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.

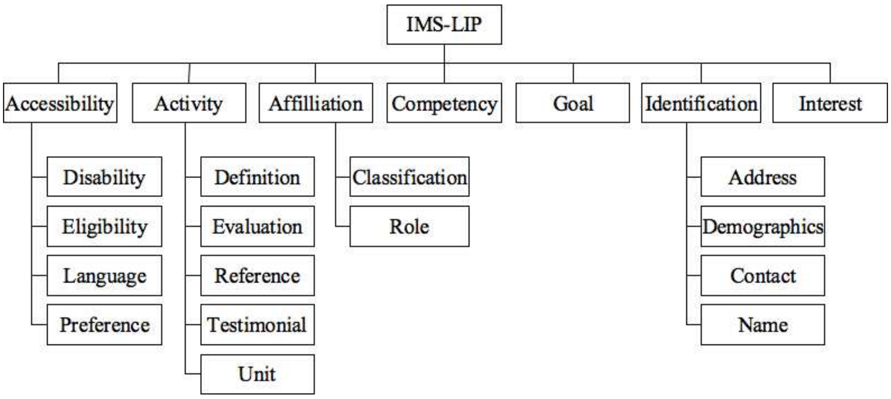

Figure 1. Hierarchy of concepts from the agent's domain ontology (IMS-LIP).

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.).

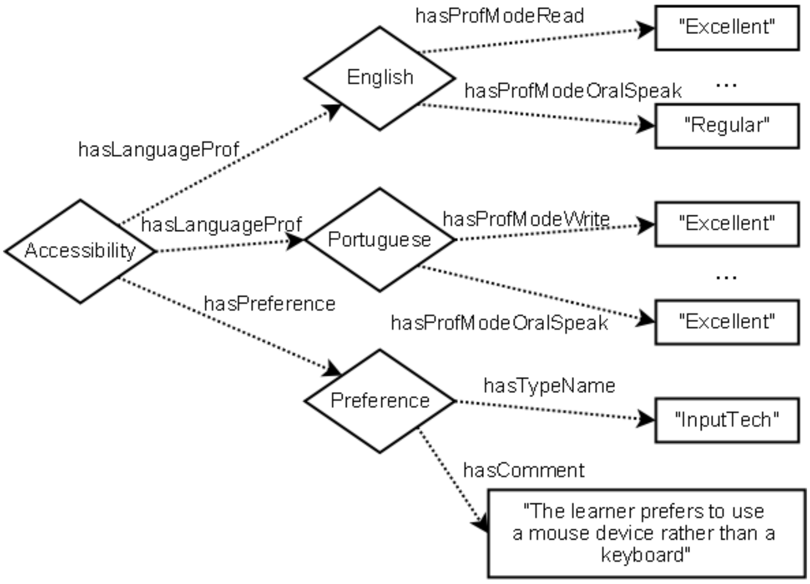

Figure 2. Individuals representing accessibility details.

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.

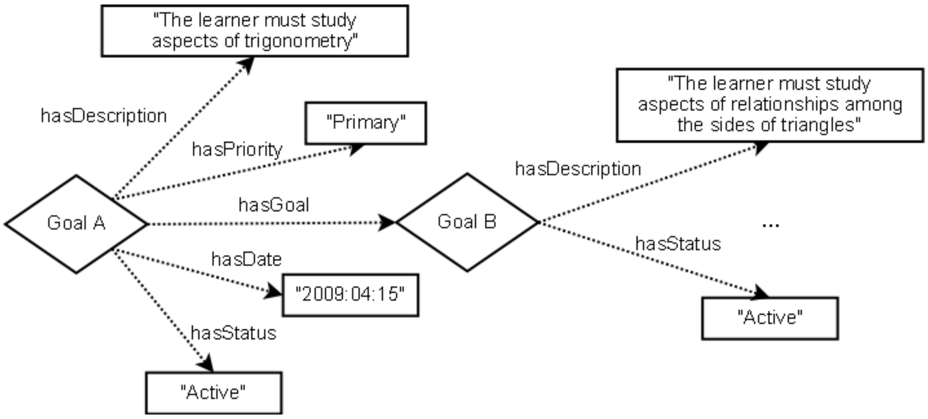
"The learner must study aspects of trigonometry"
hasDescription
"Primary"
hasPriority
"The learner must study aspects of relationships among the sides of triangles"
hasDescription
Goal A
hasGoal
Goal B
...
hasStatus
hasDate
"2009:04:15"
"Active"
hasStatus
"Active"

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.