney and Littman in a previous work, based on Pointwise Mutual Information (PMI), proposed a measure of word semantic association using information retrieval, called PMI-IR [15]. The measure PMI-IR intends to determine which “alternative”, given by a set of choices; i.e., {$\text{choice}_1,\ldots,\text{choice}_n$}, corresponds to a given word *w*, that they called “problem”. They use Church and Hanks’ PMI defined as a measure that estimates word association norms determined by word co-occurrences in a corpus as follows [6].

Let P(w) and P($choice_i$) be the probabilities of two words, w and $choice_i$ respectively; then PMI(w, $choice_i$) is the mutual information between w and $choice_i$ defined as:

$$PMI(w, choice_i) = \log_2\left(\frac{P(w, choice_i)}{P(w)P(choice_i)}\right). \quad \textbf{(1)}$$

PMI(w, $choice_i$) can be interpreted as the relation existing between the probability of w and $choice_i$ co-occurring in the same context, and the probability of w and $choice_i$ when they are statistically independent. Considering that Turney and Littman were looking for the maximum score, they proposed to drop log2, (because of its monotonicity) and P(w) (because "it has the same valued for all choices, for a given problem word"). Thus, (1) can be simplified to:

$$PMI(w, choice_i) = \frac{P(w, choice_i)}{P(choice_i)} . \quad \textbf{(2)}$$

On the other hand, these authors use four ways to calculate the probabilities in (2). But, we did not consider them because these probabilities were calculated as the number of returned matching documents from AltaVista Advance Search, by means of hits (query), and using the NEAR operator.

Turney and Littman method uses an extremely large corpus that does not require a manual process for its preparation, but this method depends on the variations and availability of an online search system. Besides, the NEAR operator considers that two words are close when they are halfway at least ten words, but it does not distinguish if the words are in the same sentence, an aspect that we consider to keep in mind.

In 2009, Brooke proposed the creation of semantic orientation Spanish dictionaries, making an analogy with adjective, noun, verb, adverb, and intensifiers dictionaries in English [6]. Each adjective, noun, verb, adverb dictionary in English is automatically translated to Spanish by means of the online bilingual dictionary Spanishdict and online Google translator, maintaining the semantic orientation of words from English. Also for the bilingual dictionary and translator, the author proposed other method using a textual corpus in Spanish formed by 400 reviews about hotels, movies, music, phones, washing machines, books, cars, and computers. From this corpus, adjectives, nouns, verbs, adverbs, and intensifiers dictionaries, with the semantic orientation for each word were extracted. In the comparison of the obtained dictionaries, Brook comments that the biggest agreement was in the adjective dictionary; but its semantic orientation agreement was the worst.

That is a valid approach to cope with the problem of sentiment classification in Spanish. But, since the words "subjective sense", as well as the intensity of this subjective, can be lost in the translation, we consider that a finest study has to be done where the proper variables of this language are taken into account, avoiding loss of generality, as far as possible.

On the other hand, Taboada & Grieve work, in 2004, used a similar strategy as that applied by Turney and Littman to classify adjectives relying on the Appraisal Linguistic Theory. This classification is used to calculate the degree to which a review (opinion texts about movies, books, cars, cookware, phones, hotels, music, and computers) expresses affect, judgment, or appreciation [14].

These authors improved Turney classification because it does not only determine whether an adjective is positive or negative, but the "adjective' overall evaluative potential", defined as the probability of using an adjective in evaluative discourses to express affect (AfP), judgment (JP), or appreciation (ApP). These are calculated as follows.

$$AfP = \frac{MI(I\ was, w)}{MI(I\ was, w) + MI(he\ was, w) + MI(it\ was, w)}, \quad \textbf{(3)}$$

$$JP = \frac{MI(he\,was, w)}{MI(I\,was, w) + MI(he\,was, w) + MI(it\,was, w)}, \quad (4)$$

$$ApP = \frac{MI(It\,was, w)}{MI(I\,was, w) + MI(he\,was, w) + MI(it\,was, w)}, \quad (5)$$

$$MI(PRO\,was,\, A) = \log_2 \left(\frac{hits(PRO\,was\,w)}{hits(PRO\,was)hits(w)} \right), \quad (6)$$

where MI is the mutual information between a word (adjective) w and a "pronoun-copula" pair (PRO) estimated from hits(query) on AltaVista.

This is a first interesting approach to classify attitude using context; but, there are several examples that show that the three proposed combinations (I was (affect), He was (judgment), It was (appreciation)) are not enough and they even fail.

Whitelaw, Garg and Argamon, in 2005, presented a method for sentiment classification that extracts and analyzes "adjectival appraisal groups" from texts, relying on the Appraisal Theory. Although they do not use a corpus-based approach, we consider important to comment their work [18].

These authors consider that an adjectival appraisal group, in English, is a coherent group of words that expresses together a particular attitude. It is formed by a head appraising adjective with an optional preceding list of appraisal modifiers (*very*, *sort of*, *not*, and other), each denoting a transformation of one or more appraisal attributes of the head. They also take into account the English language, easing the extraction of adjectival appraisal groups, i.e., they consider the word-order, inherent to this language, to remove all pre-modifiers of an appraising adjective. This word-order is not the same in Spanish, for instance.

They used semi-automated methods to build a lexicon of appraising adjectives and modifiers. They obtained, from word and phrase seeds taken from Martin [10] and Matthiessen [5] works and supported on WordNet and other thesaurus, expanded lists of "candidate" terms for each appraisal category that they considered (only the related word was included in the lists; i.e., synonyms, members of each synset and others, but with same part-of-speech as seed term). Then, the terms of each category obtained by this process were ranked by its occurrence frequency in the candidate lists. Later, a manual inspection was carried out to obtain the list of final terms, by removing less frequent ones from each category in order to reach a final set.

3 Proposed Strategy

As we known, many words in the human language are ambiguous (they do not convey a single message) when they are studied out of context; i.e., the context strongly determines the word sense. The evaluative language is not an exception (e.g. it is difficult to know if *big* or *much* conveys a negative or positive evaluation). On the other hand, according to proponents of the Appraisal Theory, some words out of context can be ambiguous according to their attitude class (e.g. *aburrido* (boring), *cómodo* (pleasant), or *agradable* (nice)). For this reason, we believe that we have to consider the contextual relations among the words in our proposal. This work, based on a corpus approach, tries to discriminate among attitude (words that convey an item evaluation, positive or negative) and no-attitude (words that do not convey an evaluation). Besides, relying on the Appraisal Theory, we intend to recognize the attitude of words. We describe a supervised strategy to learn sentiment classifiers of words. This is not a pioneer work in the machine learning using to solve sentiment classification problems; for example, Pang and Vaithyanathan (2002) and Mullen, N. Collier (2004) employed this technique for movies reviews classification [11], [12], and Wilson, Wiebe, and Hoffmann (2005) use it to classify sentiment expressions [22].

Next, we present the word classification divided in two parts. First, we present how to classify the polarity of words in positive, negative, and no-polarity. Second, we proposed how to classify attitude of words in affect, judgment, and appreciation. Then, we describe the corpus and the lexicon that we prepared as validation tools.

First, we assume that the sentences are the atomic units of coherent messages in texts. Therefore, we assume that the words that tend to co-occur in the same sentences are used with the intention to express similar or identical messages.

If one sentence conveys some appraisal, the following can happen:

a) There are words that indicate the appraisal in an explicit way.
b) There are no words that indicate the appraisal in an explicit way, but implicitly.
c) The polarity of appraisal is indicated with words of the opposed polarity. (Irony)

In this stage of our research, we are only interest in the first case; i.e., we only study the appraisal that is indicated in an explicit and direct way.

Then, considering the assumptions above, we also assume that words with a given polarity probably tend to occur in sentences of same polarity. That probably does not happen in sentences with different polarity or without polarity. Therefore, if we start forming a set of seed words and its polarity (positive, negative, and no-polarity), and if we represent them by a vector of words that occur in their sentence; then we assume that is possible to learn the context (words) of each polarity class and increasing our lexicon with new words of same polarity.

Subsequently, the attitude class of words is related to its sense and to the item (sentiment, human behavior, or object) target of evaluation. In this first stage of the research, we have assumed that in a single sentence, the evaluation of a single item prevails. Therefore, we start from the hypothesis that the words that tend to co-occur in the same sentence are being used with the intention of expressing the same kind of attitude.

According to our hypothesis, words are represented by a vector of dimension n, where n is the corpus size (sentences), assuming that sentences can be adequate to discriminate the attitude class of words. Given a word, the n-entry of the associated vector is 1 if the word is in the n-th sentence, and 0, otherwise.

Finally, taking into account the overlap of the classes inherent to Appraisal Theory, which we also found in polarity as well (see next section), we consider that some words might potentially be in more than one of these classes. Therefore, we do not treat the polarity classification neither attitude as a multi-classification problem. But, we provide a binary classifier for each polarity and attitude class. For example, for no-polarity, we take as positive examples all word vectors labeled with no-polarity, and as negative examples the remaining vectors labeled as positive and negative. Similarly, we proceed with positive and negative class and with the three attitude class as well.

3.1 The Corpus

We use as corpus a set of sentences selected from movie reviews in Spanish, gathered from the website ciao.es. This corpus was manually prepared, selecting from each review the sentences that were considered as containing words expressing some attitude class, denoted as “attitudinal sentences”. Furthermore, we added some sentences that did not contain that kind of words, denoted as “non attitudinal sentences”. To identify the words being used to express attitude, the Appraisal Theory for Spanish was taken into account [8], [9], [20]. The manual process relied on a set of tutorials and examples of this theory, as well as an annotation scheme that we prepared. Thus, we created a corpus of 1408 sentences, composed by attitudinal and non attitudinal sentences.

In addition, we compile all the words annotated manually with attitude in a list of 1247 terms. Then, taking into account the overlap of the classes, inherent to Appraisal Theory, we compile a list of words per class, being the appreciation the larger class and judgment the smallest one (788 and 287 words, respectively), and we calculated their percentages of overlap (see Fig. 2).

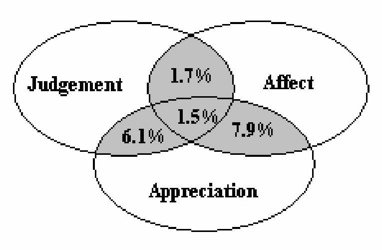


Fig. 2. Overlap percentages among attitude classes.

Table 1. Data collection statistics.

List	**Size**
Affect	352
Judgment	287
Appreciation	788
Positive	573
Negative	389
Non-attitude	178
Corpus	
No. sentences	1408
No. words	32, 920

Moreover, besides affect, judgment, and appreciation, some words from the 1247 terms manually annotated, were annotated in context as positive and negative, and we compiled two lists with 573 and 389 words, respectively, which had a 4% of overlap. Finally, we prepare a list with words that were not annotated, i.e. no-polarity words. We should remark that these words do not correspond with what has been defined as neutral polarity, which refers to words on which we can not decide about their polarity. In the Table 1, the size of the prepared collection is summarized.

4 Evaluation

To validate the proposed strategy in this work, we use as reference of "*good-classification*" the five lists of words manually classified, described in previous section. Thus, the obtained results are compared against the human judgment. Since the number of examples in the classes is unbalance, we used an over-sampling method called Smote (Synthetic minority over-sampling technique), that increases the proportion of the instances from the minority class.

Regarding the classifiers, we use the suite of Data Mining algorithms that Weka system[2] provides; namely, K*, Support Vector Machine (SVM), and Naive Bayes, NB. We selected these algorithms because we need to estimate the membership degree of the words in each class. We maintained default parameter values for each classifier, except for SVM that we set true the "buildLogisticModels" parameter to obtain the probabilistic version of this algorithm. To measure the performance of classification, we used 10-folds cross validation test method, and Precision, Recall, F-Measure and Accuracy percentage (see Tables 2 and 3). We can note that SVM and K* algorithms show the better results.

In the comparison of our results against human judgment, a good performance of attitude classification can be observed, resulting appreciation and positive the classes with worst results. Besides, we compare our proposal against the Turney & Litman results that we commented in the related work section. The other works in that section have not available results for the comparison, for this reason we do not refer to them here. We intend to use the same lists of words (657 positives and 679 negatives words from General Inquirer lexicon), but not all the words were in the English prepared corpus from The SFU Review Corpus[3] (only 114 positive and 53 negative words were recognized). The results are displayed in Table 4. We did not find reported results to compare the attitude classification.

Although our results are not directly comparable with the results of the selected method, since the corpora used in the evaluation are different, we can observe a good performance of the proposed method; even though these are not conclusive, as it is commented later on, in the future work.

---

[2] http://www.cs.waikato.ac.nz/ml/weka/

[3] http://www.sfu.ca/~mtaboada/research/SFU_Review_Corpus.html.

Table 2. Precision, Recall, F-Measure of SVM, K* and NB classifiers for polarity and attitude classification.

Polarity				**Attitude**		
Positive				**Affect**		
	Precision	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**
SVM	***0.51***	***0.615***	***0.558***	0.84	***0.832***	***0.836***
K*	0.505	0.575	0.538	***0.863***	0.808	0.835
NB	0.52	0.188	0.276	0.962	0.539	0.691
Negative				**Judgment**		
	Precision	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**
SVM	0.59	***0.779***	0.673	0.874	***0.866***	0.87
K*	***0.597***	***0.779***	***0.676***	***0.889***	0.856	***0.872***
NB	0.564	0.776	0.653	0.734	0.595	0.734
Non-attitude				**Appreciation**		
	Precision	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**
SVM	0.864	***0.837***	***0.85***	***0.572***	0.632	0.601
K*	***0.913***	0.72	***0.85***	0.553	***0.679***	***0.609***
NB	0.839	0.652	0.743	0.539	0.939	0.685

Table 3. Accuracy of SVM, K* and NB classifiers for polarity and attitude classification.

	Accuracy (%)		
Class	**SVM**	**K***	**NB**
Polarity	***76.78***	70.45	58.22
Attitude	***80.16***	79.88	64.87

Table 4. Accuracy of Hatzivassiloglou & McKeown, Turney & Littman -PMI and LSA, and our method for positive, negative, and no-attitude class.

Methods	**Corpus size (words)**	**Accuracy (%)**
Proposed method (SVM)	94 905	87.21
Proposed method (K*)	94 905	86.76
Turney & Littman -PMI	One-hundred-billion	82.84
Turney & Littman -PMI	Two billion	76.06
Turney & Littman -PMI	Ten million	61.26

5 Conclusions

In this paper, we showed our initial steps toward sentiment classification on Spanish texts, first recognizing word polarity (positive, negative, and no-polarity), and attitude (affect, judgment, and appreciation).

The results show a good performance of the proposed classification strategy when is compared against human judgments and early proposals, achieving 76.78% and 80.16% accuracy in polarity and attitude classification in Spanish (only considering

SVM algorithm). Also, we can note that when we compare our proposal with earlier results, promising results are observed. But, to reach a final conclusion, we have to test the reported method in the same or similar corpus, as far as possible.

In further works, we plan to prepare a new version of Spanish corpus, by increasing the size of sentences, and extending their domain. Besides, we will consider that more than one item could be evaluated inside a single sentence. We will work in classification of expressions (word sequences) rather than individual words, also tackling the problem with a multiclass approach, considering the overlap inherent to Appraisal Theory.

References

1. Argamon, S., Bloom, K., Esuli, A. and Sebastiani, F.: Automatically Determining Attitude Type and Force for Sentiment Analysis. *Procs. of the 3rd Language and Technology Conference* (LTC'07), pp. 369--373 (2007)
2. Balahur, A., Montoyo, A: Multilingual Feature–Driven Opinion Extraction and Summarization from Customer Reviews. *Procs. of the 13th international conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems*, Lecture Notes In Computer Science, vol. 5039, pp. 345--346 (2008)
3. Bloom, K., Garg, N.and Argamon, S.: Extracting Appraisal Expressions. *Procs. Of North American Chapter of the Association for Computational Linguistics - Human Language Technologies conference*, pp. 308--315 (2007)
4. Brooke, J.: *A semantic approach to automated text sentiment analysis*. Master Thesis. Simon Fraser University (2009)
5. Matthiessen, C.: *Lexico-grammatical cartography: English systems*. International Language Sciences Publishers (1995)
6. Church, K.W., & Hanks, P.: Word Association Norms, Mutual Information and Lexicography. Procs. of the 27th Annual Conference of the Association of Computational Linguistics New Brunswick, NJ: *Association for Computational Linguistics*, pp. 76--83 (1989)
7. Genereux, M., Santini, M.: Exploring the use of linguistic features in sentiment analysis. *Corpus Linguistics* 2007, Birmingham, UK, pp. 27-30
8. Kaplan, N.: Nuevos Desarrollos en el Estudio de la Evaluación en el Lenguaje: La Teoría de la Valoración. *Boletín de Lingüística*, Universidad Central de Venezuela, pp. 52--78 (2004)
9. Kaplan, N.: *La Construcción Discursiva del Evento Conflictivo en las Noticias por Televisión*. PhD Thesis Universidad Central de Venezuela, Facultad de Humanidades y Educacion (2007)
10. Martin, J. R., White, P.R.R.: *The Language of Evaluation: Appraisal in English*. Palgrave, London (2005)
11. Mullen, T., Collier, N.: Sentiment analysis using Support Vector Machines with diverse information sources. *Procs. of 2004 conf. on Empirical Methods in Natural Language* (2004)
12. Pang, B., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. *Procs. of 2002 conf. on Empirical Methods in Natural Language* (2002)
13. Rubin, V.L., Stanton, J.M., Liddy, E.D.: Discerning Emotions in Texts. *Procs. of AAAI Spring Symposium on Exploring Attitude and Affect in Text* (2004)
14. Taboada, M., Grieve, J.: Analyzing appraisal automatically. In *Procs. of AAAI Spring Symposium on Exploring Attitude and Affect in Text*, pp. 158--161 (2004)

15. Turney, P.D.: Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Procs. of the Twelfth European Conference on Machine Learning*, pp. 491--502 (2001)
16. Turney, P.D., Littman, M.: Unsupervised learning of semantic orientation from a hundred-billion-word corpus. Technical Report ERC-1094 (NRC 44929), National Research Council of Canada (2002)
17. Turney, P.D., Littman, M.L.: Measuring praise and criticism: Inference of semantic orientation from association, ACM Transactions on Information Systems (TOIS), vol. 21, no. 4, pp. 315--346 (2003)
18. Whitelaw, C., Navendu, G., Argamon, S.: Using Appraisal Groups for Sentiment Analysis. Procs. of the 14th ACM International Conference on Information and Knowledge Management, pp. 625--631 (2005)
19. Wiebe, J., Rapaport, W.: A Computational Theory of Perspective and Reference in Narrative. Procs. of the Association for Computational Linguistics, pp. 131--138 (1988)
20. White, P.R.R: Un recorrido por la teoría de la valoración. (Translated by Elsa Ghio). In The Appraisal Website: Homepage. http://www.grammati cs.com/appraisal /. [consulted:30/02/09] (up-to-date 2005)
21. Wiebe, J.: Recognizing Subjective Sentences: A Computational Investigation of Narrative Text. PhD Thesis. State Univ. of New York at Buffalo (1990)
22. Wilson, T., Wiebe, J., Hoffmann, P: Recognizing contextual polarity in phrase-level sentiment analysis". In Procs. of 200 conf. on Empirical Methods in Natural Language (2005)

Standardisation of Hotel Descriptions

Nuno Miranda[1] Ricardo Raminhos[1] Pedro Seabra[1]
José Saias[2] Teresa Gonçalves[2] Paulo Quaresma[2]

[1] Viatecla SA, Almada, Portugal
{nmiranda,rraminhos,pseabra}@viatecla.com

[2] Dep. Informática – Universidade de Évora, Évora, Portugal
{jsaias,tcg,pq}@di.uevora.pt

Abstract. The description of tourism products (hotel, aviation, rent-a-car and holiday packages) is strongly supported by natural language expressions. Due to tourism dynamics and the extent of its offers, manual data management is not a reliable nor scalable solution: descriptions are structured in different ways, possibly comprising different languages, complementing and/or overlapping one another. This paper presents a prototype that automatically classifies and extracts relevant knowledge from real operational hotel descriptions retrieved from the KEYforTravel tourism application framework. Captured knowledge is represented in a normalised format enabling the development of new business services.

1 Introduction

The Laboratory of Excellence .NET in Évora is the result of a protocol between ViaTecla, University of Évora and Microsoft, to introduce the experience and academic knowledge in a business context. In this sense, the Laboratory is a multi-disciplinary space with the presence of teachers, researchers and students (bachelor, master and doctorate). *Standards for Tourism Product Descriptions* is its first project, aiming for the automatic extraction of relevant features from the tourism products currently residing on KEYforTravel [7] (K4T) application.

The K4T is a tourism application framework developed by ViaTecla that provides a rapid and effective response, through the interconnection of the various participants of the tourism industry, assuring not only the view and exchange of information between them, but also the various areas of the product selling process. Since K4T gathers tourism information and products from heterogeneous sources, it is crucial to offer a standardised way of presenting its offers. The *Standards for Tourism Product Descriptions* project addresses this challenge.

One specific problem deals with hotel descriptions. Tourism operators have the hard and time-consuming task of manually carrying out the survey of hotel characteristics: they read the description provided by the hotel and insert its characteristics (equipment, services and location) into a database or a Web template. With the increasing globalisation of the tourism market, this situation becomes impractical since a single tour operator may offer thousands of hotels. The project's first practical application aims at processing that information by automatically extracting useful information and standardising the hotel description.

This paper is organised as follows: Section 2 describes system's architecture, Section 3 evaluates it and Section 4 points out conclusions and future work.

2 System's Architecture

The system aims to receive an hotel description and produce a standardised version of it. It was designed using a divide and conquer strategy where several small tools that focus on specific and simpler problems were interconnected. There are four main tools: a *Sentence Classifier*, an *Entity Extractor*, an *Ontology Instantiator* and an *Ontological Translator*, that where packed into a Web Service for the system to be available online. This architecture is depicted on Figure 1.

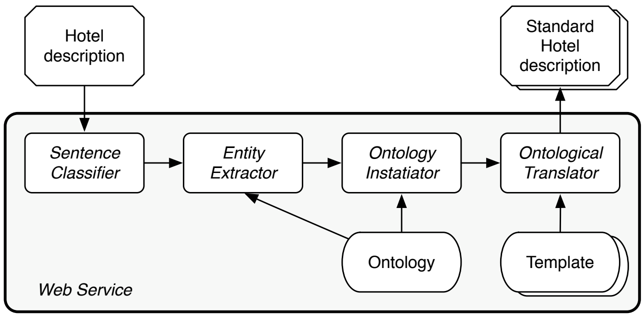


Fig. 1. System architecture diagram.

Sentence Classifier. This prototype's module is responsible for examining and classifying chunks of natural language text. It receives the crude description of the hotel and divides it into sentences. The sentences are then individually examined and automatically classified into a set of predefined classes such as "Equipment", "Service" and "Location". Each sentences can belong to more than one class, such as "Equipment" and "Service", provided it contains elements of both classes, or none of them, if it doesn't present any evidence. This classification aims at filtering the sentences by type to ensure a specific treatment to each one by the *Entity Extractor* module.

The *Sentence Classifier* was built using a Machine Learning approach. Since sentences can belong to more than one class, we have a multi-label classification problem. This kind of classification problem is typically solved by dividing it into a set of binary classification problems, where each concept is considered independently. Using a bag-of-words [6] representation for sentences, three different classification algorithms were applied and their decisions combined. The committee was comprised by a decision tree (C4.5 [5] algorithm), a Naïve Bayes classifier [3] and a support vector machine (SMO [4], the sequential minimal optimisation algorithm). This module's prototype was built using WEKA [9].

Entity Extractor. Having the sentences classified and grouped by type, the system tries to extract useful entities. This is the goal of this module that comprises two steps: finding useful entities and dealing with misspelled words.

Due to the fact that hotel descriptions are given in natural language without any pattern or consistency, the same entity can be described not only by a single term but by a set of synonyms, or even by an abbreviation. It can also be the case that the entity reference is misspelled or have failures in diacritics. This last hypothesis is very common since raw descriptions are frequently translated to different idioms and loose the regional diacritics.

To find useful entities on the description of each type of sentence a pattern matching approach was used. This is accomplished defining a set of more or less complex regular expressions able to identify synonyms, abbreviations and "almost" well-written words (e.g. `television`, `T.V.` and `TV` or `mini-bar` and `mini bar`) for common terms used for describing that kind of information.

To cope with misspelled words, the Levenshtein distance [2] is then used as a function to measure the similarity between the words not yet extracted and the ones that are considered relevant to the sentence's type.

Ontology Instantiator. To maintain the entities extracted from the system, and aiming to provide the basic structure and organisation of the involved concepts, an ontology for hotel domain was developed. We used Web Ontology Language [8] (OWL) that besides defining the structure also considers possible semantic relationships between objects and attributes. Each ontology object contains the set of regular expressions and Levenshtein functions used by the *Entity Extractor* module. In this way, the *Entity Extractor* becomes independent of specific problem at hands.

Using the developed ontology, this module generates an OWL instance that contains the entities with their attributes and their semantic relationships. This instance is then accessed using the Jena Semantic Web Framework [1].

Ontology Translator. This module is responsible for turning the extracted Entities attractive and easy to read by humans. Using a XML template that gives the skeleton for the final information representation, the *Ontological translator* fills it with the extracted instances. This template can later be replaced by another according to the preference of the tour operator or the target audience (e.g. corporate versus leisure clients).

Although in the present architecture ontology instances are the input for the *Ontology Translator*, this normalized knowledge (easily computable) can be applied to substantially expand and improve search capabilities in tourism offers since each Service, Equipment or Location item can be used in the query itself, or as a parameter in the search results refinement process. Further, since knowledge is formalized using an hierarchical structure, it may be applied to graphically map related items as well as structure navigation.

3 Evaluation

During the development of the project, and taking into account its future use, several tests were carried out using hotel descriptions residing on the Keyfor-Travel application. Figure 2 shows an example of running the system with an hotel description (in Portuguese) and three standard descriptions: a leisure template for English and a leisure and corporate templates for Portuguese.

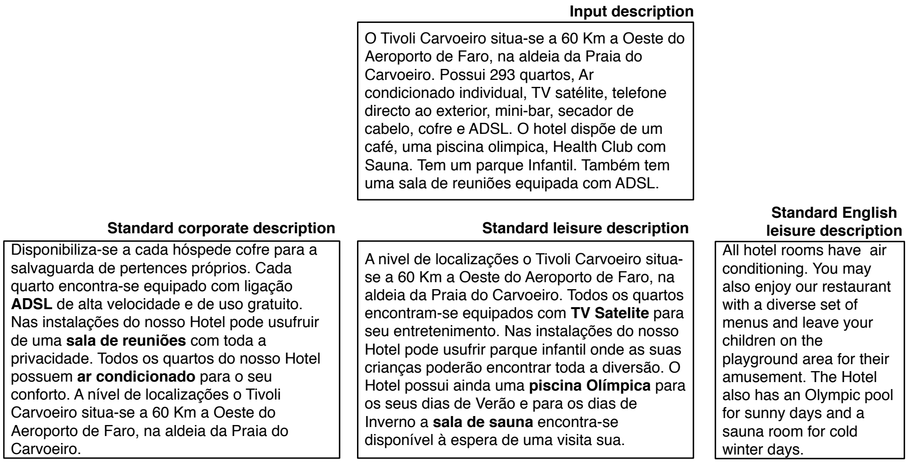


Fig. 2. Standard descriptions generated by the system.

4 Conclusions and Future Work

One can say that the project's main objectives were reached since it was possible to extract useful information, standardise it and create computable objects from plain text descriptions. Also, the used Machine Learning and string processing techniques revealed to be fully applicable to this domain. There is work to be done for normalising other tourism products and since there is room for improvements on its various modules, we hope to increase its overall performance.

In another context, one can also say that is possible to carry out joint projects between the University and business worlds and this is a good example of it.

References

1. HP Development. Jena – A Semantic Web Framework. http://jena.sourceforge.net.
2. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710, 1966. Originally publish in Russian.
3. A. McCallum and K. Nigam. A Comparison of Event Models for Naive Bayes Text Classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.
4. J. Platt. Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schölkopf, C. Burges, , and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 185–208. MIT Press, 1999.
5. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.
6. G. Salton, A. Wang, and C. Yang. A vector space model for information retrieval. *Journal of the American Society for Information Retrieval*, 18:613–620, 1975.
7. ViaTecla. KEYforTravel platform. http://www.keyfortravel.com.
8. W3C. OWL Web Ontology Language Guide. http://www.w3.org/TR/owl-guide.
9. I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, US, 2nd edition, 2005.

Relations Extracted from a Portuguese Dictionary: Results and First Evaluation

Hugo Gonçalo Oliveira[1], Diana Santos[2], Paulo Gomes[1]
hroliv@dei.uc.pt, diana.santos@sintef.no, pgomes@dei.uc.pt

[1] CISUC, University of Coimbra, Portugal
[2] Linguateca, SINTEF ICT, Oslo, Norway

Abstract. This paper presents PAPEL, a lexical resource for Portuguese, consisting of relations between terms, extracted by (semi) automatic means from a general language dictionary. An overview on the construction process is given, the included relations are presented and a quantitative vision is provided together with some examples. Synonymy relations were evaluated using a thesaurus as a golden standard and the other relations were rendered to natural language patterns and searched for in a corpus. The results of the evaluation are shown and discussed.

1 Introduction

In this paper, we present the current situation and the first automatic evaluation of PAPEL[3] - Palavras Associadas Porto Editora Linguateca, a set of relations between terms compiled into a lexical resource for Portuguese natural language processing (NLP). PAPEL [1] was constructed (semi) automatically by processing the definitions of a general dictionary of Portuguese [2], developed and owned by a large dictionary publisher, Porto Editora, and extracting relations denoted by textual patterns. The resulting relations were then validated, in the following way: synonymy relations were compared to the relations in a large thesaurus, while the other relations were rendered in natural language and searched for in a text corpora.

While for English, in the last decade, WordNet [3] was established as the standard model of a lexical database, the picture is quite different for other languages. There are attempts to create a similar database for Portuguese, namely Wordnet.BR [4], Wordnet.PT [5] and MultiWordnet.PT[4] but at the time of writing they were not freely available for download.

Those resources are however the product of time-consuming manual work, so we propose a semi-automatic construction of such a resource. Recent attempts on the automatic extraction of relations in Portuguese deal primarily with the hyponymy relation: Freitas and Quental [6] discuss its extraction from corpora while Costa and Seco [7] focus on user's search behaviour in a web search engine (logs). We are not aware of attempts to extract (semi) automatically semantic

[3] http://www.linguateca.pt/PAPEL
[4] http://mwnpt.di.fc.ul.pt

relations of other types from text written in Portuguese and compile them into one independent resource.

This paper starts by describing background work on knowledge extraction from machine readable dictionaries (MRDs) as well as overview methods for the evaluation of ontologies (Section 2). We then present the approach taken in the construction of PAPEL (Section 3), followed by a thorough description of its current contents, more precisely the relations included, their number and illustrative examples (Section 4). The evaluation attempted is finally described (Section 5), before discussing ideas for further work (Section 6).

2 Background

2.1 MRDs as a source of knowledge

The process of using MRDs in NLP started more than thirty years ago with early works of Calzolari [8], Amsler [9], and Chodorow et al. [10]. MRDs were analysed and, taking advantage of the simple structure of the definitions and of the restricted vocabulary used, procedures were developed to extract and organise lexical information. Following these ideas, Alshawi [11] proposed a specific grammar for parsing the definitions of a particular dictionary, based on the syntactic patterns used, and producing semantic structures. In the 1990s, as illustrated by Montemagni and Vanderwende [12], it became more common to use broad-coverage parsers to extract semantic information from dictionary text, claiming they were better suit to capture the distinguishing features in the definition, although this was not consensual in the community (see e.g. Hearst [13]).

In any case, one of the main reasons to use dictionaries and not (only) running text is because MRDs are the "authorities" on word sense [14]. Dictionaries have thus been exploited for several purposes, such as parsing or word sense disambiguation (WSD), but to our knowledge they have not been converted into an independent resource of its own before MindNet [15], that can therefore be claimed to be a kind of independent (dictionary-based) lexical ontology in a way that previous work was not. More recently, Nichols [16] and O‘Hara [17] also worked on the extraction of semantic relations from MRDs.

2.2 Evaluation of ontologies

When it comes to ontology evaluation, and although the information retrieval precision and recall measures are increasingly being used for evaluation in NLP (see e.g. Santos [18]), for semantic lexicons it is hard to have an independent golden standard for what should be there in the first place. The knowledge that should be represented is not clear. If we compare it with semantic data extracted from text, we have to remember that different interpretations and different meanings are often possible [19].

For domain ontologies, Brank et al. [20] divide evaluation approaches into four groups: performed by human subjects; comparison with a golden standard;

as for coverage, comparison with a collection of documents about a domain covered by the ontology; accomplishment of some task that uses the ontology.

Although the most reliable in the end, human evaluation does not take advantage of computer programs and relies heavily on time consuming work from at least one domain specialist. In order to make human evaluation easier, Navigli et al. [21] generated natural language descriptions of concepts, based on a grammar with distinct generation rules for each type of semantic relation.

Of course, the ontology can be compared with some other resource (e.g. another ontology) that is known to be correct, usually because it was created by specialists. But if this may be OK to validate a particular automatic method, it is obviously of little practical interest, because one expects to be creating new ontologies, not recreating existing ones. So, while the approach of compiling a human resource is commonly followed in joint evaluations, for example ReRelEM [22], which evaluated system's capabilities to recognise semantic relations between named entities, it can only encompass a few examples.

The third approach consists of finding how adequate a particular ontology is for representing the knowledge contained in a collection of documents, as in Brewster et al.'s [19] measurement of the fit between an ontology and a corpus. After identifying salient terms in a domain corpus and looking for them in an ontology for the same domain, the fit is proportional to the number of terms found in both corpus and ontology. The problem is that we cannot define a clear set of salient terms for general language, so this method cannot be applied to a lexical ontology that is supposed to describe the former.

The last approach, external or task-based evaluation, performs (indirect) evaluation by assessing the performance of an application which uses the ontology to do some task. Porzel and Malaka [23] proposed this approach aiming at evaluating ontologies with respect to the fit of the vocabulary, the fit of the taxonomy and the adequacy of non-taxonomic semantic relations.

These methods were used to evaluate domain ontologies, but if we consider dictionaries or lexical ontologies, evaluation is not a common practice, possibly because most of these resources are manually created by specialists. Ide and Verónis [24] are very critical of this fact and produced work for assessing the quality and usefulness of information extracted from MRDs. They concluded that the obtained the structures obtained by applying *Chodorow-like* procedures were incomplete and had several other problems but, if they merged the results extracted from several MRDs, the amount of problems decreased drastically.

Among the few attempts for evaluating information automatically extracted from MRDs, Richardson et al. [25] hand-checked a random sample of 250 semantic relations automatically extracted from a dictionary (and later included in MindNet), relying on common statistical techniques to estimate the representativeness of the accuracy for all the relations extracted. For MindNet [26], an (incomplete) evaluation of the quality of the semantic relations is mentioned, but its authors do not go very far in the description of the evaluation process. One comment made is that the quality varies according to the relation type. Likewise, to evaluate an ontology extracted automatically from a MRD, Nichols et al. [16] used Wordnet and GoiTaiKei [27] as a golden resource. In this process, they noticed that some relations were only in one of the two golden resources, which might indicate that they are both incomplete.

3 The approach for building PAPEL

The construction process followed four stages: (i) creation of the extraction grammars; (ii) extraction of the relations; (iii) manual result inspection and, finally, (iv) relations adjustment.

3.1 Extraction grammars

Inspired by Alshawi's work [11], specific grammars to parse the dictionary definitions were manually created. These grammars aim at the extraction of specifically predefined relations (described in Section 4) and are based on a previous empirical analysis of the structure of the definitions and of the vocabulary used. Table 1 shows some of the patterns that are included in the grammars together with the relations they are associated with.

Table 1. Examples of patterns used in the grammars.

Pattern	Associated relation
`tipo|género|classe|forma de`	Hypernymy
`parte|membro de`	Meronymy
`que causa|provoca|origina`	Causation
`usado|utilizado para`	Purpose
a word or an enumeration of words	Synonymy

```
PARTE{
  nome:nome * PARTE_DE:INCLUI;
  nome:adj * PARTE_DE_ALGO_COM_PROPRIEDADE:PROPRIEDADE_DE_ALGO_QUE_INCLUI;
  adj:nome * PROPRIEDADE_DE_ALGO_PARTE_DE:INCLUI_ALGO_COM_PROPRIEDADE;
}
```

Fig. 1. Examples of the description of the meronymy relations group.

Each grammar is made to process definitions of words belonging to only one of the four open grammatical categories (nouns, verbs, adjectives, adverbs). The relations to be extracted are defined according to their group, their name, the name of their inverse relation and, since cross-categorical relations can also be extracted, the grammatical category of its arguments (see the example in Figure 1). Most of the relation types have a grammar for both the type defined as direct and the inverse type.

3.2 Relation extraction proper

In the extraction stage, a chart parser uses the grammars to process the dictionary definitions. Every time a definition suits the grammar rules, the parser generates a derivation tree. Although it is possible to get more than one derivation for the same grammar/definition pair, currently only the derivation with less unidentified nodes is chosen. The same derivation tree can be used to extract several relations, provided they were identified by the grammar (see the example in Figure 2).

```
cometa, s. m. - astro geralmente constitudo
por núcleo, cabeleira e cauda
```

→ núcleo **PARTE_DE** cometa
→ cabeleira **PARTE_DE** cometa
→ cauda **PARTE_DE** cometa

```
[RAIZ]
   [QUALQUERCOISA]
     > [astro]
     [QUALQUERCOISA]
       > [geralmente]
   [PADRAO_CONSTITUIDO]
     [VERBO_PARTE_PP]
       > [constitudo]
     [PREP]
       > [por]
   [ENUM_PARTE]
     [PARTE_DE]
       > [nucleo]
     [VIRG]
       > [,]
     [ENUM_PARTE]
       [PARTE_DE]
         > [cabeleira]
       [CONJ]
         > [e]
       [PARTE_DE]
         > [cauda]
```

Fig. 2. Derivation for the definition of *cometa*.

3.3 Manual result inspection

The extraction results are inspected in order to identify systematic problems, and with the two previous steps form a loop that can be repeated at will.

Results from different extraction runs can be automatically compared to guarantee that newer results are better than older ones. After this procedure, we go back to the first stage, in which newer versions of the grammars are created, hopefully with some of the identified problems corrected.

3.4 Relations adjustment

After several loops of processing, the construction enters a new stage, where the relations with inadequate arguments (i.e. arguments whose grammatical category does not agree with the relation name) are either corrected or discarded. In order to simplify the relation set, all relations are translated into the type defined as direct. This stipulation was made based on what seemed more natural to the grammar writer, and not on frequency considerations. For example, *manga* INCLUI *punho* is translated to *punho* PARTE_DE *manga* and *dor* RESULTADO_DE *distensão* becomes *distensão* CAUSADOR_DE *dor*.

Given that the grammars have little grammatical information (introduced manually) and each dictionary entry only contains the grammatical category of the word being defined, in some cases we get relations with arguments that do not belong to the correct category. So, the grammatical category of each argument is verified, with the help of the grammatical information in the dictionary and, when the argument is not defined in the dictionary, with the help of the Jspell [28] morphological analyser. If the arguments of a relation are not adequate but there is a relation type that belongs to the same group and suits the categories of the arguments, the relation type is replaced, otherwise the relation is discarded. For example, the relation *loucura* ACCAO_QUE_CAUSA *desvario* becomes *loucura* CAUSADOR_DE *desvario*, because both arguments are nouns. During this verification, if an argument is not in the lemma form, it is changed to it, again with the help of Jspell.

4 A closer look at PAPEL

The set of relation types in PAPEL was chosen after reviewing the relations described in the literature and included in similar resources like WordNet [3] or MindNet [29]. We also took into consideration potential relation types that could be extracted from the most frequent patterns used in the definitions of the dictionary. The relations extracted are divided into eight main groups and have specific names according to the group and the grammatical category of the arguments.

After automatically correcting some of the relations based on the grammatical category of their arguments and removing duplicate relations we got the final set, comprising slightly more than 200,000 relations. Table 2 presents the specific types of relations, the grammatical categories of their attributes and quantifies each type of relation in PAPEL along with examples of extracted relations.

As it can be seen, synonymy and hypernymy are the most frequent relations, and we note that this set can still be further augmented if relations are combined and rules are applied in order to give rise to new implicit relations, as we intend to experiment later. This can be done in a similar fashion to what was done in the ReRelEM [22] task where, before comparing the golden collection and the participant run, they both had their relation set expanded with the application of inverse and transitive rules[5].

5 Evaluation of PAPEL

As we stated in Section 2, the evaluation of lexical ontologies is not that common. In order to evaluate PAPEL, we had a look at the approaches used to evaluate domain ontologies, discussed in the same section.

[5] These programs have been made publicly available by the HAREM [30] organisers.

Table 2. The relations of PAPEL.

Group	Name	Args.	Qnt.	Examples
Synonymy	SINONIMO_DE	*same*	80,432	(*flexível*, *moldável*)
Hypernymy	HIPERONIMO_DE	n,n	63,455	(*planta*, *salva*)
Meronymy	PARTE_DE	n,n	14,453	(*cauda*, *cometa*)
	PARTE_DE_ALGO_COM_PROP	n,adj	3,715	(*tampa*, *coberto*)
	PROP_DE_ALGO_PARTE_DE	adj,n	962	(*celular*, *célula*)
Cause	CAUSADOR_DE	n,n	1,125	(*fricção*, *assadura*)
	CAUSADOR_DE_ALGO_COM_PROP	n,adj	16	(*paixão*, *passional*)
	PROP_DE_ALGO_CAUSADOR_DE	adj,n	5,15	(*reactivo*, *reacção*)
	ACCAO_QUE_CAUSA	v,n	6,424	(*limpar*, *purgação*)
	CAUSADOR_DA_ACCAO	n,v	39	(*gases*, *fumigar*)
Producer	PRODUTOR_DE	n,n	932	(*romãzeira*, *romã*)
	PRODUTOR_DE_ALGO_COM_PROP	n,adj	31	(*sublimação*, *sublimado*)
	PROP_DE_ALGO_PRODUTOR_DE	adj,n	348	(*fotógeno*, *luz*)
Purpose	FINALIDADE_DE	n,n	2,095	(*defesa*, *armadura*)
	FINALIDADE_DE_ALGO_COM_PROP	n,adj	23	(*reprodução*, *reprodutor*)
	ACCAO_FINALIDADE_DE	v,n	5,640	(*fazer_rir*, *comédia*)
	ACC_FINALIDADE_DE_ALGO_COM_PROP	v,adj	255	(*corrigir*, *correccional*)
	MANEIRA_POR_MEIO_DE	adv,n	1,433	(*timidamente*, *timidez*)
Place	LOCAL_ORIGEM_DE	n,n	768	(*Japão*, *japonês*)
Property	PROP_DE_ALGO_REFERENTE_A	adj,n	3,700	(*dinâmico*, *movimento*)
	PROP_DO_QUE	adj,v	17,028	(*familiar*, *ser_conhecido*)

We had so far not the time to do human evaluation, nor access to similar resources. Fortunately, TeP (Thesaurus Eletrônico para o Português do Brasil) [31] is not only available through a web interface[6], but its knowledge base can be fully downloaded, so we used it as the golden standard for the evaluation of synonymy relations. Of course we are fully aware of lexical differences between the two varieties of Portuguese [32], but we believe that the common core is still and by far largest.

When it comes to the other relations, we decided to follow another validation approach, which can be presented as a combination of task-based evaluation and text corpora validation. We developed a "deconstruction" procedure based on grammars for translating the relations into natural language patterns, which were then searched for in a corpus, in order to identify whether the corpus lent the relations some support.

5.1 Evaluation of synonymy

TeP is an electronic thesaurus manually created for Brazilian Portuguese, comprising 19,888 synsets and 44,678 lexical units. Similarly to WordNet, each synset is a list of words that can have the same meaning.

The evaluation process should compare our set of synonymy relations with the relations implicitly defined by every synset. But since there were terms in PAPEL that did not appear in TeP and vice-versa, we first discarded all relations with at least one argument not either in TeP or in PAPEL and were left with about 68% of PAPEL and 35% of the 405,026 TeP relations[7] that we used for

[6] http://www.nilc.icmc.usp.br/tep2/

[7] To convert Tep, all elements of a synset were considered to be involved in a synonymy relation with all other elements of the same synset.

validation. After comparing both sets, 50% of the relations of PAPEL were found in TeP and 39% of the relations in TeP were in PAPEL.

As we said back in Section 4, the relations of PAPEL include only relations that were found in the dictionary and have not been the target of any kind of combination to infer new relations. If they were, we would have more relations to submit to the evaluation process. So, we applied the transitivity rule to our set: each pair of relations was combined and every time a pair had one common argument and a different one, a new synonymy relation was inferred (A SINONIMO_DE B $\wedge$ B SINONIMO_DE C $\rightarrow$ A SINONIMO_DE C).

After applying the transitivity rule, our 80,432 synonymy relations became 689,073. It was applied only once, otherwise the set would be much larger but would also have more inconsistencies than it already had after the first expansion. This happens because in PAPEL the key structure is the term and we do not handle polysemy or even homonymy, which means that two homographs will be treated as the same word. Transitivity in these conditions can give rise to clearly incorrect relations, such as: *queda* SINONIMO_DE *ruína* $\wedge$ *queda* SINONIMO_DE *habilidade* $\rightarrow$ *ruína* SINONIMO_DE *habilidade*. After comparing the expanded set with TeP, as expected the number of correct cases attested in TeP dropped to just 14% but, on the other hand, almost all synonyms in TeP were attested in our resource: 90%.

5.2 Validation of other relations between nouns

To assess the correctness of the non-synonymy relations we searched for natural language renderings of those relations in a corpus. This procedure was partly inspired by Etzioni et al. [33], who search for hypernymy patterns in the web to evaluate if a named entity is an instance of a specific class.

For this purpose we used CETEMPúblico, an annotated corpus provided by Linguateca with text from the Público newspaper published between 1991 and 1998, amounting to approximately 180 million words [34]. Despite being available for download, we used the AC/DC project [35] interface to query the corpus.

Although we started trying out queries in order to check all relations in our resource, we soon realised that for some of the relations it would be extremely improbable to find them in any (naturally occurring) text. For example, it is unlikely to find patterns to validate most of our cross-categorical relations, which seem to be precisely characteristic of the "dictionary genre": *liquidar* ACCAO_QUE_CAUSA *liquidação*, *fósforo* PARTE_DE_ALGO_COM_PROPRIEDADE *fosforoso*. It is not likely to find both arguments of each one of these relations in the same sentence.

As consequence, in the validation procedure followed, we dealt only with the noun to noun relations, and also with the cases attested in the corpus. That is, before starting the validation, all the relations including at least one argument that is not present in CETEMPúblico were discarded, by using the frequency lists of words and lemmas also provided by Linguateca.

We also had to select two random samples of the two most represented relations in our relation set, because they were just too many to validate in a short time: so a random sample of 3,145 hypernymy relations (8%) and of 2,343 meronymy relations (63%) were selected. For the remaining relations, whose results are shown in Table 3, we used the complete noun to noun relation sets.

Table 3. Results of the validation of non-synonymy noun to noun relations.

Relation	Relations w/ args in CETEMPúblico	%	Sample	%	Hits	%
Hypernymy	40,079	63%	3,145	8%	560	18%
Meronymy	3,746	35%	2,343	63%	521	22%
Causation	557	50%	557	100%	20	4%
Producer	414	44%	414	100%	12	3%
Purpose	1,718	59%	1,718	100%	173	10%

As one can see, around 20% of the hypernymy and meronymy relations seem to be supported by the text in the corpus. When it comes to other relations, the percentage of hits is smaller. We believe anyway that the evidence is good since it is obvious that a 200 million-word corpus of a newspaper genre has to contain much less general knowledge that a general dictionary. Besides, we only used a small set of patterns – often similar to the ones used in the extraction grammars – while there is a huge amount of possibilities to represent each relation in unrestricted text, full of modifiers and anaphoric references, contrary to simple and structured dictionary definitions.

To give a more concrete feeling about the validation results, Table 4 contains some correct relations that were supported by the corpus, but also relations that seem to be supported but are incorrect. On the other hand, there were correct relations that were not found in CETEMPúblico, for instance: *fruto* HIPERONIMO_DE *alperce*, *algoritmia* PARTE_DE *matemática*, *ausência* CAUSADOR_DE *saudade* or *aquecimento* FINALIDADE_DE *salamandra*.

6 Conclusions and further work

A new publicly-available lexical resource for Portuguese was presented in detail in this paper along with some first attempt to evaluate it. We expect it to be useful for all kinds of NLP applications, from automatic generation of text to intelligent search and writing aids, as well as to more theoretical studies of the semantics of the Portuguese language.

Although the evaluation results are still preliminary, we intend to use other sources of information. For instance, validate the relations by searching for indicative sentences in the whole web, through general search engines. However we will have to deal with even more patterns for expressing the relations because these engines do not have lemma or other linguistic annotations. We also plan to use other corpus resources, for example those already annotated with synonyms.

It should be emphasised that we do not see this resource as a final one, but as an important seed for further enrichment by the whole community. Not

Table 4. Results of the validation of non-synonymy noun to noun relations.

Relation	Correct	Support
língua HIPERONIMO_DE *italiano*	Yes	*As **línguas latinas, como o italiano** ou o português, tornam-se mais fáceis por causa das vogais.*
arbusto PARTE_DE *floresta*	Yes	*A **floresta é um conjunto de árvores, arbustos** e ervas de várias qualidades e tamanhos.*
cólera CAUSADOR_DE *diarreia*	Yes	*A **cólera provoca fortes diarreias** e vómitos e pode levar à desidratação e, consequentemente, à morte em poucas horas.*
oliveira PRODUTOR_DE *azeitona*	Yes	*Também a quantidade e tamanho das **azeitonas produzidas por uma oliveira** biológica é inferior, já que não são utilizados compostos de azoto que ajudam a planta a crescer.*
recrutamento FINALIDADE_DE *inspecção*	Yes	*Menos de metade dos jovens entre os 20 e os 22 anos apresentaram-se às **inspecções para recrutamento**, revelou o ministro da Defesa.*
músico PARTE_DE *música*	No	*... um espectáculo baseado na obra "Cantos de Maldoror", de Lautréamont, com **música composta pelo músico** inglês Steven Severin...*
fim FINALIDADE_DE *sempre*	No	*Sicília aponta **sempre para o fim** do dia, para o fim da luz.*

only we suppose there can be other sources for (semi) automatically enriching it – according to Ide and Verónis [24], the information in dictionaries is often inconsistent and incomplete, so, in order to minimize that, a lexical ontology should be the result of a merge of several sources – but we intend to go on devising ways to increase coverage, linguistic correctness and validation.

Acknowledgments

We would like to thank the R&D group of Porto Editora for making their dictionary available for this research. The project PAPEL is supported by the Linguateca, jointly funded by the Portuguese Government, the European Union (FEDER and FSE), under contract ref. POSC/339/1.3/C/NAC, and by UMIC and FCCN.

References

1. Gonçalo Oliveira, H., Gomes, P., Santos, D., Seco, N.: PAPEL: a dictionary-based lexical ontology for Portuguese. In Teixeira, A., de Lima, V.L.S., de Oliveira, L.C., Quaresma, P., eds.: Proc. Computational Processing of the Portuguese Language, 8th Intl. Conference (PROPOR). Volume Vol. 5190., Springer Verlag (2008) 31–40
2. : Dicionário PRO da Língua Portuguesa. Porto Editora, Porto (2005)
3. Fellbaum, C., ed.: WordNet: An Electronic Lexical Database (Language, Speech, and Communication). The MIT Press (May 1998)
4. Dias-da-Silva, B.C., Oliveira, M., Moraes, H.: Groundwork for the Development of the Brazilian Portuguese Wordnet. In Mamede, N., Ranchhod, E., eds.: Proc. Advances in Natural Language Processing: 3rd Intl. Conference. Lecture Notes in Artificial Intelligence, Berlin/Heidelberg, Springer (23-26 de Junho 2002) 189–196

5. Marrafa, P.: Portuguese wordnet: general architecture and internal semantic relations. Documentação de Estudos em Linguística Teórica e Aplicada (DELTA) **18** (2002) 131–146
6. Freitas, C., Quental, V.: Subsídios para a elaboração automática de taxonomias. In: XXVII Congresso da SBC - V Workshop em Tecnologia da Informação e da Linguagem Humana (TIL). (2007) 1585–1594
7. Costa, R.P., Seco, N.: Hyponymy extraction and web search behavior analysis based on query reformulation. In: Proc. 11th Ibero-American Conference on Artificial Intelligence. LNAI, Springer Verlag (2008) 332–341
8. Calzolari, N., Pecchia, L., Zampolli, A.: Working on the italian machine dictionary: a semantic approach. In: Proc. 5th conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1973) 49–52
9. Amsler, R.A.: A taxonomy for english nouns and verbs. In: Proc. 19th annual meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1981) 133–138
10. Chodorow, M.S., Byrd, R.J., Heidorn, G.E.: Extracting semantic hierarchies from a large on-line dictionary. In: Proc. 23rd annual meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1985) 299–304
11. Alshawi, H.: Analysing the dictionary definitions. Computational lexicography for natural language processing (1989) 153–169
12. Montemagni, S., Vanderwende, L.: Structural patterns vs. string patterns for extracting semantic information from dictionaries. In: Proc. 14th conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1992) 546–552
13. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proc. 14th conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1992) 539–545
14. Kilgarriff, A.: Ï don't believe in word senses. Computing and the Humanities **31**(2) (1997) 91–113
15. Richardson, S.D., Dolan, W.B., Vanderwende, L.: Mindnet: Acquiring and structuring semantic information from text. In: COLING-ACL. (1998) 1098–1102
16. Nichols, E., Bond, F., Flickinger, D.: Robust ontology acquisition from machine-readable dictionaries. In Kaelbling, L.P., Saffiotti, A., eds.: Proc. 19th Intl. Joint Conference on Artificial Intelligence (IJCAI), Professional Book Center (2005) 1111–1116
17. O'Hara, T.P.: Empirical Acquisition of Conceptual Distinctions via Dictionary Definitions. PhD thesis, NMSU CS (August 2005)
18. Santos, D.: Evaluation in natural language processing (6-17 August 2007)
19. Brewster, C., Alani, H., Dasmahapatra, S., Wilks, Y.: Data-driven ontology evaluation. In: Proc. Language Resources and Evaluation Conference (LREC), Lisbon, Portugal, European Language Resources Association (2004) 164–168
20. Brank, J., Grobelnik, M., Mladenic', D.: A survey of ontology evaluation techniques. In: Proc. Conference on Data Mining and Data Warehouses (SiKDD). (2005)
21. Navigli, R., Velardi, P., Cucchiarelli, A., Neri, F.: Quantitative and qualitative evaluation of the ontolearn ontology learning system. In: Proc. 20th Intl. conference on Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2004)

22. Freitas, C., Santos, D., Mota, C., Oliveira, H.G., Carvalho, P.: Detection of relations between named entities: report of a shared task. In: Semantic Evaluations: Recent Achievements and Future Directions (SEW), NAACL-HLT Workshop. (junho 2009)
23. Porzel, R., Malaka, R.: A task-based approach for ontology evaluation. In Buitelaar, P., Handschuh, S., Magnini, B., eds.: Proc. ECAI 2004 Workshop on Ontology Learning and Population, Valencia, Spain (2004)
24. Ide, N., Veronis, J.: Knowledge extraction from machine-readable dictionaries: An evaluation. In Steffens, P., ed.: Machine Translation and the Lexicon, LNAI, Springer-Verlag (1995)
25. Richardson, S., Vanderwende, L., Dolan, W.: Combining dictionary-based and example-based methods for natural language analysis. In: Proc. 5th Intl. Conference on Theoretical and Methodological Issues in Machine Translation, Kyoto, Japan (1993) 69–79
26. Vanderwende, L., Kacmarcik, G., Suzuki, H., Menezes, A.: Mindnet: An automatically-created lexical resource. In: HLT/EMNLP, The Association for Computational Linguistics (2005)
27. Ikehara, S., Miyazaki, M., Shirai, S., Yokoo, A., Nakaiwa, H., Ogura, K., Ooyama, Y., Hayashi, Y.: Goi-Taikei – A Japanese Lexicon. Iwanami Shoten, Tokyo, Japan (1997)
28. Simões, A.M., Almeida, J.: Jspell.pm – um módulo de análise morfológica para uso em processamento de linguagem natural. In: Actas do XVII Encontro da Associação Portuguesa de Linguística, Lisboa (2002) 485–495
29. Richardson, S.D., Dolan, W.B., Vanderwende, L.: Mindnet: acquiring and structuring semantic information from text. In: Proc. 17th Intl. Conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1998) 1098–1102
30. Santos, D., Freitas, C., Oliveira, H.G., Carvalho, P.: Second HAREM: new challenges and old wisdom. In et al., A.T., ed.: Computational Processing of the Portuguese Language, 8th International Conference, Proceedings (PROPOR 2008). Volume Vol. 5190., Springer Verlag (2008) 212–215
31. Maziero, E., Pardo, T., Di Felippo, A., Dias-da Silva, B.: A base de dados lexical e a interface web do tep 2.0 - thesaurus eletrônico para o português do brasil. In: VI Workshop em Tecnologia da Informação e da Linguagem Humana (TIL). (2008) 390–392
32. Wittmann, L., Pêgo, T., Santos, D.: Português do Brasil e de Portugal: alguns contrastes. In: Actas do XI Encontro Nacional da Associação Portuguesa de Linguística, Lisboa, APL/Colibri (1995) 465–487
33. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web: an experimental study. Artificial Intelligence **165**(1) (2005) 91–134
34. Santos, D., Rocha, P.: Evaluating CETEMPblico, a free resource for Portuguese. In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics. (9-11 July 2001) 442–449
35. Santos, D., Bick, E.: Providing Internet access to Portuguese corpora: the AC/DC project. In Gavrilidou, M., Carayannis, G., Markantonatou, S., Piperidis, S., Stainhauer, G., eds.: Proc. of the 2nd International Conference on Language Resources and Evaluation (LREC). (2000) 205–210

Multi-label Hierarchical Text Classification using the ACM Taxonomy

António Paulo Santos[1], Fátima Rodrigues[1]

[1]GECAD – Knowledge Engineering and Decision Support Group,
Institute of Engineering – Polytechnic of Porto, Portugal
{pgsa, mfc}@isep.ipp.pt

Abstract. Many of the works of text classification involve the attribution of each text a single class label from a predefined set of classes, usually small and flat organized (flat classification). However, there are more complex classification problems in which we can assign to each text more than one class (multi-label classification), that can be organized in a hierarchical structure (hierarchical classification) to support thematic searches by browsing topics of interests. In this paper, a problem of multi-label hierarchical text classification is presented. The experiment involves the creation of a multi-label hierarchical text collection, its pre-processing, followed by the application of different classifiers to the collection, and finally, the evaluation of the classifiers performance.

Keywords: multi-label hierarchical text classification; ACM taxonomy

1 Introduction

The classification of texts consists on the allocation of one or more previously existing categories to text documents, based on their content. More formally, considering a set of categories $C = (C_1, C_2, \ldots, C_{|C|})$ and a set of classified documents $D = (d_1, d_2, ..., d_{|D|})$, using a method or algorithm for learning, the intention is to build a classifier or a classification function which maps the documents into categories. The classifier is then used to classify new documents, not yet rated.

The multi-label hierarchical classification of documents is based on the task of assigning any number of classes, which are organized in a hierarchical structure, to text documents. In the literature there are many contributions about multi-label classification and also many about hierarchical classification. However, if we focus on the combination of these two problems, we find only a few contributions, based in AI techniques, with some limitations. Multi-label classification methods have been categorized into two different groups [16]: *problem transformation methods* and *algorithm adaptation methods*. The methods of the first group are algorithm independent. They transform the multi-label classification task into one or more single-label classification tasks. The methods of the second group extend specific learning algorithms in order to handle directly with multi-label classification. There are multi-label extensions of various algorithms, such as: decision tree [3], support vector machine [4], neural network [23] and k-nearest neighbour method [21].

In single-label classification, the forecast of a document class can only be right or wrong, since the document only belongs to one class. In multi-label classification, the forecast may be right, wrong or partially right, because, in the case documents belong to two or more classes, the projection can hit them all (classes), any of them or just a few of them. Several measures were proposed to evaluate the multi-label classifiers [24]. They are divided into *example-based* and *label-based* evaluation measures [17]. The first are calculated based on the average differences of the actual and projected set of labels over all test examples. The later decomposes the evaluation process into separate evaluations for each label, and then calculates the average of all labels. *Hamming Loss* and *Classification Accuracy* [14] are e*xample-based* evaluation measures. Any known measure for binary evaluation, such as accuracy, precision and recall, can be used as a *label-based* evaluation measure. The calculation of these measures for all labels can be achieved using two averaging operations, called macro-averaging and micro-averaging.

As a set of pre-defined classes we will use the ones defined in the latest ACM Classification Scheme, version 1998[1]. The defined classes are related to computer science and are organized hierarchically into four levels. The first level comprises 11 major partitions subjects codified A...K; these are subdivided into 81 second-level topics, which are further subdivided into third-layer topics. These are then subdivided in uncoded subtopics called subject descriptors. An important aspect to prepare a document for publication by ACM Press is to provide the information to index it according to the ACM classification system. At the moment, the authors, as specialists in the content of their publications, provide the categories and general terms. Instructions and guidelines for this procedure, besides its complexity and extensiveness (about 12 steps), are always dependent on the judgement of each author.

In this article, there are two main contributions: the creation of a truly hierarchical multi-label document collection, extract from the ACM library, and the development of a methodology for multi-label hierarchical text classification composed by various pre-processing techniques and a combination of various classification algorithms.

The paper is organized as follows. In section 2, the creation of the multi-label hierarchical text collection is reported, and also a simplified description of the ACM website extractor/collector system. Section 3 describes the pre-processing techniques used and the representation of texts chosen. Section 4 is about feature selection and the various documents collections created to test the classifiers. The next section is devoted to the creation of the multi-label classifiers. In Section 6, the results of the classifiers´ performance evaluation are presented, and in the last section, conclusions are drawn and some suggestions are made for future work.

---

[1] Available http://www.acm.org/class/1998/

2 Multi-label Hierarchical Text Collection

The most important element of a classifier is its training set. A training set is just a set of documents that exemplify the different classifications as fully as possible. If the training set is poor, the classifier cannot classify incoming documents correctly.

There are several sets of texts, including Reuters-21578[2], 20 Newsgroups[3], as well as biological data sets ENZYME [20], used and referenced in the literature for multi-label classification. However, these collections are not suitable for the study of multi-label hierarchical classification, because, besides not having an original hierarchical structure (Reuters-21578, Reuters-RCV1), they are not multi-label (20 Newsgroups) or are not even a text collection (ENZYME).

Due to the lack of a truly hierarchical and multi-label collection of texts, we developed a solution able to autonomously navigate through the ACM digital library, in order to collect Web pages and extract the relevant data to build a test collection. From the various types of scientific documents available (journals, magazines, proceedings, among others), we chose to collect the proceedings because this is the largest type of document represented in the ACM library and it covers a large share of the ACM categories involved, thereby exempting the collection of another type of document.

2.1 Architecture

Using the SAX2 API (Simple API for XML version 2) the ACM tree was extracted from a XML file and stored in a database[4]. Due to the large number of Web pages needed and the high volume of data necessary to extract, we designed and implemented a system capable of automatically browsing the relevant Web pages from the ACM website and extract the contents of interest to be stored in the created database. The designed system architecture was based on Google's [2] architecture. Figure 1 shows a simplified architecture of the system, where the Extractor, its key component, is represented. The system works as follows: (1) the Crawler is initially supplied with an address (URL) of the ACM digital library; (2) the Crawler delivers the Web Page to the Extractor; (3) The Extractor consults its set of rules and:

a. If the page has details of a scientific article, it extracts data such as title, keywords, abstract, primary and secondary classification, etc. and save data in the database;
b. If the page points to scientific articles, their links are extracted and saved in the database;
c. If the page is not any of the previous types, then it is not relevant and therefore nothing is done.

(4) The Crawler checks if there are web addresses in the database and if there are, goes back to point 2; otherwise, ends.

---

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[3] http://people.csail.mit.edu/jrennie/20Newsgroups/

[4] Available at http://www.dei.isep.ipp.pt/~i000313/

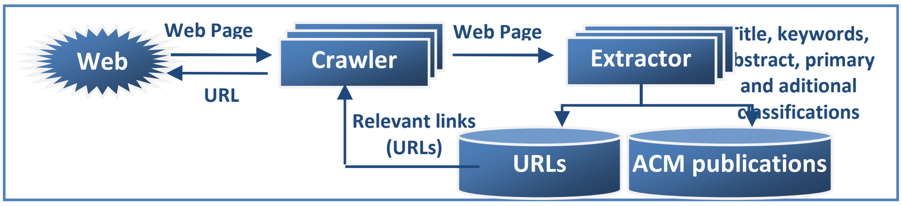


Fig.1. Simplified architecture of the collection and extraction system

The implemented process has the ability to find relevant Web pages and ignore the others. 112,000 Web pages were collected; of those 106,000 correspond to pages with information on scientific documents such as title, abstract, keywords, general terms, ACM classification, authors and the connection (link) to the full document in *pdf* format. The information of each document was obtained first with the identification of relevant Web pages, followed by the acquisition and retrieval of information. The information concerned by each publication was extracted from the web pages based on manually written rules. This was possible because the pages of the ACM site are reasonably structured which facilitates the writing of rules to automatically extract the information. From the documents collected in the ACM portal, it is possible to infer that a document may have between zero and seven primary classifications and between zero and thirty-six additional classifications - such a large number of classifications only occur in extreme cases. Considering the number of classifications, ignoring the distinction between primary and secondary classifications, the collected documents are distributed according to the number of classifications, as shown in figure 2.

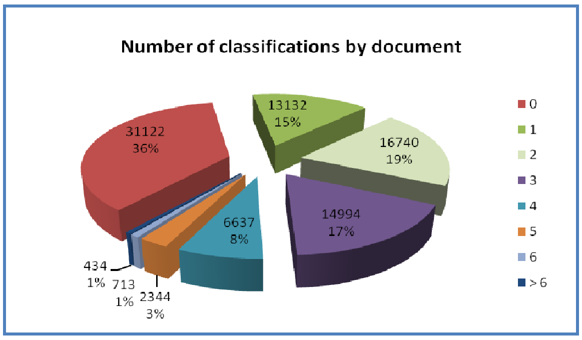


Fig. 2**.** Classification of documents collected in the ACM Portal

The distribution expresses the following: 31,122 documents do not have any classification, whereas only 13132 documents have one classification and the others have two or more classifications.

3 Pre-processing and Representation of Texts

After obtaining the document text collection, where each training document is composed by title, abstract, keywords and their classifications, it was necessary to preprocess it in order to get better performance from the classification algorithms. In this step, we used the API WVTool 1.1 (Word Vector Tool) [13]. The preprocessing tasks performed were:

- Term normalization, consisting of changing all characters to lowercase, removal of accents and treatment of equivalent terms (although written in different ways - for example, color/colour);
- Segmentation, division of text in single units (procedure known as tokenization), where all the characters (not letters) were recognized as separators, resulting units (tokens) with just letters;
- Stopwords removal, based on a standard English list of words, incorporated into the API;
- Reduction of the words to his radical, by the Stemming process, using the Porter Stemmer algorithm [11];
- Pruning (elimination) in the collection of words with frequency lower than 3. We have not done the elimination of words with frequency greater than a given value, because these words were deleted in the removal of stopwords or excluded in the selection of the most important terms (task reported in the next section).

After the documents pre-processing, each document was represented as a vector of terms to facilitate its manipulation. In this representation, each component of the vector represents a word and has an associated weight according to the TF×IDF (term frequency inverse document frequency) measure. This measure represents the number of times the term occurs in the document, normalized according to the total number of terms in the collection of documents. This measure gives a greater weight to terms that appear in the document more frequently but rarely in the collection of documents. The resulting vector for each document was normalized to Euclidean length.

4 Feature Selection

After the documents pre-processing phase, the number of terms originally in the text collection remained high, although it was considerably reduced. The high number of terms or features is typical of text classification problems. This high number of features is not desirable because it significantly increases the amount of time necessary for a classifier to learn. In fact, not all the terms used in text documents are relevant to describe them (and may even reduce the quality of the classifier´s learning). As a result, it is common to choose a subset made up of the most important terms. To assign a value which represents the importance of a term, there are several measures such as: information gain [12], mutual information, χ^2 (chi-square) and frequency [9]. We apply the measure information gain, since is one of the most effective measures [18].

4.1 Document Collection

As described before, each document in the ACM collection has several primary and secondary classifications (multi-label classification), and each classification is organized in a hierarchical structure of four layers. However, in our experiments we only considered the first two levels of the ACM hierarchy, that is, the documents are classified according to the classes A..K of the first level, and the topics of the second level. From the collection of documents gathered from ACM site, we have created two smaller collections of documents: one with 5000 documents and the other with 10,000 documents. Both are described on table 1.

Table 1. Documents Collection

	Multi-Label 5000	Multi-Label 10000
Number of documents	5000	10000
Total number of labels	11306	23786
Average number of labels per document	2,2612	2,3786
Maximum number of labels per document	14	19
Number of distinct terms after removing *Stopwords*	11743	16475
Number of distinct terms after removing *Stopwords* and pruning of terms with lenght < 3	4467	5697
Average number of documents by category in the 1st level (11 categories)	454,5	909
Average number of documents by category in the second level (81 categories)	61,7	123,4

Note: each document has one or more labels. In other words, a document belongs to one or more categories, and therefore contributes as a unit for one or more categories. Since each of the collections above has a high number of different terms, we created new collections of documents from each of them, each one with the 200, 400, 600, 800 and 1000 most important terms, selected according to the gain of information measure.

5 Multi-label hierarchical classification

5.1 Applied Algorithms

We apllied various plain multi-label classification algorithms: *Binary Relevance* (BR), *Label Powerset* (LP) [16] and *Multi-Label k-Nearest Neighbor* (MLkNN) [21]. The first two are *problem transformation methods*, while the last one is an *adaptation algorithm*. The first two methods were chosen because they correspond to the two most basic approaches to multi-label classification problems. The MLkNN was chosen as representative of the latest methods of *problem adaptation*. The applied methods are in the API Mulan[5] (Multi-label classification). The BR and LP algorithms were applied using the following basic classification algorithms:

– Sequential Minimal Optimization (SMO), to train a classifier based on support vector machines using polinomial kernels [10];
– IBK, which is the implementation of the Weka software of k-Nearest Neighbor Algorithm [1];
– Naive Bayes Multinomial [6];
– SVM (Support Vector Machine), using the library libSVM[6];
– J48, which is the implementation of the C4.5 algorithm [9] for decision trees learning (available in Weka).

With the exception of the SVM algorithm, all other algorithms are in the Weka[7] software. All experiments involving the IBK algorithm were performed with the k-value equal to 1 and 5. All experiments involving the MLkNN were performed with k equal to 1, 5, 10, and 30. The different values assigned to k, both in algorithms IBK and MLkNN, define the number of documents neighbors, for which the algorithm is based to make a decision. In all other algorithms, the parameters used were the default in their APIs.

The three multi-label classification algorithms applied are flat classification algorithms, so their direct application to this problem is not possible; therefore we adopted the local hierarchical approach described in the next section.

5.2 Methods to Handle Hierarchical Problems

5.2.1 Methods of Problem Transformation

A solution to handle the problem of hierarchical classification is to transform the hierarchical structure of categories in a flat structure and thus treat it as a problem of flat classification, for which there are known solutions and with good results [5] [19].

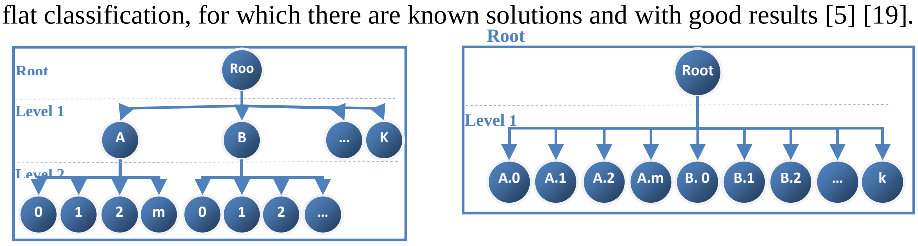


Fig. 3. Transformation of a hierarchical structure into a flat structure

This approach does not make use of the hierarchy of categories, resulting in the loss of this knowledge. Despite the results in [8] were very similar when dealing with

---

[5] Available http://mlkd.csd.auth.gr/multilabel.html
[6] Available http://www.csie.ntu.edu.tw/~cjlin/libsvm
[7] Available http://www.cs.waikato.ac.nz/ml/weka

the problem of classification in a hierarchical and flat shape, it is our opinion that such results were possible due to the reduced number of categories involved in the problem.

The results reported in [7] with a higher number of categories, support the use of a hierarchy between the categories. For this reason in this work we adopt the local hierarchical approach.

5.2.2 Local Hierarchical Approach

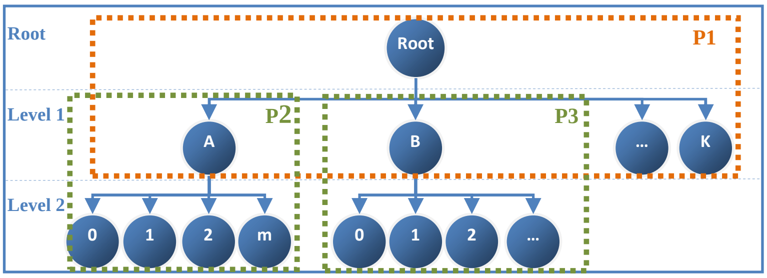


Fig. 4. Local hierarchical approach

In the local hierarchical approach the classification of a new document starts by building one or more classifiers in the root node (P1 problem solving), whose task is to indicate which categories A,B,C,...,K of the first level are relevant to the document. Assuming that A and B were chosen as relevant categories, it is now necessary to go down to the second level and build one or more classifiers, in each of these categories (separately solve the problems P2 and P3). Assuming that the classifier or classifiers responsible for selecting the children categories of node A indicate as relevant category 1, and the classifier or classifiers responsible for selecting the categories of child B indicate as relevant categories 0 and 2, they are then assigned the following ratings to the document: A → 1, B → 0, and B → 2. Note that the details of each category to which the document belongs are made indicating the path from the root, because if a document belongs to a subcategory, it also belongs to all their ancestors.

5.2.3 Global Hierarchical Approach

In the global hierarchical approach (or *big-bang*) [15], the class hierarchy is treated as a whole, and thus only a single classifier is responsible to discriminate all the classes [7]. This approach is similar to the transformation of a hierarchical structure into a flat structure approach, but it somehow considers the hierarchical relationships between classes. For this reason, the use of the flat classification methods in its original form is not possible, it is necessary to do some transformations in order to capture the relationships between classes.

The construction of a classifier using the global approach is more complex than following the local approach: it is computationally more complex and not flexible; for example, each time there is a change in the hierarchy of classes, the classifier needs to be trained again. The local hierarchical approach is identified as computationally more efficient than the global approach, but has a weakness in the spread of errors, that is, a wrong choice of a category in a given level of the hierarchy means that the error is propagated to all its descendants nodes.

6 Evaluation

The best known methods of evaluation of classifiers, such as the holdout method, k-fold cross-validation, leave-one-out and boostrap were designed to assess problems of plain classification. Since this problem is a hierarchical classification problem it was necessary to adapt the method k-fold cross-validation for each level of the hierarchy. As there is still no consensus or a clear trend on the evaluation measures to be applied to multi-label hierarchical classification problems, we chose to implement several measures based on examples.

Note that, for each classifier, all the learning steps (learning, classification and evaluation) were made together and not separately. The experiments were performed following the k-fold cross-validation method with k = 3 (where k is the number of subsets and the number of times that cross-validation process is repeated). Although this value is not as popular as the values k = 10, k = 5, shown to be the most appropriate, given the large number of experiments performed, and the size of text sets.

7 Results

With respect to the ACM tree, we only used the first and second level, because going down to the third level would result in categories with small numbers of documents. In both 5000 and 10,000 documents collections we have applied the various algorithms above refered, using the 200, 400, 600, 800 and 1000 most important terms in each collection, selected according to the information gain measure. Next, we only present the classification results at the second level of the tree, because only at this level the final document classification is obtained, see figures 5 and 6. Among all combinations of algorithms tested we only present the best results that match those obtained with *Binary Relevance* combined with *Naive Bayes Multinomial* (B.R. NB-M.); *Label Powerset* combined with *Sequential Minimal Optimization* (L.P.SMO) and *Multilabel k-Nearest Neighbor* (MLkNN).

The best results were obtained with the algorithms *Binary Relevance* combined with *Naive Bayes Multinomial*. It is clear that the results obtained with this combination of algorithms are independent of both the size of the collection or the number of terms used, that is, the results are similar in both collections, with 5000 and 10,000 documents and are better with smaller number documents terms, that is, 200 terms are sufficient to characterize this type of scientific documents collected from the ACM repository. In what concerns the other algorithms, the algorithm MLkNN gives better results than those obtained with the algorithm L.P.SMO. Regarding the documents collection, the MLkNN algorithm has the best results with the largest collection (10,000 documents) and with the largest number of terms (1000 terms). But the L.P.SMO algorithm has better results with the smallest collection (5000 documents) and also needs a large number of terms (1000).

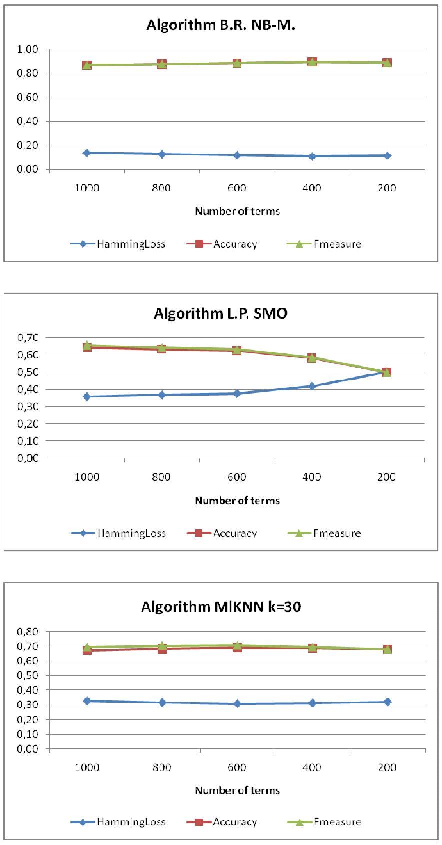


Fig. 5. Results of the different algorithms on dataset 5000, according to the number of terms

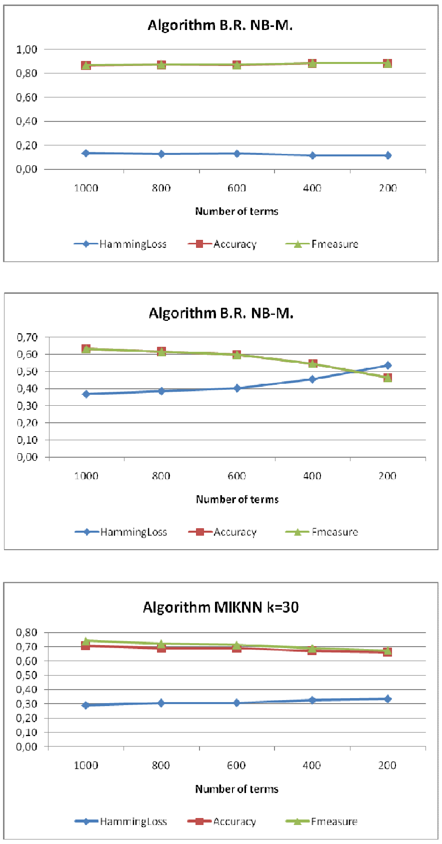


Fig. 6. Results of the different algorithms on dataset 10000, according to the number of terms

8 Conclusions and Future Work

Text classification has received the attention of researchers, since the mid 90´s. However, the multi-label hierarchical text classification still remains actual and offers exciting challenges, which give space for new research, and optimization of contributions already available. In this paper, we give an overview of hierarchical classification problems and their solutions. A multi-label hierarchical text collection classified according to the ACM scheme was created and pre-processed using various techniques. We have tested two problem transformation methods, *Binary Relevance* and *Label Powerset*, combined with various classification algorithms and also an adaptation method, *Multilabel k-Nearest Neighbor*. All of them were evaluated using two text collections with different number of terms. The best results were obtained with the combination *Binary Relevance* with *Naive Bayes Multinomial*. From the various experiences performed we can also conclude that this combination does not depend of both the size of the collection and the number of different terms used. We want to stress that these conclusions are only valid for the hierarchical multi-label document collection extracted from the ACM library.

In addition to the work done in this article, here are some proposals for future work:

- Investigate ways to recover from errors in a given level, that is, prevent the spread of errors to lower levels of the hierarchy, which is the main drawback of hierarchical local approach;
- Investigate measures for the evaluation of multi-label hierarchical classification that takes into account the distance in the tree provided between the predict class and the correct class;
- Conducting experiments using a hierarchical approach that combines at each level different classifiers using boosting, bagging or stacking.

Acknowledgments. We thank the Foundation for Science and Technology for the support expressed by funding the research project COPSRO - Computational Approach to Ontology Profiling of Scientific Research Organizations. The authors also thank all the support provided by GECAD.

References

1. Aha, D., Kibler D., Albert M.: Instance-based learning algorithms. *Machine Learning 6,* pp. 37--66 (1991)
2. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the seventh international conference on World Wide Web 7 Computer Networks*, pp. 107--117 (1998)
3. Clare, A., King, R.: Knowledge discovery in multi-label phenotype data. In *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 42--53. Freiburg: Germany (2001)
4. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 22--30 (2004)
5. Joachims, T.:Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning* pp.137--142 (1998)
6. John, George H., Langley P. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, pp.* 338–345 (1995)

7. Kiritchenko, Svetlana. *Hierarchical Text Categorization and Its Application to Bioinformatics*. Ph.D., Faculty of Engineering University of Ottawa, Canada, (2005)
8. Koller, D., Sahami, M.: Hierarchically classifying documents using very few words. In *Proceedings of 14th International Conference on Machine Learning ICML-97*, pp.170--178 (1997)
9. Mladenic, D., Grobelnik, M.: Feature selection for unbalanced class distribution and Naive Bayes. In *Proceedings of the 16th International Conference on Machine Learning*, San Francisco, CA: Morgan Kaufmann Publishers, pp.258--267 (1999)
10. Platt, J.: Fast training of support vector machines using sequential minimal optimization. *Scholkopf, B., Burges, C., Smola, A, eds: Advances in Kernel Methods - Support Vector Learning.* MIT Press (1998)
11. Porter, F.: An algorithm for suffix stripping. *Program*, 14 no. 3, pp. 130--137 (1980)
12. Quinlan, R.: *Constructing Decision Tree in C4.5: Programs for Machine Learning*. Morgan Kaufman Publisher,.pp. 17--26 (1993)
13. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylo, J.: Kernel-based learning of hierarchical multilabel classification models. In *Journal of Machine Learning Research*, (7) pp. 1601--1626 (2006)
14. Schapire, E., Yoram ,S.: BoosTexter: A boosting-based system for text categorization. *Machine Learning* (39) pp.135--168 (2000)
15. Sun, A., Ee-Peng, L.: Hierarchical Text Classification and Evaluation. In *Proceedings of IEEE International Conference on Data Mining (ICDM 2001)*, pp. 521--528. California, USA (2001)
16. Tsoumakas, G., Katakis, I.: "Multi-Label Classification: An Overview." *International Journal of Data Warehousing and Mining* 3(3):1--13 (2007)
17. Tsoumakas, G., Vlahavas I.: Random k-Labelsets: An Ensemble Method for Multilabel Classification. In *Proceedings of the 18th European Conference on Machine Learning*, pp. 406--417, Springer Verlag, LNAI 4701, Warsaw, Poland (2007)
18. Yang Y. and J. Pedersen: A comparative study on feature selection in text categorization. In J. D. H. Fisher, editor, *The Fourteenth International Conference on Machine Learning (ICML'97)*, pp. 412-420. Morgan Kaufmann (1997)
19. Yang, Y.: An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval* Vol.1, No.1/2 (1999)
20. Wurst, M.: The Word Vector Tool.User Guide/Operator Reference/Developer Tutorial (2007)
21. Zhang, L., Zhou., Z.: A k-nearest neighbor based algorithm for multi-label classifcation. In *Proceedings of the 1st IEEE International Conference on Granular Computing*, pp. 718--721 (2005)
22. Zhang, L.,Zhou Z.: Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7), pp. 2038--2048 (2007)
23. Zhang, M., Zhou., Z.: Multi-label neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering* 18, pp. 1338--1351 (2006)
24. Zheng, Z., Wu, X., Srihari, R.: Feature selection for text categorization on imbalanced data. *SIGKDD Explorations Newsletter*, 6(1), pp. 80--89 (2004)

The Design of OPTIMISM, an Opinion Mining System for Portuguese Politics

Mário J. Silva[1], Paula Carvalho[1], Luís Sarmento[2], Pedro Magalhães[3] and Eugénio Oliveira[2]

[1] University of Lisbon, Faculty of Sciences, LASIGE
{mjs, pcc}@di.fc.ul.pt

[2] University of Porto, Faculty of Engineering, DEI - LIACC
{las, eco}@fe.up.pt

[3] University of Lisbon, Instituto de Ciências Sociais
pedro.magalhaes@ics.ul.pt

Abstract. We present the design of OPTIMISM, an opinion mining system for detection and classification of opinions about relevant political actors, regarding a particular topic of debate. The system gathers opinion-rich texts from Portuguese social media, which are then classified according to their semantic orientation and intensity. Together with the main design decisions, we present the linguistic resources for sentiment analysis in Portuguese under development and the evaluation plan. Evaluation includes comparing opinion statistics produced by the system against poll data collected purposefully for this evaluation.

Keywords: Opinion mining; Sentiment analysis; Natural Language Processing; Crowdsourcing; Social Web

1 Introduction

With the increasing availability of user-generated contents (UGC), such as blogs, Internet forums and social networks, citizens have more opportunities to express their opinions and make them available to everyone. Publicly available opinions provide valuable information for decision-making processes based on a new collective intelligence paradigm designated as *crowdsourcing* [15]. Therefore, the computational treatment of sentiment and opinions has been viewed as a challenging area of research that can serve different purposes.

Existing opinion mining systems are generally designed to deal with specific text genres and topics (for example, movie and product reviews) in English. Since specialized reviews are topic delimited, research so far has mainly focused on identifying the overall sentiment expressed in self-contained reviews (e.g. movie reviews) or the sentiment about a possible set of features that are targets of opinion (e.g. product reviews). However, more complex UGC (e.g. blog entries, opinion articles) usually express both positive and negative opinions about an open set of possible entities and topics. Hence, it is important to determine in these cases not only whether a given document, paragraph or sentence conveys a positive or negative sentiment or opinion, but also which entities are mentioned in a positive or negative way, regarding a particular issue.

This paper presents the design of OPTIMISM, an opinion mining system for detection and classification, in quasi real-time, of opinions about relevant political actors, regarding a particular topic of debate. The system gathers opinion-rich texts from Portuguese social media, which are then classified according to their semantic orientation and intensity. These include weblog posts, comments to those posts, and comments to news published in mainstream media.

The main goal of the system is to detect trends in the electoral behaviour ahead of existing polling technology by mining the social media websites, investigating how these trends correlate with those obtained using conventional polling methods [7], [18]. Recently, Google researchers have demonstrated on their *flu trends* website how they could predict influenza epidemics two weeks ahead of the existing surveillance network through the analysis of query log data [10]. We intend to attain comparable results regarding trends on political entities.

In the next section we review related work. In Section 3 we address issues related to harvesting opinionated text. Section 4 presents the set of linguistic resources being developed for sentiment analysis in Portuguese. In Section 5, we summarize our approach for learning text classifiers, which is based on a reference corpus that we are building semi-automatically. In Section 6, we present an outline of the implementation and evaluation plans for the OPTIMISM prototype. We conclude this paper by highlighting the main features of OPTIMISM.

2 Related Work

Research on opinion mining has taken three main interrelated research lines (see the comprehensive survey by Pang & Lee [25]):

(i) Development of linguistic resources for sentiment analysis, such as lexica and manually annotated corpora;
(ii) Implementation of different algorithms for text analysis and classification according to their subjectivity and semantic orientation;
(iii) Extraction of opinions from text, possibly including different types of relations with associated content.

Opinionated text has been generally classified according to sentiment polarity and intensity. Work has focused on movie and product reviews (see, among others, [23], [24] and [20]), and, more recently, on electoral behaviour [17, 21, 26]. Sentiment classifiers have been implemented using mainly machine learning based approaches [25]. Latent Semantic Analysis and Semantic Orientation – Pointwise Mutual Inclusion have been successfully applied on the construction of sentiment lexical resources [27]. However, past research experiments show that the performance of classifiers depends on multiple linguistic factors, such as text genre, topic, and, especially, the coverage and precision of linguistic information [16, 28].

Most systems for sentiment analysis make use of sentiment lexicons, containing information about the predictable polarity of words, mainly evaluative adjectives. These have been particularly studied and used as features not only to detect subjective information, but also to classify the sentiment polarity in texts [9, 13].

Simpler approaches are based on the selection of relevant lexical features, and the verification of their occurrence in a given document, which has been globally classified according to the polarity of the prevailing features. In spite of their simplicity, such approaches have shown reasonable performance in specific types of texts, such as movie reviews. However, we believe that they do not provide satisfactory results when dealing with more complex texts, involving different topics and targets of opinion. Moreover, it must be stressed that the contextual polarity of a phrase or sentence containing a sentiment word may be different from the prior polarity assigned to that word in the lexicon [29].

For example, Carvalho has shown that the meaning (and polarity) of adjectives in Portuguese depends on multiple factors, such as their position in adnominal context (e.g. *Um homem pobre / A poor man* vs. *Um pobre homem / A miserable man*), the nature of the noun to which an adjective is related (e.g. *Uma pessoa importante / An important person* vs. *Uma quantia importante / A considerable amount*) and the type of auxiliary verb with which it co-occurs (e.g. *Ele é muito inteligente / He is very intelligent* vs. *Hoje, ele está muito inteligente / Today, he is very intelligent*) [4]. She also demonstrated that some evaluative adjectives can exhibit a particular behaviour when they are included in specific constructions, such as 'cross-constructions', where the adjective fills the head of a noun phrase (e.g. *O comunista do ministro / The communist of the minister*), and exclamative constructions expressing insult (e.g. *Seu atrasado mental / You moron!*).

Regarding the selection of contents from the web for text mining, available crawlers can be classified in four major classes determined by the used harvesting strategy: (i) broad crawlers, which collect the largest amount of information possible within a limited time interval [14]; (ii) incremental crawlers, which revisit previously fetched pages, looking for changes [8]; (iii) focused crawlers, which harvest information relevant to a specific topic aided by a classification algorithm for filtering out irrelevant contents [5]; (iv) deep crawlers that also collect information relevant to a topic, but, unlike focused crawlers, have the capacity of filling forms in web pages and collect the returned pages [22].

With the advent of web 2.0 sites, such as Blogspot.com and Facebook.com, the internet domain and language-based criteria that we successfully used in the past for delimiting the Portuguese Web no longer apply (see [11], for the former, and [19], for the latter). Instead of personal web pages, political actors are now changing how they present themselves on the web, relying on hosted pages at specific applications. Many web users create web avatars that do not have a home page with true or relevant information: they simply visit popular thematic forums on the web and comment on the topic of the moment under a common nickname on a regular basis.

The opinion mining prototypes to be released by the OPTIMISM will require quasi real-time feeding with relevant Portuguese texts conveying opinions about political entities as they are published on the web. The selection of contents for mining opinions will have to follow a radically different approach.

3 Text Harvesting for Crowdsourcing

One of the main difficulties faced while crawling the first generation of the web was that contents were largely unstructured, which made their parsing hard for detecting links. On the other hand, the Portuguese web was largely organized as a collection of linked websites, each hosted under its own sub-domain of the top-level .PT Internet domain.

The selection of contents for mining opinions will have to follow a radically different approach. In the OPTIMISM crawler, we simply collect most of the data we are interested in with existing tools, by subscribing the newsfeeds associated to the relevant political actors. Their contents and meta-data are already available as data streams in the Atom or RSS2.0 syndication formats (XML). Web sites like Rssmeme.com and Pipes.yahoo.com provide support for the creation of *mashups* that can be programmed to perform some of the necessary work of aggregating newsfeed data for our demonstrator. Many news websites and blogs also enable the download of the comments by their users tied to each news or blog post.

The new challenge is finding the semantic associations among these comments, made in hundreds or thousands of independently run discussions on the same topic, having some of the users participating in several of them simultaneously. In the social web, these associations are no longer established by HTML links, but through *hashes* (twitter inherited the concept from chat forums), nicknames, and *search keywords* (where the first *hit* indirectly points to the referenced web page).

In OPTIMISM, the configuration of the automatic discovery process mainly becomes an activity of identifying the appropriate keywords for locating the relevant contents, which requires linguistic and information science skills.

4 Linguistic Resources for Sentiment Analysis

In this section, we describe the linguistic resources being developed in the scope of the OPTIMISM project, namely a sentiment lexicon and a library of syntactic-semantic patterns where polarity-bearing predicates may occur.

4.1 Sentiment Lexicon

In previous work, we explored the syntactic and semantic information described in the linguistic resources developed by Carvalho [4]. These comprise 4,250 intransitive adjectives, characterized by occurring with human subjects and having no complements. The properties considered in such resources concern, among others, the constraints imposed by adjectives respecting (i) the type of auxiliary verb with which they can co-occur, (ii) their modification by a quantifier adverb or by a morpheme of degree, (iii) their presence in specific constructions, such as *characterizing indefinite constructions* (where the adjective appears after an indefinite article, in predicative context), *cross-constructions* (where the adjective fills the head of a noun phrase), and exclamative constructions expressing insult. Furthermore, these resources describe the predicative nouns morpho-syntactically associated with each adjective (e.g. *belo/beleza; beautiful/beauty*).

We have then selected the possible polar predicates from these resources, and classified each predicate according to its predictable polarity, which may be *0*, *1*, or *-1*. These codes represent, respectively, a neutral, positive or negative semantic orientation. We are not presently assessing the potential levels of intensity exhibited by predicates from the same polarity class (e.g. *feio/horrível; ugly/horrible*), although that may be considered in the future.

The sentiment lexicon has also been enriched with new entries, collected from diverse *corpora* on the web. At the present, it is composed by a total of 6,055 intransitive predicates (more precisely, 3,533 adjectives and 2,522 names). In terms of polarity frequency, 55.5% of the predicates are classified as negatives, 21.8% as positives and the remaining 22.7% as neutral.

Future developments will also include the enlargement of the sentiment lexicon, namely in what concerns predicate verbs and multiword expressions, by exploring bootstrapping approaches [2]. We will also take into consideration particular common and proper names that may convey polarity when used metaphorically (e.g. *Ele é um verdadeiro Hitler / He is a real Hitler*).

4.2 Ontology of Political Entities

Entity recognition for political opinion mining in social media has several particularities, which makes it different from more generic entity recognition tasks. We are initially considering a relatively small set of pre-defined entities, including politicians and some organizations (such as political parties). In this restricted scenario, most names are unambiguous, which may suggest that simple dictionary-based recognition can lead to acceptable results.

However, a significant proportion of political mentions are indirectly made, using paraphrastic constructions, ergonyms (e.g. *o primeiro-ministro / the prime-minister*, *a líder da oposição / the leader of the opposition, o candidato do PSD à CML / the candidate from PSD for CML*). Identifying constructions of this type and associating them with their corresponding political entities is a challenging task, because they admit different lexical and syntactic variations and are prone to ambiguity. Moreover, political actors and their roles in the political scene change quickly over time (e.g. a national parliament member may become a European parliament candidate or a city mayor). Additional challenges involve dealing with absence of capitalization, acronyms, metaphoric mentions and neologisms (e.g. "Pinócrates", which results from the amalgam of the names *Pinocchio* and *Sócrates*, for referring to the current Portuguese prime-minister *José Sócrates*).

To assist the opinion extraction task, we are developing an ontology of political entities that includes the names of the political actors and corresponding variants (neologisms, acronyms, etc.), and their roles in the political scene. This ontology, which initially covers the Portuguese political environment, is being compiled semi-automatically, by mining news items using simple patterns to find possible ergonyms (usually placed in apposition to the political entity), and conflating name variations using heuristics based on lexical inclusion and edit-distance criteria. For example, we found 23 distinct mentions for the politician *Paulo Rangel* ("líder da bancada do PSD", "cabeça-de-lista social-democrata ao Parlamento Europeu", "candidato do PSD às eleições europeias", among others). We have been mining newspaper RSS feeds for about a year, to collect the most common paraphrases (presently, for more than 1300 entities). Metaphoric mentions and neologisms commonly found in UGC will be manually added to ontology, at least in a first stage. We are in the process of evaluating the ontology. We plan to use this ontological knowledge in the entity recognition module, which scans texts to identify mentions of political entities by name or paraphrase (using both exact and soft matches with the ontology contents) and performs entity resolution.

4.3 Lexico-Syntactic Rules

Opinionated messages posted in blogs, internet forums and social networks are normally short, ungrammatical and incomprehensible when taken out of context. Moreover, opinions and sentiments are mostly expressed indirectly, making use of figurative language, such as metaphors. They thus represent a hard challenge for mining approaches relying exclusively on parsers. See, for example, the fragment of a comment to a news article published in the online newspaper *Público*, which was parsed by PALAVRAS [3].

durão [durão] **ADJ** M S @SUBJ>
fez [fazer] <fmc> **V** PS 3S IND VFIN @FMV
muito [muito] <quant> **DET** M S @<ACC @>N
bem [bem] <quant> **ADV** @<ADVL @>A
em [em] **PRP** @<ADVL
fugir [fugir] **V** INF @IMV **@#ICL-P<**
de [de] **PRP** @<ADVS
Portugal [Portugal] **PROP** M S @P<
e [e] <co-fmc> <co-inf> **KC** @CO
ir [ir] **V** INF @IMV
para [para] **PRP** @<ADVL
um [um] <arti> **DET** M S @>N
super [super] <*1> <n> **ADJ** M S @P<
taxo [taxar] <fmc> <*2> **V** PR 1S IND VFIN @FMV
em [em] <sam-> **PRP** @<ADVL
a [o] <artd> <-sam> **DET** F S @>N
UE [UE] **PROP** F S @P<
. [.] PU <<<

In this case, the word *durão* was recognized as an adjective, instead of a proper noun (*Durão Barroso*), because it is written in lowercase. On the other hand, the prior negative noun "tacho" (*approx*. "job for the boys") was analyzed as a verb (*taxar / to tax*) due to a common "Internet-accepted variation" ("*ch*" → "*x*").

We believe that robust opinion detection/classification in UGC can only be achieved by using text classification techniques exploring the rich set of features that may be extracted from text and other fine-grained linguistic resources. Some of these features may consist of lexically-derived information, such as n-grams or information about the prior-polarity of words, while others may be identified using lexico-syntactic patterns to detect typical opinion-bearing structures. Hence, in parallel to lexicon development, we are creating high precision lexico-syntactic rules. We started by confining the analysis to an ensemble of typical syntactic constructions where predicative adjectives and nouns, such as the ones described in the sentiment lexicon, may occur. Below, we illustrate some elementary adjectival predicative constructions including a political entity (PE):

(1) PE Vasp* Prep$^{?}$ Vcop^{1} Adv* Adj
(e.g. *Sócrates continua a ser muito convencido*$_{neg}$ → S_{neg})
(2) PE Vasp* Prep$^{?}$ Vaux^{1} Artind Adj$_{pos|neut}$ Adj$_{neg}$
(e.g. *Sócrates é um bom*$_{pos}$ *mentiroso*$_{neg}$ → S_{neg})
(3) PE Advneg Vasp* Prep$^{?}$ Vaux^{1} Adv* Adj
(e.g. *Sócrates nunca foi desonesto*$_{neg}$ → S_{pos})
(4) PE deixar de Vaux^{1} Adv* Adj
(e.g. *Sócrates deixou de ser autoritário*$_{neg}$ → S_{pos})
(5) PE Vasp* Prep$^{?}$ Vaux^{1} Adv* Adj, mas Adv* Adj
(e.g. *Sócrates é arrogante*$_{neg}$, mas *competente*$_{pos}$ → S_{pos})
(6) Artdef Adj$_{pos|neut}$ do|da|dos|das PE...
(e.g. *O inteligente*$_{pos}$ *do Sócrates...* → S_{neg})

In construction (1), the adjective occurs in a predicative context, i.e., it relates a PE with their subject through a copulative verb (*Vcop*), which may be preceded by an aspectual verb (*Vasp*). Moreover, the adjective may be modified by an adverb (*Adv*). The application of this regular expression makes it possible to extract the sentences that match this pattern, and classify them according to the polarity assigned to adjectives in the lexicon.

However, sentences often include more than one lexical unit assigned to opposite polarities in the lexicon, as illustrated in construction (2). As a result, it is crucial to identify which is the predicate of the sentence, in order to classify it. Different types of negation are also considered in syntactic-semantic rules, as illustrated in constructions (3) and (4). The sentences matching those patterns must be classified as positive or negative, by reversing the polarity value assigned to the predicate they refer to. This implies that they must be classified as negative if containing a positive predicate, and vice-versa.

Lexico-syntactic rules also account for the possibility of co-occurrence in the same sentence of two or more predicates presenting similar or opposite polarities. For instance, the subject of construction (5) is modified by two predicates, which are combined in a coordinate structure, by means of the *mas* (*but*) adversative conjunction. In this case, we classify the sentence according to the polarity code exhibited by the last element of the coordination.

In some cases, the prior polarity assigned to a predicate at lexical level can be changed in particular contexts, such as the case of construction (6).

5 Learning Opinion Classifiers

Practical opinion classification cannot be performed using only a set of predefined lexico-syntactic rules [6]. To allow generalization, and thus increase recall in opinion classification, it is crucial to train an automatic text classifier. For this purpose, we are currently developing a reference corpus for the training and evaluation of opinion classification procedures.

5.1 Reference Corpus

We started by collecting opinion-rich data from the web site of one of the most read Portuguese newspapers. We obtained a collection of 8,211 news and linked comments to that news posted by on-line readers. This collection covers a period of 5 months (November 2008 to March 2009). It includes about 250,000 user posts, containing approximately 1 million sentences. We are now proceeding with the annotation of this collection in order to develop the reference corpus. We will identify mentions to political entities in the comments (both by name and by paraphrastic or anaphoric mention) and detect whether there is an expressed positive or negative opinion about those entities at the sentence level.

At present, we are focusing on a small set of popular political entities occurring in specific constructions, including those described in Section 4.3, in order to find high precision candidate sentences expressing positive and negative opinions. Such candidate sentences are being manually validated, and the remaining sentences of the comments mentioning the same entity are being automatically classified. This procedure is based on the assumption that non-recognized opinionated constructions expressed in a particular comment about a specific entity are consistent with the opinions previously found by rules in the rest of the comment. This enables quick annotation of the collection about certain political entities, which includes opinionated sentences with a large diversity of structures. Some of these would not be easily detected by linguistic rules.

Results obtained so far show that the performance of lexico-syntactic rules depends on the polarity conveyed in recognized text. In the case of expressed negative opinions, the precision rate is approximately 90%. For positive opinions expressed in text, the precision decreases to 60%. In both cases, we achieved a very low recall, which is not surprising due to the small number and strict type of syntactic patterns explored in this work.

Our experiments also demonstrate that it is possible to propagate the opinions found by the lexico-syntactic patterns to other sentences of the user comments mentioning the same entity, increasing the number and diversity of annotated sentences. Again, propagation success is higher for negative opinions (almost 100%) than for positive opinions (around 75%).

5.2 Feature Selection and Classification Algorithms

The challenge for correct classification lies in selecting the correct set of features for describing opinions in text. Some of the relevant features for opinion classification are associated with the presence in text of elements listed in the sentiment lexicon. However, there are relevant clues about opinion in features related to (i) punctuation usage (e.g. heavy punctuation); (ii) graphic marks (e.g. full capitalization); (iii) distance of elements to the entity at stake (e.g. immediately before/after entity); (iv) POS information, and (v) more refined syntactic and semantic information. The best feature selection of features will be investigated by evaluating the performance of classification over the reference corpus with different machine learning algorithms [12].

So far we have made initial experiments in a more constrained setting. We manually annotated a selected set of about 2,000 short news (title plus 1-3 sentences) gathered from newspaper RSS feeds and covering either a negative or positive event associated to two political entities: the prime-minister and the leader of the opposition of Portugal. We trained an SVM classifier for predicting if unseen news are about either a positive or negative event associated to one of the two. We experimented several feature sets, including bag-of-words, n-grams plus relative positions, prior polarity of words and POS. Results show that we are able to achieve a precision rate over 90% and an F-measure over 80% in predicting whether news are positive or negative for several of the possible feature combinations. Although UGC represents a much greater challenge, and despite the fact that we are considering a scenario with fewer restrictions in OPTIMISM, we believe that these results support the feature exploration strategy we are following.

6 Implementation and Evaluation Plans

We plan to release software prototypes capable of processing Portuguese texts conveying opinions about relevant political entities as they are published on the web for user evaluation. We have adopted an agile methodology organized in short sprints that add/modify functionalities as a result of observing new potential uses, alternative user interfaces or new characteristics of the opinion mining and web crawling modules of the system. The software components will be continuously optimized, as the training corpora improve and we obtain user feedback.

We started by defining a simple database model for storing the collected news items and comments (and metadata). A focused web crawler populates the database and the opinion mining software then fetches sets of news items for annotation from the database. The OPTIMISM software will also include a library of components for computing statistics on opinion data and plotting these statistics as charts and trends. The actual user interface design on both prototypes will depend on the characteristics of the obtained data, the methods used to process such data, and observation of user access patterns.

The evaluation of the OPTIMISM crawler will focus on measuring its capability to re-discover, from a set of initial seeds, known relevant blogs and Web2.0 from a known list to be compiled by the political scientists in the team, while minimizing the number of irrelevant blogs and comments to the task at hand.

Evaluation of the system will be performed through periodic user satisfaction surveys. We will also need to ensure that the perception on the quality and usefulness of the mining software is not hampered by a poorly designed user interface.

Planned experiments also include exploring the relationship between sentiments towards political entities as measured by the prototype and those obtained in political opinion polls. On the one hand, polls to be conducted by the Catholic University's Poll Centre will include questions designed to detect sub-samples of social web users, enabling the matching of data generated by the prototype with that obtained among representative samples at the same time periods. On the other hand, trends in all published public opinion polls and those obtained through the prototype with be analyzed both through conventional time series techniques and wavelet decomposition techniques appropriate for analyzing the relationship between two signals and the role of critical events [1].

7 Conclusion

OPTIMISM is a system for performing real-time opinion mining about political entities over the enormous wealth of user-generated content being currently produced. This system differs from most previous opinion mining projects in multiple aspects. It aims at detecting opinions about specific political entities regarding an open set of topics or issues. It is intended to explore highly reactive and short-lived user-generated content such as social web sites' status messages and comments posted on on-line newspapers, which represent a great challenge due to their short, and often non-grammatical, nature. For content selection, we are planning to explore deep crawling through interaction with global search engines (that perform broad crawls) for locating new potentially relevant content, but we incorporate aspects of incremental and deep crawling for obtaining the data.

We are currently using a novel semi-automatic strategy for building a reference corpus for training text classifiers, which mines newspaper comments with high-precision lexical-syntactic patterns to gather a diverse set of examples of opinionated text. Such corpus will be used to explore feature selection policies to be used by text-classification algorithms.

Finally, the evaluation plan includes contrastive analysis of observation data derived from the prototype's use against true political opinion poll data.

Acknowledgments:

This work was partially supported by grants SFRH/BPD/45416/2008 and SFRH/BD/23590/2005 from FCT-Portugal, co-financed by POSI.

References

1. Aguiar-Conraria, L., Magalhães, P.C.: Dynamical decomposition of political time-series, paper prepared for the American Political Science Association 2009 meeting.
2. Banea, C., Mihalcea, R., Wiebe, J.: A Bootstrapping Method for Building Subjectivity Lexicons for Languages with Scarce Resources. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, European Language Resources Association (ELRA), Marrakech, Morocco (2008)
3. Bick, E.: The Parsing System "Palavras": *Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Dr.phil. thesis. Aarhus University. Aarhus University Press. Denmark (2000)
4. Carvalho, P.: *Análise e Representação de Construções Adjectivais para Processamento Automático de Texto. Adjectivos Intransitivos Humanos*. PhD Thesis, University of Lisbon (2007)
5. Chakrabarti, S., Dom, B.E., Kumar, S.R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., Kleinberg, J.: Mining the Web´s link structure. In *Computer*, 32(8):60.67 (1999)
6. Dasgupta, S., Ng, V.: Mine the Easy, Classify the Hard: Experiments with Automatic Sentiment Classification. To appear in *Proceedings of Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP)*, 2009.
7. Drezner, D, Farrell, H.: Introduction: Blogs, Politics and Power, In *Public Choice*, 134, 1-2, pp. 1--13 (2008)
8. Edwards, J., McCurley, K.S., Tomlin, J.A.: An adaptive model for optimizing performance of an incremental web crawler. In *International World Wide Web Conference*, pp. 106–113 (2001)
9. Esuli, A., Sebastiani, F.: Determining term subjectivity and term orientation for opinion mining. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, EACL'06*, Trento, Italy (2006)
10. Ginsberg J., Mohebbi, M.H., Patel, R.S., Brammer, L., Smolinski, M.S., Brilliant, L.: Detecting influenza epidemics using search engine query data, *Nature* (19 Nov 2008), doi: 10.1038/nature07634, Letters to Editor (2008)
11. Gomes D., Silva M.J.: The Viúva Negra crawler: an experience report. In *Software: Practice and Experience (SPE)* 2(38), pp. 161–168 (2008)
12. Guyon, I., Elisseff, A.: An Introduction to Variable and Feature Selection**.** In *Journal of Machine Learning Research* 3, pp. 1157--1182. (2003)
13. Hatzivassiloglou, V., McKeown, K.: Predicting the Semantic Orientation of Adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, Madrid, Spain (1997)
14. Heydon, A., Najork, M.: Mercator: A scalable, extensible web crawler. In *World Wide Web*, 2(4), pp. 219–229 (1999)
15. Howe, J.: The Rise of Crowdsourcing, *Wired*, Vol. 14, No. 6 (2006)
16. Kim, S., Hovy, E.: Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. In *Proceedings of ACL/COLING Workshop on Sentiment and Subjectivity in Text*, Sydney, Australia (2006)
17. Kim, S, Hovy, E.: Crystal: Analyzing predictive opinions on the web. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (2007)
18. Magalhães, P.: Pre-Election Polls in Portugal: Accuracy, Bias and Sources of Error, 1991-2004, *International Journal of Public Opinion Research* 17 (4), pp. 399--421 (2005)

19. Martins, B, Silva, M.J.: Language Identification in Web Pages. In *Proceedings of ACM-SAC-DE, the Document Engineering Track of the 20th ACM Symposium on Applied Computing*, pp. 764--768, Santa Fe, Mexico (2005)
20. Morinaga, S., Yamanishi, K., Tateishi, K., Fukushima, T.: Mining product reputations on the Web. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 341--349 (2002)
21. Mullen, T., Malouf, R.: A preliminary investigation into sentiment analysis of informal political discourse. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pp. 159--162 (2006)
22. Ntoulas, A., Zerfos, P., Cho, J.: Downloading textual hidden web content through keyword queries. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pp. 100–109, ACM Press, NY, USA (2005)
23. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd ACL*, Barcelona, Spain (2004)
24. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification Using Machine Learning Techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2002)*, Philadelphia, USA (2002)
25. Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis, In *Foundations and Trends in Information Retrieval* 2(1-2), pp. 1–135 (2008)
26. Thomas, M., Pang, B., Lee, L.: Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 327–335 (2006)
27. Turney, P., Littman, M.: Measuring praise and criticism: Inference of semantic orientation from association. In *ACM Transactions on Information Systems*, 21(4), pp. 315–346 (2003)
28. Whitelaw, C., Garg, N., Argamon, S.: Using Appraisal groups for sentiment analysis. In *Proceedings of ACM SIGIR Conference on Information and Knowledge Management (CIKM 2005)*, Bremen, Germany (2005)
29. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity: an exploration of features for phrase-level sentiment analysis. In *Computational Linguistics* 35(3), pp. 1-34 (2009)

Chapter 12

WNI - Web and Network Intelligence

An Ultra-Fast Modularity-Based Graph Clustering Algorithm

Leonardo Jesus Almeida and Alneu de Andrade Lopes

Institute of Mathematics and Computer Science - ICMC
University of São Paulo/Campus of São Carlos - (USP São Carlos)
P.O. Box 668, São Carlos, SP, Brazil

Abstract. In this paper, we propose a multilevel graph partitioning scheme to speed up a modularity-based graph clustering technique. The modularity-based algorithm was proposed by Newman for partitioning graphs into communities without the input of the number of clusters. The algorithm seeks to maximize a modularity measure. However, its worst-time complexity on sparse graphs is $O(n^2)$, where n is the number of vertices, which can be prohibitive for many applications. The multilevel graph partitioning scheme consists of three phases: (i) reduction of the size (coarsen) of original graph by collapsing vertices and edges, (ii) partitioning the coarsened graph, and (iii) uncoarsen it to obtain a partition for the original graph. The rationale behind this strategy is to apply a computationally expensive method in a coarsened graph, i.e., with a significantly reduced number of vertices and edges. Empirical evaluation using this approach demonstrate a significant speed up of the modularity-based algorithm, keeping a good quality clusters partitioning.

Key words: Clustering, Graph Clustering, Multilevel Graph Clustering

1 Introduction

Clustering is an important problem with many applications, and a number of different algorithms have been proposed over the past decades. Recently, compelling application domains have become available in which to concept learning requires effective handling of relational data. Social networks and web analysis, for instance, require the representation of structural relations among people or web pages. In these domains, data must represent relationship between entities rather than entity's attributes. Such representation motivates approaches to mine concepts from graph-based data.

Graphs are formed by a set of vertices and a set of edges that are connections between pairs of vertices [1, 2]. Graph clustering is the task of partitioning similar vertices into clusters taking into consideration the edge structure of the graph so that there are many edges within each cluster and few edges between the clusters [3, 4]. Unfortunately, finding an optimal partition is a problem known to be NP-complete [5], so it's necessary to use some heuristics for a practical solution.

To overcome the problem of finding the number of clusters and speed up the clustering process we apply the Newman's modularity algorithm [6] as the partitioning algorithm in a multilevel graph partitioning approach [7–9]. A multilevel graph partitioning scheme [10, 8, 9] consists of three phases: (i) reduction of the size (coarsen) of the original graph by collapsing vertices and edges; (ii) partitioning the coarsened graph; and (iii) uncoarsen it to construct a partition for the original graph . The rationale behind this strategy is to apply a computationally expensive method in a coarsened graph with significant reduction of number of vertices and edges. This approach can speed up the fast modularity algorithm, keeping a good quality cluster partitioning.

In the following section we describe some background concepts in graph clustering. In Section 3, we explain the proposed approach of joining Multilevel Graph Partition and Newman's modularity Q. In Section 4 we evaluate the approach on some datasets. Finally, in Section 5, we present the conclusions and future work.

2 Graph Clustering

The problem of partitioning a graph in k different clusters is defined as follows: Given a graph $G = (V, E)$ with $|V| = n$, finding subgraphs $V_1, V_2, ..., V_k$ such that $V_i \cap V_j = \emptyset$ for $i \neq j$, $\bigcup_i V_i = V$ and the number of edges of E that connects vertices from different clusters are minimized. This quantity is known as *edge-cut*. The clusters founded are generally represented in a vector P of length n, such that for each vertex $v \in V$,$P[v]$ is in $[1, k]$. The value of k is usually given as input for many graph clustering algorithm, although there are techniques to suggest a good value.

The above goal described can be interpreted in various ways leading different criteria for optimization. We briefly review some specific objective functions that are most related to our work:

Ratio association maximization It takes the partitioning scheme that leads to the highest intra-cluster average degree, i.e., the summation of number of edges in the cluster i, $links(V_i, V_i)$, divided by the number of vertices in i, $|V_i|$, expressed by Equation 1.

$$RAssoc(G) = \max_{V_1,...,V_k} \sum_{i=1}^{k} \frac{links(V_i, V_i)}{|V_i|}. \quad (1)$$

Ratio cut minimization The difference between Ratio cut and Ratio association is that the first seeks to minimize the *edge-cut* while the second seeks to maximize internal cluster edges weight. It is obtained with Equation 2

$$RCut(G) = \min_{V_1,...,V_k} \sum_{i=1}^{k} \frac{links(V_i, \overline{V_i})}{|V_i|}, \quad (2)$$

where $\overline{V_i}$ is the subset of vertices $V - V_i$.

Normalized cut minimization The normalized cut objective [11] seeks to minimize the cut relative to the number of edges in a cluster instead of its size. The objective is expressed by Equation 3

$$NCut(G) = \min_{V_1,\ldots,V_k} \sum_{i=1}^{k} \frac{links(V_i, \overline{V_i})}{degree(V_i)}, \tag{3}$$

where $degree(V_i) = |\{(u, v) \in E | u, v \in V_i\}|$.

Modularity maximization Modularity measures the fraction of the cluster's inner edges e_{ii} minus the expected value a_i if the edges are placed at random [12]. Let e_{ij} be the half fraction of edges in G that connects vertex in different clusters. So, $|E| = e_{ij} + e_{ji} + e_{ii}$. We can calculate a_i by $a_i = \sum_j e_{ij}$. The measure is given by Equation 4

$$Q = \sum_i (e_{ii} - a_i^2). \tag{4}$$

If the edges are connected at random the fraction of inner edges in a cluster i is a_i^2, which produces $Q = 0$. Any $Q \neq 0$ indicates the presence of communities structure.

In 2004, Newman proposed an agglomerative hierarchical algorithm for partitioning graphs into communities without the input of the number of clusters [6]. The algorithm seeks to maximize the modularity measure given by Eq.4. Starting with each vertex as a single community, it repeatedly joins communities together in pairs. At each step is selected the pair which promotes the highest modularity gain, that is calculated by Equation 5.

$$\Delta Q = 2(e_{ij} - a_i a_j). \tag{5}$$

Following a join, it's necessary to update the e_{ij}'s values, a task with worst-time complexity $O(n)$. At any time, there will be at most $|E|$ edges in graph, thus each step of the algorithm takes $O(|E| + n)$. The worst-time cost occurs when it's necessary to perform $n - 1$ joins, so the entire algorithm runs in time $O((|E| + n)n)$, or $O(n^2)$ in a sparse graph [6], which can be prohibitive.

In the next section, we propose a graph clustering technique based on the Newman's modularity algorithm [6] applied in a multilevel graph partitioning approach [7–9], aiming to speed up the clustering process.

3 Multilevel Graph Partitioning

Multilevel schemes [10, 7] are relatively fast and provide excellent partitions for a wide variety of graphs. Formally, a multilevel scheme works as follows: consider a weighted graph $G_0 = (V_0, E_0)$, with weights both on vertices and edges. A multilevel graph partitioning algorithm consists of the following three phases, illustrated in Figure 1.

Coarsening phase The graph G_0 is transformed into a sequence of smaller graphs $G_1, G_2, ..., G_t$ such that $|V_0| > |V_1| > |V_2| > ... > |V_t|$.

Partitioning phase A k-way partition P_t of the graph $G_t = (V_t, E_t)$ is computed that partitions V_t into k parts.

Uncoarsening phase The partition Pm of G_t is projected back to G_0 by going through intermediate partitions $P_{t-1}, P_{t-2}, ..., P_1, P_0$.

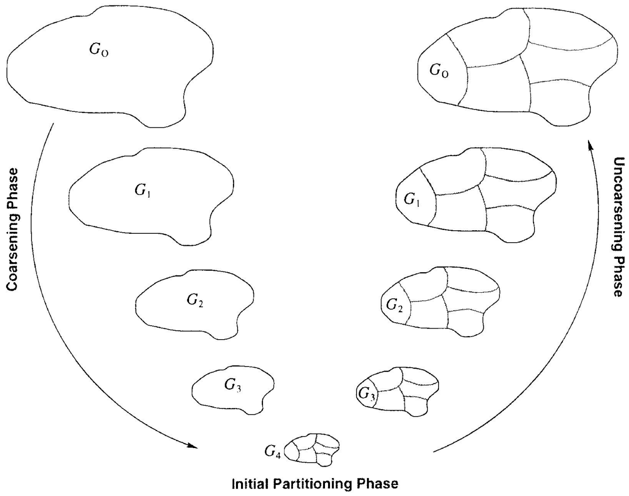


Fig. 1. Multilevel Graph Partition Scheme [8]

3.1 Coarsening phase

In the Coarsening phase, the initial graph G_0 is iteratively reduced to smaller graphs, such that the number of vertices and edges decrease. Coarsen a graph from G_i to G_{i+1}, set of vertices in G_i are combined to form supervertices in G_{i+1}. In order to preserve the connectivity in the smaller graph, the edges of supervertex are taken to be the union of the edges of corresponding vertices in the previous graph. In the case where the vertices which form the supervertex have common adjacent vertices, the edge between the supervertex and these vertices will be the union of the old edges.

The subset of vertices pairs chosen to be collapsed can be formally defined in terms of matchings. A matching of a graph is a set of edges, no two of which are incident on the same vertex [7]. In the following we describe three criteria for selecting the maximal matching. We notice that the complexity of the coarsening phase, $O(|E|)$, where $|E|$ is again the number of edges in the graph, is asymptotically similar to three methods, although the proportionality constant of Modified heavy-edge matching is greater [8].

Random matching (RM) The random matching generates the maximal matching using a randomized algorithm. It works as follows. Initially, all vertices in the graph are marked as unmatched. Then, the vertices are randomly visited until all vertices are visited or the graph size desired reduction is achieved. If an unmatched vertex v is selected for matching, we then seek randomly one of its neighbors that has not been matched yet. If a vertex u exists, we mark both vertices as matched and include the edge (v, u) in matching.

Heavy-edge matching (HEM) The idea behind heavy-edge matching is to reduce the edge-weight of the coarser graph by selecting a maximal matching whose edges have large weight. A heavy-edge matching is, using a randomized algorithm, similar to random matching. Let v be the selected vertex, and H the subset of unmatched v's neighbors. Heavy-edge matching selects for matching the vertex $u \in H$, such that the edge (v, u) has greater weight.

Modified heavy-edge matching (MHEM) The Modified heavy-edge matching is closely similar to heavy-edge matching. The difference occurs when there are more than one maximal weight adjacent vertices. Let v be the selected vertex, and H the subset of unmatched v's neighbors with maximal edge weight, and W_{v-u} the sum of the weights of the edges of u that connect u to vertices adjacent to v. Modified Heavy-edge matching selects for matching the vertex $u \in H$, such that W_{v-u} is maximized.

3.2 Partitioning phase

The partitioning phase of a multilevel scheme computes the partition P_t of the coarser graph G_t. Various algorithms can be used to obtain the partition such as iterative, hierarchical, divisive or agglomerative [4]. Since the size of the coarser graph G_t is significantly smaller then the original one (depending on coarsening stop criterion, $|V_t| < 100$), this step will spend a small amount of time [7].

Our implementation uses the Fast Modularity algorithm in this phase. The algorithm is an agglomerative hierarchical method that seeks a greedy optimization of modularity Q measure. As one feature, the method calculates the modularity value along each step and it selects the partition of the step which had the greater modularity value.

3.3 Uncoarsening phase

During the uncoarsening phase, the partition found out in partitioning phase (P_t) is projected back through each level until the original graph is achieved. Since each supervertex of G_{i+1} should contain two or more distinct vertices of G_i, we can obtain P_i by simply assigning to the collapse vertices the same partition of supervertex in P_{i+1}. Even though P_{i+1} is a local minima partition of G_{i+1}, the projected partition P_i may not be at a local minima with respect to G_i . Since G_i is less coarse, it has more degrees of freedom that can be used to improve P_i , and decrease the edge-cut. Hence, it may still be possible to improve the projected partition of G_{i-1} by local refinement heuristics [7].

4 Experiments

We have applied our algorithm on two scientific graphs: *Netscience* and *CBR-ILP-IR*. The *Netscience* is a network of coauthorships between scientists [13]. The graph has a total of 1,589 vertices in it, representing scientists from a broad variety of fields. As the graph has more than one component, only the 379 vertices and 914 edges falling in the largest connected were used. The *CBR-ILP-IR* is a network of similarity between articles from three different topics: Case Based Reasoning, Inductive Logic Programming and Information Retrieval. The graph has a total of 574 vertices that represents documents, and the 19,213 edges represent the similarity between the documents calculated using cosine similarity measure.

We evaluated the proposed algorithm partitioning quality considering the objective functions described in Section 2. We also considered the algorithm runtime in milliseconds. The three matching schemes described in Section 3 were implemented and used in Coarsening Phase for the two graphs. In Partitioning Phase we used our Fast Modularity algorithm [6] implementation. We decided using no refinement approach, so in the Uncoarsening Phase we simple assign to inner vertices the same partition of Supervertex. The decision was taken because we aimed to verify the effectiveness of using the Fast Modularity Algorithm on the partitioning Phase. Adding a refinement algorithm would certainly improve the quality performance, but it could hide an bad initial partition created by the Fast Modularity. As the three approaches are based on random choices, we repeated 100 times each run and we considered the measures average. The experiments were perfomed on an Intel Xeon 2 GHz processor with 4GBytes of RAM memory.

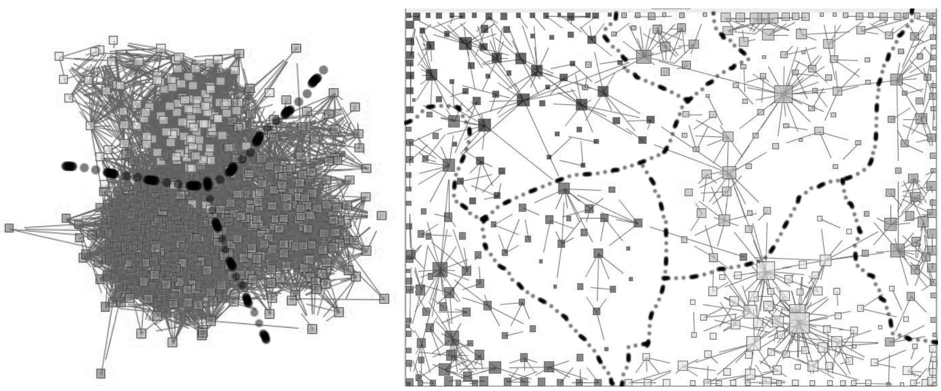
In Table 1 is presented the results of our experiments with *Netscience* and *CBR-ILP-IR* data sets. For the both graphs, HEM and MHEM lead to partitions whose quality measures are better than that ones produced by RM. HEM and MHEM have similar overall quality. Nevertheless, all the three matching schemes have worse quality results than that obtained by applying the Fast Modularity algorithm on entire *CBR-ILP-IR* graph. The *CBR-ILP-IR* graph is denser than *Netscience*, and for many edges, the weights values are similar. Thus, as the matchings are performed at random, it could make unsuitable joins.

The variation of modularity Q is minimal, but it had a little increase when it was used multilivel partitioning HEM or MHEM on *NetScience*. Otherwise, it decrease a little on *CBR-ILP-IR*. On *Netscience*, the coarsening phase produced reduced graphs for partitioning better than on *CBR-ILP-IR*, because of its sparsity. So, as the number of edges is smaller in reduced graph, the fractions e_{ii} and e_{ij} are more meaningful for Fast Modularity. For both graph, the RM schemes produced low modularity values, which can be explain because RM tends do produce graphs with larger edge-cut than HEM and MHEM.

In Figure 2 it's shown a visualization of *Netscience* and *CBR-ILP-IR* graphs. The regions limited by dashed lines represent the partitions founded by our algorithm using HEM in coarsening phase, and Fast Community as partitioning algorithm. In spite of the high number of edges, in Figure 2(a) we can see the three different groups representing the three main areas CBR, ILP and IR. The *Netscience* graph, Figure 2(b), has less edges, so it is clearer to we see each of seven partitions founded by the our algorithm. Again the lines separate the partitions, and the shape's size indicates the vertex degree, which means how many papers an author published with coauthors.

Table 1. Objectives functions values and elapsed algorithm runtime for *CBR-ILP-IR* and *Netscience* data sets

CBR-ILP-IR	None	RM	HEM	MHEM	*Netscience*	None	RM	HEM	MHEM
RAssoc(G)	**161.37**	84.57	120.34	122.81		**74.83**	30.86	34.62	35.35
RCut(G)	**40.11**	97.12	49.13	53.42		5.64	2.69	**1.84**	1.87
NCut(G)	**0.63**	1.44	0.69	0.76		1.19	0.58	**0.38**	0.39
Q	**0.388**	0.194	0.352	0.345		0.462	0.463	**0.478**	0.477
$Time(ms)$	68514	**24345**	26276	26841		22670	6422	**5779**	5895



(a) *CBR-ILP-IR* (b) *Netscience*

Fig. 2. Graph representing CBR-ILP-IR and Netscience data sets rendered by Prefuse [14]. The colors represent the clustering found by the algorithm using HEM in Coarsening Phase and Fast Modularity in Partitioning Phase.

We ran another experiment comparing Fast Modularity and HEM on two other networks. The *cond-2003* is a network of coauthorships between scientists posting preprints on the Condensed Matter E-Print Archive between 1995 and 2003, and the *cond-2005* which is an extension including postings until 2005 [15]. Both networks are available on Newman's network datasets [1]. The *cond-2003* network has 31,162 vertices and 116,181 edges, while *cond-2005* has 40,420 vertices and 171,734 edges.

In Table 2 is presented the results of second experiment. For *cond-2003* network the modularity values produced by both algorithms are close, besides HEM were a little better on *cond-2005*. For both networks, HEM runtime is smaller than Fast Modularty. As our aim was to analise how runtime grows along with the network size, we also plotted a graphic comparison between both algorithms runtime as is shown in Figure 3. The points in each graphics are related to the number of vertices (edges) in *netscience, CBR-ILP-IR,cond-2003* and *cond-2005*, respectively. Taking into account only the number of vertices in the network the Fast Modularity's runtime grows faster than HEM, Figure 3(a). The same occurred when the runtimes was plotted against the number of edges, as it is presented in Figure 3(b). However the HEM runtime grow curve tends to fit a liner function in both graphics.

Performance is an important issue when we work on large graphs, and the main advantage of our approach is the runtime reduction. On the evaluated graphs the multilevel approach speed up the clustering tasks by factors from 2 to 7 compared to Fast Modularity algorithm.

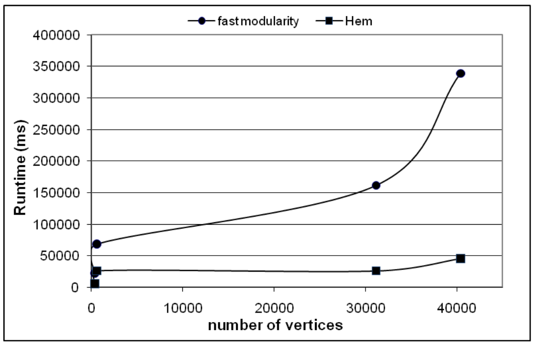
Table 2. Modularity quality and elapsed algorithm runtime for *cond-2003* and *cond-2005* networks

cond-2003	Fast Modularity	HEM	*cond-2005*	Fast Modularity	HEM
Q	**0.415**	0.411		0.397	**0.406**
$Time(ms)$	161399	**25097**		338995	**45444**

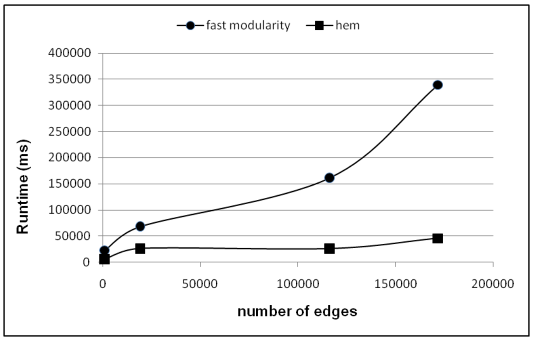
5 Conclusion

In this paper we presented a method for clustering data represented as graphs. The method follows a multilevel graph clustering schemes, making use of the Fast Modularity algorithm in the partitioning phase. This approach has a considerable speed advantage over direct applying the partitioning algorithm on an entire graph. As the partitioning algorithm works on the reduced graph, the runtime complexity is smaller than it would be if it was done on entire graph. So, we observed that the entire method worst-time is $O(|E|)$, which allow us to perform

[1] http://www-personal.umich.edu/ mejn/netdata/



(a) *Runtime by number of vertices*



(b) *Runtime by number of edges*

Fig. 3. Performance comparison of Fast Modularity and HEM

clustering in large graphs in a short time. Moreover, the method keeps partition quality, since the coarser graph, on which the clustering algorithm is applied, is built attempting to be a good representation of original one.

In our experiments, we evaluated the proposed algorithm on three different graphs observing the runtime and four objectives measures. For the two graphs, we compared the results of using multilevel partitioning RM, HEM and MHEM against the original Fast Modularity Algorithm. HEM and MHEM produced quality results close to Fast Modularity, but in much smaller runtime.

We consider that our method can also have the partitions quality improved by applying a refinement algorithm on uncoarsening phase. The refinement algorithm will allow to switch boundary vertices from a partition to other upgrading an objective function, without increase the global worst-time. Another interesting point of improvement is to modify Fast Modularity algorithm in the Partitioning Phase to take into account edges weights.

References

1. Bollobas, B.: Modern Graph Theory. Springer (July 1998)
2. Diestel, R.: Graph Theory (Graduate Texts in Mathematics). Springer (2005)
3. Cook, D.J., Holder, L.B., Ketkar, N.: Unsupervised and supervised learning in graph data. In: Mining Graph Data. John Wiley & Sons (2007)
4. Schaeffer, S.E.: Graph clustering. Computer Science Review **1**(1) (2007) 27–64
5. Garey, M., Johnson, D., Stockmeyer, L.: Some simplified NP-complete graph problems. Theoretical Computer Science **1** (1976) 237–267
6. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review E **69** (2004) 066133
7. Karypis, G., Kumar, V.: Analysis of multilevel graph partitioning. In: Supercomputing '95: Proceedings of the 1995 ACM/IEEE conference on Supercomputing, New York, NY, USA, ACM Press (1995)
8. Karypis, G., Kumar, V.: Multilevel k-way partitioning scheme for irregular graphs. Journal of Parallel and Distributed Computing **48** (1998) 96–129
9. Dhillon, I.S., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors: A multilevel approach. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(11) (2007) 1944–1957
10. Hendrickson, B., Leland, R.: A multilevel algorithm for partitioning graphs. Technical Report SAND93-1301, Sandia National Laboratories (1993)
11. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8) (2000) 888–905
12. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E **69** (2004) 026113
13. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) **74**(3) (2006) 036104
14. Heer, J., Card, S.K., Landay, J.A.: prefuse: a toolkit for interactive information visualization. In: CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2005) 421–430
15. Newman, M.E.J.: The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences of the United States of America **98** (2001) 404

An Updated Portrait of the Portuguese Web

João Miranda and Daniel Gomes

Foundation for National Scientific Computing
Av. do Brasil - 101, 1700-066 Lisboa, Portugal
{joao.miranda,daniel.gomes}@fccn.pt
http://arquivo.pt

Abstract. This study presents an updated characterization of the Portuguese Web derived from a crawl of 48 million contents composed by all media types (2.5 TB of data), performed in March, 2008. The resulting data was analyzed to characterize contents, sites and domains. Sites are typically small and most IP addresses host a single site. The content size and duplication rate vary according to media type. This study was performed within the scope of the Portuguese Web Archive.

Key words: Portuguese Web Archive, Measurements, Characterization

1 Introduction

The Web is an important component of the modern world and a massive source of information. Web characterization is important because it helps studying and describing its evolution, but it is a challenging task due to the large amount of contents involved and the complexity of relations that can be extracted from a large crawl.

The results presented in this study are derived from a crawl performed within the scope of the Portuguese Web Archive (PWA), a project of the Foundation for National Scientific Computing (FCCN) that aims to preserve the information published on the Portuguese Web [8]. One of the objectives of the PWA is to monitor its evolution and periodically publish studies characterizing it. The main contribution of this study is an updated characterization of the Portuguese Web, presenting measurements for metrics that were not previously studied and that can be used as baseline for future trend analysis.

This paper is organized as follows. Section 2 presents related work and Section 3 presents the methodology adopted to conduct the experiment. Section 4, 5 and 6 characterize contents, sites and domains, respectively. Section 7 draws the main conclusions and proposes future work.

2 Related work

Several studies characterizing national Webs have been published during the last years. Baeza-Yates et al. characterized several national webs and compared the results derived from 12 Web characterization studies, unveiling similarities and

Status Code	% codes	Description
200	85.2%	Success - OK
302	7.2%	Redirection - Found
404	5.1%	Client Error - Not Found
301	1.3%	Redirection - Moved Permanently
303	0.4%	Redirection - See Other
403	0.2%	Client Error - Forbidden
500	0.2%	Server Error - Internal Server Error
400	0.2%	Client Error - Bad Request
401	0.2%	Client Error - Unauthorized
503	0.1%	Server Error - Service Unavailable
Other	0.0%	Other codes

Table 1. The 10 most common response codes.

differences between the collections [1]. Modesto et al. characterized the evolution of the Web of Brazil [16], making a comparison with the results previously obtained by Veloso [20]. Zabicka and Matejka analysed the Czech Web archive, performing a characterization of its contents [21]. Lasfargues et al. presented a characterization of the French Web derived from a crawl performed in 2007 [15].

Previous studies contributed to characterize the Portuguese Web. Nicolau et al. defined a set of metrics to characterize the Web within the national scientific community network [18]. Noronha et al. performed a crawl of selected online publications and characterized the obtained collection [19]. Gomes et al. produced two previous characterizations of the Portuguese Web. One derived from a crawl of 3.2 million textual contents performed in 2003 [10]. The other, presented the most prevalent media types on the Portuguese Web, based on a crawl performed in 2005 [7].

3 Methodology

The following terminology was adopted in this study. A *crawler* is a program that iteratively downloads contents and extracts links to find new ones. A *seed* is a URL used in the set of initial addresses to bootstrap a new crawl. A *site* is identified by a fully qualified domain name. For instance, `www.fccn.pt` and `arquivo-web.fccn.pt` are two different sites. A *content* is a file resulting from a successful HTTP download (200 response code). The presented amounts of data correspond to decimal multiples. For instance, 1 KB corresponds to 10^3 bytes.

The Web characterization results presented in this study were extracted from a crawl of the Portuguese Web, containing information belonging to all media types, performed by the PWA between March and May, 2008, using the Heritrix 1.12.1 crawler [17]. It started from a set of 180 000 seeds under the .PT domain, generated from a previous crawl. Table 1 presents the 10 most logged response codes. The total number of logged responses was 57 148 455. Some constraints were imposed to prevent the crawler against hazardous situations that could degrade its performance and bias results, such as spider traps, that are sites that generate an infinite number of addresses [11].

The following crawling constraints based on previous research results were imposed to use effectively our resources and respect politeness best practices toward servers [6]. The maximum number of URLs crawled per site was 10 000. The maximum size per content was 10 MB. Logical URL depth measures the number of hops from the entry page of a site to a given content. The maximum logical depth imposed was 5. The physical URL depth measures the number of slashes contained within a URL. For instance, the URL `www.a.com/b/c.html` presents a physical depth value of 2. The maximum physical URL depth imposed was 10. The exclusion rules can be provided through a file named *robots.txt* or a meta-tag *robots* embedded on a page [12]. The crawler respected the rules provided through both methods. Additionally, a courtesy pause of 2 seconds between requests to the same site was respected to avoid overloading servers with consecutive requests.

The boundaries of a national Web are difficult to define accurately [7]. However, country code top-level domains are a good hint that a site belongs to a national Web. It was assumed that a content belongs to the Portuguese Web if it met at least one of the following conditions:

1. Its site domain name was hosted under .PT;
2. It was hosted outside the .PT domain but it was embedded on a page hosted under the .PT domain. For instance, if a page under .PT required an image under .FR to be presented completely and maintain its original layout, that image was also crawled and considered to be part of the Portuguese Web;
3. It was hosted outside the .PT domain but it was redirected from a .PT domain site. This situation is frequent with sites of international companies that register their domain under .PT but redirect to a main site under other top-level domain.

4 Contents

The number of contents downloaded was 48 718 404 in a total amount of data of 2.5 TB. The number of contents excluded due to robots exclusion rules was 9.4% of the total number of requests processed.

4.1 URL length and physical depth

The URL length of contents is a feature used in search engine ranking algorithms to identify relevant results [4]. The URL length was counted as the number of characters excluding the protocol element. For instance, in `http://www.a.com/b.php?f=2` only the `www.a.com/b.php?f=2` string was considered. Thus, this URL presents a length of 19 characters. Figure 1 presents the URL length distribution. The obtained results show that 66% of the URLs have a length between 40 and 80 characters. URLs that resulted in a successful download presented a length varying from 5 to 2 072 characters. We inspected manually 10 of the longer URLs and they were all related to dynamically generated contents.

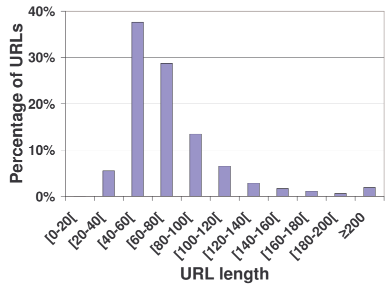


Fig. 1. URL length distribution (avg: 74.5, median: 63).

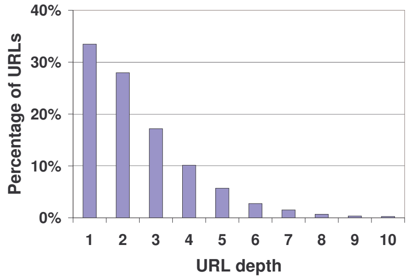


Fig. 2. URL physical depth distribution (avg: 2.5, median: 2).

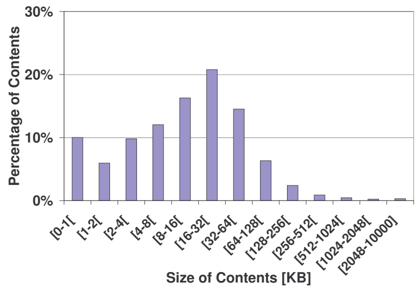


Fig. 3. Content size distribution in KB (avg: 50.4, median: 13.6).

	Media type	Avg size
1	text/html	31 KB
2	app'n/pdf	483 KB
3	image/jpeg	36 KB
4	text/plain	212 KB
5	app'n/zip	1 540 KB
6	app'n/x-gzip	844 KB
7	audio/mpeg	2 576 KB
8	app'n/octet-stream	996 KB
9	app'n/x-shockwave-flash	144 KB
10	image/gif	9 KB

Table 2. Average size of the media types that contributed with the larger amount of data.

The URL physical depth is a characteristic that reflects the importance of contents or the hierarchy of files, especially in sites with static contents. In a site with static contents, a content located deep in the site is often less relevant than one at the top [4]. Plus, deep URLs depths are associated with increased link failures [14, 5]. URL physical depth distribution is important when setting depth limits for crawling the Web. Figure 2 presents the URL physical depth distribution and shows that 94.4% of the URLs have a depth between 1 and 5, and only 5.6% have a depth between 6 and 10. The obtained results suggest that Web publishers are not organizing contents on their sites following a tree directory hierarchy, because the large majority of the contents are found at shallow levels of physical depth. Thus, relevance heuristics based on URL physical depth may be compromised by this fact.

4.2 Media types and sizes

Analyzing content sizes is useful to estimate the resources required to create Web data repositories. Figure 3 presents the distribution of content size. The obtained results show that 99% of the contents present sizes lower than 512 KB. The imposed maximum limit of 10 MB resulted in a total of 32 321 truncated

	Media type	% contents
1	text/html	57.8%
2	image/jpeg	22.8%
3	image/gif	9.4%
4	text/xml	1.9%
5	app'n/pdf	1.9%
6	image/png	1.3%
7	text/plain	1.0%
8	app'n/x-shockwave-flash	0.7%
9	text/css	0.7%
10	app'n/x-javascript	0.5%
-	Other	2.0%

Table 3. Top 10 media types measured by number of downloaded contents.

	Media type	% data
1	text/html	35.4%
2	app'n/pdf	17.9%
3	image/jpeg	16.1%
4	text/plain	4.2%
5	app'n/zip	3.0%
6	app'n/x-gzip	2.7%
7	audio/mpeg	2.7%
8	app'n/octet-stream	2.4%
9	app'n/x-shockwave-flash	2.1%
10	image/gif	1.6%
-	Other	11.9%

Table 4. Top 10 media types measured by amount of data.

contents, which represents just 0.05% of the total downloaded contents. Table 2 presents the average size of the media types that contributed with the larger amount of data.

There are hundreds of formats for digital contents and they all can be potentially published on the Web. However, only some formats are commonly used due to their characteristics, such as size or portability. It is interesting to follow which are currently the most prevalent media types, for instance, to select software format interpreters to include in mobile phone browsers that have limited capacities in comparison to desktop computers. Table 3 and Table 4 present the most prevalent media types regarding the number of contents and the total amount of data, respectively. The *text/html* type was the most common, with 57.8% of the contents and 35.4% of the total amount of data downloaded. A comparison between Table 3 and Table 4 shows that 6 media types exist in both. However, their relative presence varies. For instance, the *application/pdf* type occupies the 2^{nd} position in Table 4 with 17.9% of the total amount of data downloaded but the 5^{th} position in Table 3 with 1.9% of the number of downloaded contents.

4.3 Dynamically generated contents

There are contents dynamically generated on-the-fly when the Web server receives a request and that do not physically exist on disk [2]. We identified the presence of dynamically generated contents through the analysis of URLs following two approaches: embedded parameters and extension analysis. The former is based on the existence of a question mark in the URL, which according to the HTTP protocol indicates that the content received parameters from the client to be generated. For instance, `www.a.com/b.php?c=3` was assumed to be dynamically generated. The latter is based on the analysis of known extensions for dynamically generated content technology - PHP: Hypertext Preprocessor, Active Server Pages, JavaServer Pages, ColdFusion and Common Gateway Interface. We assumed these extensions to begin with the strings *.php*, *.asp*, *.jsp*, *.cfm*, *.cgi*. However, the presented methods have limitations and are unable to identify dynamically generated contents on several situations. For instance, when con-

Type	% dynamic contents	% total contents
PHP: Hypertext Preprocessor (.php)	48.3%	22.4%
Active Server Pages (.asp)	21.6%	10.0%
JavaServer Pages (.jsp)	1.9%	0.9%
ColdFusion (.cfm)	0.6%	0.3%
Common Gateway Interface (.cgi)	0.4%	0.2%
Other	27.0%	12.5%

Table 5. Distribution of dynamically generated contents.

tents are served without filename extension, when seemingly static URLs that contain no parameters reference dynamically generated contents or when contents are dynamically generated in response to a request containing parameters sent through the HTTP POST method. Thus, the obtained results should be interpreted as the minimum percentage of dynamically generated contents on the Web.

The obtained results show that 44.4% of the contents contained embedded parameters. Table 5 presents the distribution of dynamically generated contents. The 2^{nd} column refers to the percentage of contents regarding the total number of dynamically generated contents. The 3^{rd} refers to the percentage of contents regarding the total number of contents downloaded. The total number of contents with known dynamic extensions was 33.8%. The PHP: Hypertext Preprocessor technology was the most prevalent presenting 48.3% of the dynamic contents. The total percentage of dynamically generated contents identified through both methods was 46.3%.

4.4 Duplication

Despite the hypertextual capacities of the Web to reference and reuse contents without performing physical duplication, contents available on the Web are not unique. Duplicates occur when the same content is referenced by distinct URLs. For instance, when contents are repeated in different directories of a site, physically duplicated across sites, or as it happens, for instance, with mirror sites.

During the crawl, a SHA1 digest was generated for each content and recorded in the crawl log. This digest was used to measure content duplication. Measuring duplication is useful to help choosing adequate storage systems according to their duplicates elimination features [9]. Approximately 48.7 million downloaded contents were crawled for 40 million different digests, which means that 17.7% of the downloaded contents were referenced by distinct URLs, representing 15.2% of the total amount of data downloaded.

Table 6 presents duplication distribution across media types. The 2^{nd} column presents the prevalence of each media type within the total set of duplicates. The 3^{rd} column presents the prevalence of duplicates within each media type. The 4^{th} column presents the percentage of each media type duplicates that were found within the same site. The objective of this analysis was to identify which media types contributed with the larger amount of duplicates and if there were media types more prone to be duplicated. The obtained results can be used to tune

Media type	% total duplicates	% dup within media type	% dup within same site
text/html	38.1%	11.4%	72.9%
image/gif	23.3%	42.8%	44.4%
image/jpeg	19.4%	14.7%	46.6%
image/png	3.3%	44.9%	42.6%
text/plain	2.6%	45.0%	29.6%
app'n/pdf	2.5%	22.7%	20.1%
text/css	2.3%	58.4%	57.9%
app'n/x-shockwave-flash	2.0%	46.9%	86.4%
text/xml	1.7%	15.8%	71.0%
app'n/x-javascript	1.5%	57.6%	38.1%
Other	3.4%	28.8%	-

Table 6. Distribution of media type prevalence regarding the total set of duplicates, the percentage of duplicates within each media type and the percentage of duplicates from each media type within the same site.

duplicates detection mechanisms according to the media type of the contents. Duplication per media type is useful to define strategies for identifying duplicates within a Web data repository or proxy because it allows to identify the media types with higher probability of being duplicated or that generate higher volume of duplicated information.

The obtained results show that the media types that contributed with the larger number of duplicates are also the most common on the Web. However, there is not a direct relation between the prevalence percentages for duplicates and contents. For instance, GIF images are a stronger responsible for the amount of duplicates on the Web than other media types because they represent 23.3% of the duplicates but only 9.4% of the contents available on the Web. Empirically, this result was not surprising because image icons are commonly copied across sites. On the other hand, HTML contents are less prone to be duplicated because only a small edition on their text or layout originates a new digest. The obtained results also show that *text/css* and *application/x-javascript* presented the highest levels of duplication with 58.4% and 57.6%, respectively. We found these results surprising, so we randomly analyzed 20 *text/css* files and noticed that only 2 of them were unique within the crawl and 16 were repeated within the same site or on its subdomains. According to Web design best practices [13], the creation of independent files containing CSS or JavaScript code that are shared by different pages enables code reuse which facilitates maintenance tasks, reduces pages size and enables browsers to keep files locally on cache, without having to download them to render each page. However, the obtained results show that CSS and JavaScript contents are commonly duplicated instead of being reused, presenting 57.9% and 38.1% of duplicates within the same site, inhibiting the presented advantages.

Based on the obtained results, we conclude that some media types are more prone to be duplicated than others.

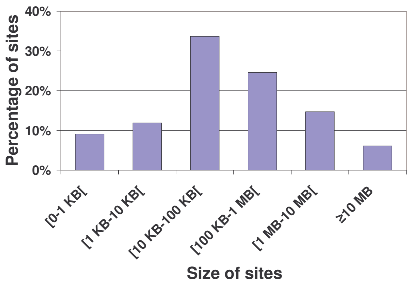


Fig. 4. Distribution of the total amount of data downloaded per site (avg: 6.8 MB, median: 71 KB).

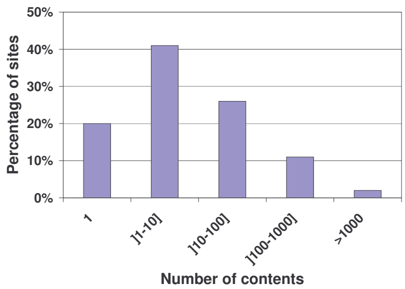


Fig. 5. Distribution of the number of contents downloaded per site (avg: 134.9, median: 5).

5 Sites

A site was considered as being valid if it returned a 200 response code to at least one request. The total number of sites visited was 484 398 and 74.6% of them were valid. In the crawl, 125 393 sites had the file *robots.txt*, 34.7% of the total number of sites that returned a 200 response code to at least one request.

5.1 Site size

The number of contents per site influences the crawler's data partitioning of its queues of URLs to visit [6]. Figure 4 presents the distribution of the total amount of data downloaded per site and shows that 85% of the sites provided between 1 KB and 10 MB of data. We analyzed the top 10 sites that served the largest amount of data and they were data repositories, such as FTP sites, video or Podcast archives.

Figure 5 presents the distribution of contents crawled per site. The obtained results show that sites are typically small, 87% presented less than 100 contents. Only 2% of the total sites contained more than 1 000 contents. Two relatively large sites were found, one with 280 609 contents and another with 557 978 contents. They belonged to two blog platforms.

5.2 Successful responses

The percentage of successful responses returned by a site is an indicator of its quality. A site that presents a large percentage of broken links or errors mines the trust of its users.

Figure 6 presents the distribution of the percentage of sites across increasing thresholds of successful responses. Redirects are used to maintain links to contents that changed their addresses. The obtained results show that only 18% of the visited sites returned OK responses to all of the requests but this percentage significantly increased to 39% when considering Successful and Redirection

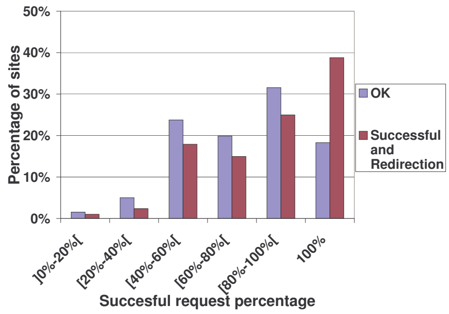


Fig. 6. Percentage of sites per successful request percentage. The left series regards the OK responses and the right series regards the Successful and Redirection responses.

responses. This increase might be due to the existence of sites containing just one URL which is a redirect. For instance, to redirect an old site domain to a new one. Note that these single-URL sites present 100% successful responses. The obtained results show that if these sites were excluded, the percentage of sites presenting 100% successful responses would drop from 39% to 30.8%.

One may think that larger sites would be harder to maintain and should present a higher rate of broken links. However, the correlation factor between site size and OK responses was 0.06 and 0.04 for Successful and Redirection responses. This shows that there is no relation between site size and successful response percentage.

5.3 Sites hosted per IP address

Virtual hosts enable a single Web server to host several sites. We assumed that each server is identified by an IP address.

The crawler logs do not include the IP address of the visited sites. Therefore, we made a post-crawl resolution of the site names after the crawl had finished. We were unable to resolve the name for 1.3% of the sites that were online during the crawl. Regarding the distribution of sites hosted per IP address, the obtained results show that 75% of the IP addresses host 1 site. Only 2% of the IP addresses host more than 10 sites. On average, each IP address hosts 4 sites (median of 1). We have found two IP addresses hosting over 30 000 sites each, both belonging to SAPO, a Portuguese Internet and Web Service Provider.

Measuring the distribution of sites across IP addresses is useful to define politeness policies for crawling. For instance, a crawler must be set to respect a courtesy pause between requests to the same IP address or to the same site, to avoid server overload. The obtained results show that, in general, crawling courtesy pauses based on site name are adequate because most servers host a single site.

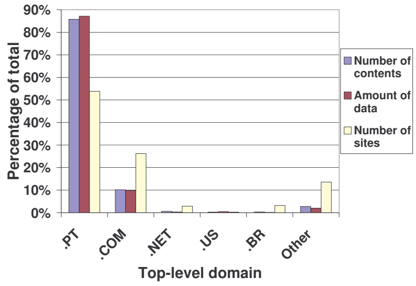


Fig. 7. Distribution of the number of contents, amount of data and sites per top-level domain.

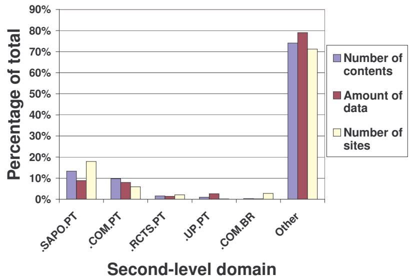


Fig. 8. Distribution of the number of contents, amount of data and sites per second-level domain.

6 Domains

A national Web is not limited to the corresponding country code top-level domain. Archiving a national Web disregarding other top-level domains prevents pages from being displayed correctly when those pages have embedded contents hosted in other domains.

Figure 7 presents the number of contents, amount of data and sites per top-level domain. The .PT domain was the most representative top-level domain, with 85.8% of the contents downloaded, 87% of the amount of data downloaded and 53.9% of the valid sites visited.

Figure 8 presents the number of contents, amount of data and sites per second-level domain. The .SAPO.PT domain is private but appears before the .COM.PT domain which is open for public registration. A possible explanation for this fact is that SAPO is one of the largest communication companies in Portugal with a network that hosts several popular radio and newspaper sites. It is also an Internet Service Provider, hosting the personal Web pages from its users, has a blog platform and was the first Portuguese Web portal. Higher education institutions also play a significant role on the Portuguese Web, representing a total of 7.9% of the number of contents and 12.6% of the total amount of data downloaded. These results were obtained through a list of domains of higher education institutions [3].

7 Conclusions and future work

This study presented an updated characterization of the Portuguese Web derived from a crawl performed in 2008. The Portuguese Web contents are referenced by long URL addresses that contain few directories within them. One explanation for this fact is that approximately half of the Portuguese contents are dynamically generated, using mainly the PHP technology. The prevalent media types are HTML pages, JPEG and GIF images, representing 90% of the contents.

However, if we measure the total amount of data provided per each media type, the dominant formats are HTML pages, PDF documents and JPEG images, representing 69.4% of the total amount of data crawled. The content size and duplication rate vary according to media type. In general, 99% of the contents are smaller than 512 KB and 17.7% of the contents are referenced by distinct URLs, which frequently occur on the same site. Prevalent media types are responsible for most of the duplicates. However, some media types that present relatively small presence on the Web are very prone to be duplicated. An unexpected result was that duplication is prevalent among CSS and JavaScript files, which are files that are supposed to be reused across pages.

Sites are typically small and the large majority of their contents are found at shallow levels of physical depth (measured by the number of slashes within the URL), which may compromise relevance heuristics based on URL depth. Half of the sites presented a successful response rate bellow 80%. Most IP addresses host a single site and only 2% host more than 10 sites.

Although most sites are referenced by their own second-level domain, the obtained results show that a domain belonging to a private communication company hosts an important share of the Portuguese Web under their sub-domains. Over 10% of the Portuguese Web contents were hosted under .COM because they were embedded or redirected from contents hosted under the .PT domain.

The crawl log that was used in this study is available at `http://arquivo.pt/resources` for research purposes. Future work will involve gathering statistics extracted from content analysis over different metrics: accessibility for people with disabilities, respect for format specifications and link structure analysis.

References

1. R. Baeza-Yates, C. Castillo, and E. Efthimiadis. Characterization of national web domains. *ACM Transactions on Internet Technology*, 7(2), 2007.
2. C. Castillo. *Effective Web Crawling.* PhD thesis, University of Chile, November 2004.
3. Direcção-Geral do Ensino Superior (Directorate-General for Higher Education) - Home Page. http://www.dges.mctes.pt/DGES, January 2009.
4. R. Fagin, R. Kumar, K. Mccurley, J. Novak, D. Sivakumar, J. Tomlin, and D. Williamson. Searching the workplace web, 2003.
5. D. H.-L. Goh and P. K. Ng. Link decay in leading information science journals. *J. Am. Soc. Inf. Sci. Technol.*, 58(1):15–24, 2007.
6. D. Gomes. *Web Modelling for Web Warehouse Design.* PhD thesis, University of Lisbon, March 2007.
7. D. Gomes, S. Freitas, and M. J. Silva. Design and selection criteria for a national web archive. In *ECDL 2006 - 10th European Conference on Research and Advanced Technology for Digital Libraries*, number 4172/2006 in LNCS, pages 196–207. Springer-Verlag, September 2006.
8. D. Gomes, A. Nogueira, J. Miranda, and M. Costa. Introducing the Portuguese web archive initiative. In *8th International Web Archiving Workshop (IWAW08)*, Aarhus, Denmark, September 2008.

9. D. Gomes, A. L. Santos, and M. J. Silva. Managing duplicates in a web archive. In L. M. Liebrock, editor, *Proceedings of the 21th Annual ACM Symposium on Applied Computing (ACM-SAC-06)*, Dijon, France, April 2006.
10. D. Gomes and M. J. Silva. Characterizing a national community web. *ACM Transactions on Internet Technology*, 5(3):508–531, 2005.
11. A. Heydon and M. Najork. Mercator: A scalable, extensible web crawler. *World Wide Web*, 2(4):219–229, 1999.
12. M. Koster. A standard for robot exclusion. `http://www.robotstxt.org/wc/norobots.html`, June 1994.
13. S. Koyani, R. Bailey, and J. Nall. *Research-Based Web Design & Usability Guidelines*. Department of Health and Human Services, 2006.
14. W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proc. of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34, 2002.
15. F. Lasfargues, C. Oury, and B. Wendland. Legal deposit of the French Web: harvesting strategies for a national domain. In *8th International Web Archiving Workshop (IWAW08)*, Aarhus, Denmark, September 2008.
16. M. Modesto, Álvaro R. Pereira Jr., N. Ziviani, C. Castillo, and R. Baeza-Yales. Um novo retrato da web brasileira. In *XXXII SEMISH - Anais do Seminário Integrado de Software e Hardware*, pages 2005–2017, São Leopoldo, RS, July 2005.
17. G. Mohr, M. Kimpton, M. Stack, and I. Ranitovic. Introduction to heritrix, an archival quality web crawler. In *4th International Web Archiving Workshop (IWAW04)*, Bath, UK, September 2004.
18. M. J. Nicolau, J. Macedo, and A. Costa. Caracterização da informação WWW na RCCN. Technical report, Universidade do Minho, 1997.
19. N. Noronha, J. P. Campos, D. Gomes, M. J. Silva, and J. Borbinha. A deposit for digital collections. In P. Constantopoulos and I. T. Sølvberg, editors, *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries, ECDL*, volume 2163 of *LNCS*, pages 200–212. Springer, 2001.
20. E. A. Veloso, E. de Moura, P. Golgher, A. da Silva, R. Almeida, A. Laender, B. Ribeiro-Neto, and N. Ziviani. Um retrato da web brasileira. In *Simpósio Brasileiro de Computação*, Curitiba, Brasil, July 2000.
21. P. Zabicka. Czech Web archive analysis. *New Review of Hypermedia and Multimedia*, 13(1):27–37, 2007.

Improving Web User Experience with Document Activity Sparklines

Sérgio Nunes[1], Cristina Ribeiro[1,2] and Gabriel David[1,2]

[1] Departamento de Engenharia Informática
Faculdade de Engenharia da Universidade do Porto
[2] INESC-Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto PORTUGAL
{sergio.nunes,mcr,gtd}@fe.up.pt

Abstract. The temporal dimension of the web has been mostly ignored when designing user interfaces for both searching and document browsing. The dynamic nature of the web is invisible to the typical user despite the fact that most web documents and sites change at a very rapid pace. In this paper we present and describe visual extensions applied to both individual web sites and search engines, that capture content activity over time. This idea is implemented with a prototype that processes publicly available web feeds to generate activity profiles and enhances selected web sites using sparklines. These proposals bring a temporal perspective to the user's browsing experience.

1 Introduction

Previous research on web dynamics has shown that web documents change at a very fast pace. Ntoulas et al. [4] have analyzed both content and link structure evolution of web pages and found that, after one year, 50% of the content on the web is new. Temporal information about web documents can be found either as document-based or as web-based evidence. A detailed survey about possible sources of temporal information on the web is presented in Nunes [5].

The use of temporal information in end-user interfaces has already been addressed. In the context of personal information retrieval Dumais et al. [3] found that time is an important information retrieval cue. In the field of web search, Alonso et al. [2] have used timelines to improve the functionality of search applications. They describe a prototype based on the DBLP bibliography collection. Our work addresses the same problem, proposing a different approach to bring temporal data to the web user's interface. Also, we present a prototype that works with live web data.

We have developed interface enhancements that use publicly available web feeds to present *content activity sparklines*. We leverage on the idea of sparklines, a type of information-rich graphics proposed by Tufte [6]. In the following sections we describe two types of UI improvements — website activity profiles and search engine results enhancements.

2 Activity Profiles

Web feeds are XML documents that provide information about content updates. The Atom Syndication Format is a popular IETF standard for publishing web feeds [1]. Typically, web feeds are structured in items containing a timestamp. Using the timestamp information we are able to aggregate items by day to produce an *activity profile*. Figure 1 shows the activity profile of a blog as a sparkline on the browser's top left corner. Even though this profile is only based on 13 updates, it captures very clearly the blogger's recent activity. Also highlighted in the figure is the additional info available as a tooltip when the mouse hovers the image.

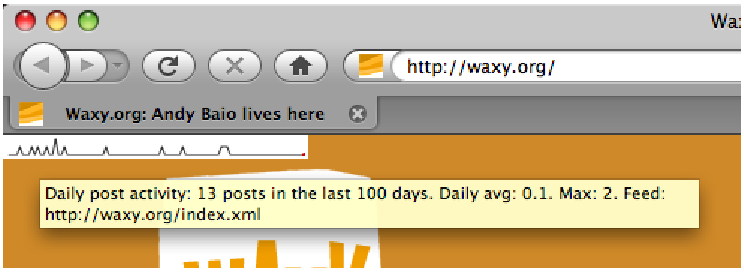


Fig. 1. A blog's activity profile with tooltip.

After conducting an informal survey over a large number of public websites, we found that activity profiles for blogs and news feeds are intuitive and represent an interesting extension to the standard UI. These client-side extensions bring a temporal perspective to the user's browsing experience.

3 Enhancing Search Results

Following upon the idea of using web feeds as a source of data for producing activity profiles, we apply the same idea to current search engine interfaces. We developed extensions for search engines focused in news and blogs, since these typically include public web feeds. Figure 2 shows a set of live results from the Icerocket Blog Search[3] with activity sparklines presented next to each result. We chose to use sparklines due to their high data density and small size, resulting in a low visual impact in the typical web search results layout.

This interface element informs the end-user about the recent activity of each result item. For instance, it is very easy to spot inactive blogs on a long list of search results. On the other hand, it is easy to identify activity peaks related to specific subjects.

[3] `http://www.icerocket.com/`

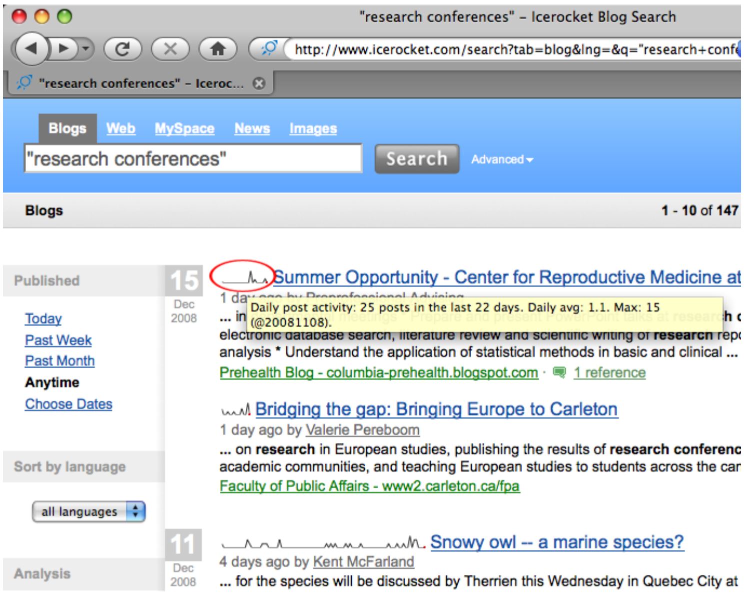


Fig. 2. Icerocket Blog Search with sparklines.

4 Prototype Development

The prototypes were developed using Greasemonkey[4], a Javascript client-side extension to the Firefox web browser. Our Greasemonkey scripts are activated each time a web page is loaded. If available, public web feeds are automatically discovered and parsed. This data is then used to produce a daily frequency count of the feed's activity. Finally, sparklines are drawn in real-time using the Google Chart API[5] and embedded in the document's layout.

As shown in the previous screenshots, a tooltip is added to each sparkline containing additional information about the feed's activity. For each feed, the tooltip provides details on the total number of posts, the total number of days covered, the daily average and maximum value, along with the day of highest activity. While the activity profile extension works on every site that publishes a web feed, the search results extension only works on selected search engines (e.g. Google Blog Search, IceRocket, Technorati).

All prototypes are available online at `http://irlab.fe.up.pt/p/sparkfeeds`. These are fully functional prototypes that were successfully used as a proof of concept in real-world scenarios. The main challenges faced during the development of these extensions were the parsing of dates (due to the plethora of possible "standards") and the parsing of web documents for embedding the sparklines.

[4] `http://www.greasespot.net/`
[5] `http://code.google.com/apis/chart/`

5 Conclusions

Currently available web feeds can be used as a source of temporal information to improve end-user web interfaces. We describe UI enhancements for both individual sites and search engine results. Also, we present fully functional prototypes based on client-side extensions. Given the strong temporal nature of the web, we feel that exposing temporal information to the final user presents several opportunities for UI developments. Google has recently launched a public experimental interface that presents results on a timeline[6].

Preliminary ad-hoc tests conducted with a small number of users resulted in very positive feedback. This was particularly evident in open problems, where the exploratory nature of the task is clearer. We plan to design and conduct users studies as future work. For instance, we plan to evaluate if these enhancements result in significant improvements when users try to complete specific tasks.

6 Acknowledgments

This work was supported in part by Fundação para a Ciência e a Tecnologia (FCT) and Fundo Social Europeu (FSE - III Quadro Comunitário de Apoio), under grant SFRH/BD/31043/2006.

References

1. The Atom syndication format. Technical report, Internet Engineering Task Force, December 2005.
2. O. Alonso, R. Baeza-Yates, and M. Gertz. Exploratory search using timelines. In *SIGCHI 2007 Workshop on Exploratory Search and HCI Workshop*, 2007.
3. S. Dumais, E. Cutrell, J. J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff I've seen: a system for personal information retrieval and re-use. In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 72–79. ACM Press, 2003.
4. A. Ntoulas, J. Cho, and C. Olston. What's new on the web?: the evolution of the web from a search engine perspective. In *WWW '04: Proceedings of the 13th International World Wide Web Conference*, pages 1–12, New York, NY, USA, 2004. ACM Press.
5. S. Nunes. Exploring temporal evidence in web information retrieval. In A. MacFarlane, L. Azzopardi, and I. Ounis, editors, *BCS IRSG Symposium Future Directions in Information Access (FDIA 2007)*, pages 44–50. BCS IRSG, BCS IRSG, August 2007.
6. E. R. Tufte. *Beautiful Evidence.* Graphics Press, 2006.

[6] `http://www.google.com/experimental/`

Author Index

Abreu, Ricardo 391
Abreu, Salvador 139
Aires, José 513, S-587
Albisua, Iñaki 329
Alferes, José Júlio S-101
Almeida, J. J. S-400
Almeida, Leonardo Jesus 579
Almeida, Rui S-239
Alves, Victor 341
Analide, Cesar 402
Anastácio, Ivo S-598
Andrade, Pedro Ribeiro de S-535
Anh, Han The 115, S-138
Antunes, Luis S-508, S-547
Arbelaitz, Olatz 329
Areias, Miguel S-113
Azevedo, Francisco S-213
Azevedo, João S-251
Azevedo, José Luís S-275, S-350

Balsa, João S-508, S-547
Barahona, Pedro S-86, S-201
Barata, Fábio S-53
Barbosa, Luís 181
Barreiro, Anabela S-79
Bedregal, B.C. 451
Bessa Jr, José Elievam S-3
Bocsi, Rajmund S-449
Braga, Igor Assis 489
Branco, António 501
Bruns, Ralf S-27
Bulas-Cruz, José 155

Cachulo, Nuno S-386
Calado, Pável S-598
Câmara, Gilberto S-535
Cardoso, Amílcar 167
Carmo, Maria Beatriz 391
Carvalho, Paula 565
Castro Júnior, Alberto Nogueira de 217
Castro, António J. M. 3, 181, S-461
Castro, Filipe 279
Cavique, Luís S-363
Certo, João 231
Chao, Cástor Sánchez 101
Chauhan, Aneesh S-263
Chen, Ning S-374
Cláudio, Ana Paula 391
Coelho, Helder 413, 439, S-497, S-547
Correia, Luís 53, 65, S-41
Correia, Marco S-201
Corrente, Gustavo S-299, S-323
Corruble, Vincent 379
Cortez, Paulo S-386
Costa, Antônio Carlos da Rocha 367, 439, 451, 475
Costa, Paulo 243, 300
Costa, Raúl S-386
Cruz, J.L. Pérez de la 205
Cunha, André Luiz S-3
Cunha, Bernardo S-275
Cunha, João S-275

Damásio, Carlos Viegas 115
Dastani, Mehdi S-174
David, Gabriel 601
Deconinck, Geert 15
Dhar, Joydip 87
Dias, Gaël S-610, S-622
Dimuro, Graçaliz Pereira 367, 451
Domingues, Marcos Aurélio 315
Driessens, Kurt 15
Drury, Brett S-400
Duarte, João S-374
Duarte, Pedro S-473
Dunkel, Jürgen S-27

Escudeiro, Nuno Filipe S-411
Esteves, Edgar F. S-15

Fachada, Nuno S-53
Ferreira, Marcos 243
Ferreira Jr, Paulo S-508
Figueira, Dario S-287
Figueiredo, Lino S-187
Figueiredo, Nuno M. S-299
Filipe, Nelson S-323
Formella, Arno 101
França, Robson dos Santos 463

Gabaldon, Alfredo S-101

Gama, João 353, S-423
Gaspar, Graça 391
Gaudio, Rosa Del 501
Gomes, Daniel 589
Gomes, Luís 513, S-587
Gomes, Paulo 127, 541
Gonçalves, João F. B. S-15
Gonçalves, Nelson S-310
Gonçalves, Teresa 537
Gouveia, Manuel 243
Grilo, Carlos 65, S-41
Gulyás, László S-449, S-572
Gurrutxaga, Ibai 329

Hernández, Laritza 525
Herpson, Cédric 379
Holvoet, Tom 15
Huang, Chung-Yuan 425

Isidoro, Carlos S-53

Jeannes, Fernanda Mendez 475
Jorge, Alípio Mário S-239, S-411, S-673

Klakow, Dietrich S-658
Kosina, Petr S-423

Labeeuw, Wouter 15
Lambov, Dinko S-622
Lau, Nuno S-275, S-299, S-323, S-350
Legéndi, Richárd S-449
Leite, João S-101
Lira, Tiago 267
Lobo, Victor J. A. S. 77
Lopes, Alneu de Andrade 579
Lopes, Gabriel Pereira 513, S-587, S-646
Lopes, Manuel S-287
Lopes Cardoso, Henrique S-560
López-Lopez, A. 525
Lynce, Inês 193

Machuca, E. 205
Magalhães, Pedro 565
Máhr, Tamás S-449
Malheiros, Paulo 300
Mandow, L. 205
Manso, António 53
Mariano, Pedro 65, S-485
Marietto, Maria das Graças Bruno 463
Marques, Nuno C. 319
Marques-Silva, Joao 193
Marreiros, Goreti 155, S-187
Martín, José I. 329
Martins, Bruno S-598
Martins, Paulo Triunfante 77
Matos, Sérgio S-79
Matsubara, Edson Takashi 489
Medina, José E. 525
Melo, Francisco S. 254
Mendes, Armando B. S-363
Meyer, John-Jules Ch. S-174
Mingote, Luís S-213
Miranda, Catarina S-673
Miranda, João 589
Miranda, Nuno 537
Monard, Maria Carolina 489
Moniz, Luís 391
Monteiro, Antonio Miguel Vieira S-535
Moraliyski, Rumen S-610
Moreira, António Paulo 243, 300
Mota, António 3, S-461
Moura Pires, João Carlos 139
Muggleton, Stephen S-150
Muguerza, Javier 329

Nascimento, Amanda S-263
Neves, António J. R. S-299, S-350
Neves, João C. S-374
Neves, José 155
Noncheva, Veska S-622
Nozawa, Erika Handa 217
Nunes, Davide S-547
Nunes, Sérgio 601

Oliveira, António 167
Oliveira, Elaine Harada Teixeira de 217
Oliveira, Eugénio 3, 565, S-15, S-520, S-560
Oliveira, Gerardo 267
Oliveira, Hugo Gonçalo 541
Oliveira, João Lobato 231
Oliveira, José Luis S-79
Oliveira, Miguel S-251
Oliveira, Pedro 127
Ortony, Andrew S-163
Ossowski, Sascha 39, S-27
Otero, Pablo Gamallo S-634

Pacheco, Pedro S-251
Passos, Lúcio Sanchez 27

Pawlowski, Oliver S-27
Peixoto, Hugo 279
Pequeno, Valéria 139
Pereira, António S-473
Pereira, Artur S-299
Pereira, Luís Moniz S-138
Pereira, Rui S-225, S-338
Pereira, Welma S-435
Pérez, Jesús M. 329
Perona, Iñigo 329
Perriquet, Olivier S-86
Pinheiro, Vitor 341
Portela, João 279
Pöschel, Thorsten 101

Quaresma, Paulo 537
Quek, Boon-Kiat S-163

Rajoria, Sandeep 87
Raminhos, Ricardo 537
Ramli, Carroline Kencana 115
Ramos, Carlos 155, S-187
Reis, Luís Paulo 3, 181, 231, 243, 267, 279, 288, S-239, S-251, S-461, S-473
Rezende, Solange Oliveira 315
Ribeiro, Bernardete S-374
Ribeiro, Cristina 601
Ribeiro, João S-485
Rocha, Ana Paula S-520
Rocha, Ricardo S-113
Rodrigues, Fátima 553
Rodrigues, João S-275, S-350
Rosa, Agostinho S-53
Rosas, Eugénio 402
Rossetti, Rosaldo J. F. 27, S-15
Ruesch, Jonas S-287

Saias, José 537
Sakamoto, Kayo S-163
Santiago, Catarina 300
Santos, A.V. 451
Santos, António Paulo 553
Santos, Diana 541
Santos, Elder Rizzon 413
Santos, Jorge M. A. S-363
Santos, José C. A. S-150
Santos, Ricardo 155
Santos Costa, Vítor S-126
Sarmento, Luís 565
Seabra, Pedro 537
Seabra Lopes, Luís S-225, S-263, S-323, S-338, S-485
Sebastião, Raquel 353
Sequeira, João S-310
Setti, José Reynaldo S-3
Silva, Augusto S-338
Silva, João S-350
Silva, Joaquim Ferreira da S-646
Silva, Mário J. 565
Silva, Ricardo 267
Silva, Sara S-65
Soares, Carlos 87, S-435
Sousa, Armando 288, 300
Sousa, Jorge Pinho de 87
Steinberger, Margarethe Born 463
Steunebrink, Bas R. S-174
Sun, Chuen-Tsai 425
Szabó, Attila S-572

Tamaddoni-Nezhad, Alireza S-150
Teixeira, Rui 279
Torgo, Luis S-435
Tóth, István János S-572
Trigo, Paulo S-497

Urbano, Joana S-520
Urbano, Paulo S-508, S-547

Vanneschi, Leonardo S-65
Vasirani, Matteo 39
Veloso, Manuela 254
Ventura, Francisco 167
Ventura, Rodrigo S-287
Vicari, Rosa Maria 217, 413
Vieira, Armando S. S-374
Vinagre, Eugénia S-187

Wang, Sheng-Wen 425
Werneck, Bruno S-263
Weyns, Danny 15
Wiegand, Michael S-658