DOMAIN ONTOLOGY GENERATION USING LMF STANDARDIZED DICTIONARY STRUCTURE

Feten Baccar Ben Amar, Bilel Gargouri
MIRACL Laboratory, University of Sfax, FSEGS, B.P. 1088, 3018, Sfax, Tunisia

Abdelmajid Ben Hamadou
MIRACL Laboratory, University of Sfax, ISIMS, B.P. 242, 3021 Sakiet-Ezzit, Tunisia

Keywords: Core Domain Ontology Generation, LMF Standardized Dictionary, Ontology Quality, Arabic Language.

Abstract: The present paper proposes a methodology for generating core domain ontology from LMF standardized dictionary (ISO-24613). It consists in deriving the ontological entities systematically from the explicit information, taking advantage of the LMF dictionary structure. Indeed, such finely-structured source incorporates multi-domain lexical knowledge of morphological, syntactic and semantic levels, lending itself to ontological interpretations. The basic feature of the proposed methodology lies in the proper building of ontologies. To this end, we have integrated a validation stage into the suggested process in order to maintain the coherence of the resulting formalized ontology core during this process. Furthermore, this methodology has been implemented in a rule-based system, whose high-performance is shown through an experiment carried out on the Arabic language. This choice is explained not only by the great deficiency of work on Arabic ontology building, but also by the availability within our research team of an LMF standardized Arabic dictionary.

1 INTRODUCTION

Domain ontologies are "engineering artifacts" describing a set of relevant domain-specific concepts and their relationships in a formal way. Although the area of ontology learning aiming to automate the ontology creation process has been dealt with by plenty of work, it is still a long way from being fully automatic and deployable on a large scale (Cimiano et al., 2009). This is essentially because it is a time-consuming and difficult task that requires significant human involvement for the validation of each step throughout this process.

In order to reduce the costs, research on (semi-) automatic ontology building from scratch has been conducted using a variety of resources, such as raw texts (Aussenac-Gilles et al., 2008), Machine-Readable Dictionaries (MRDs) (Kurematsu et al., 2004; Rigau et al., 1998), and thesauri (Chrisment et al., 2008). Obviously, these resources have different features, and therefore, each proposed process is based on a different approach with respect to principles, design criteria, NLP techniques, etc.

As linguistic information is increasingly required in ontologies mainly in NLP applications (Buitelaar et al., 2009), among the considered terminological resources, MRDs represent one of the most likely and suitable sources promoting the knowledge extraction both at ontological and lexical levels. However, since much information has not yet been encoded, the access to the potential wealth of information in dictionaries remains limited to software applications.

Recently, Lexical Markup Framework (LMF) (ISO 24613, 2008), which is a standard for the representation and construction of lexical resources, has been defined. Basically, its meta-model provides a common and shared representation of lexical objects that allows the encoding of rich linguistic information, including morphological, syntactic, and semantic aspects (Francopoulo and George, 2008).

It is in this context that we have proposed a new approach that makes use of LMF standardized dictionaries to generate domain ontologies (Baccar

Baccar Ben Amar F., Gargouri B. and Ben Hamadou A..
DOMAIN ONTOLOGY GENERATION USING LMF STANDARDIZED DICTIONARY STRUCTURE.
DOI: 10.5220/0003512803960401
In *Proceedings of the 6th International Conference on Software and Database Technologies* (ICSOFT-2011), pages 396-401
ISBN: 978-989-8425-77-5

et al., 2010). Such resource incorporates widely-accepted and commonly-referenced diversified linguistic knowledge lending itself to ontological interpretations. Indeed, finely-structured knowledge in an LMF-standardized dictionary paves the way for the constitution of core domain ontology. Furthermore, the abundance of texts available in the definitions, explanations and examples are very interesting to realize the core enrichment and above all to provide the ontology elements with linguistic grounding.

On the other hand, the nature of ontologies as reference models for a domain requires a high degree of quality of the respective model. Indeed, several approaches have been considered in literature in order to assess ontology construction methodologies. However, a comprehensive and consensual standard methodology seems to be out of reach (Almeida, 2009). Yet, evaluating the ontology as a whole is a costly and challenging task especially when the reduction of human intervention is sought. This can be deemed as a major impediment that may elucidate the ontologies' failure not only to be reused in others but also to be exploited in final applications.

The ultimate objective of this paper is to show the way in which an LMF-compliant MRD is exploited for the core domain ontology generation. In fact, its systematic organization allowed us to implement a fully automatic and iterative process for a direct dictionary transformation of some particular information into ontological elements. Additionally, the suggested process includes a validation phase in order to preserve the quality of the produced ontology throughout its development life cycle. Furthermore, the proposed methodology has been implemented in a rule-based system, whose high-performance has proven to be trustworthy through an experiment carried out on an Arabic dictionary (Baccar et al., 2008).

The remainder of this paper is structured as follows: Section 2 gives a related work overview of ontology construction based on (semi-) structured resources. Section 3 presents the proposed methodology for generating the core domain ontologies from LMF standardized dictionaries. Section 4 presents the details of implementation. As for Section 5, it is devoted to describe our experimentation as well as to discuss the results quality. Finally, Section 6 concludes the paper with opening perspectives for future work.

2 RELATED WORK

Since ontology engineering has long been a tedious task requiring considerable human involvement and effort, many proposals have been suggested to facilitate knowledge domain acquisition. It is also widely recognized that taking into account relevant resources from the very beginning of the ontology development process yields more effective results. Accordingly, recent approaches based on a variety of structured or semi-structured resources, such as XML documents (Aussenac-Gilles and Kamel, 2009), UML models (Na et al., 2006) and so on, have been proposed with the aim of producing an early stage of domain ontology through rule-based learning techniques. The resulting primary ontologies, also named core or kernel ontologies, can help in the quick modeling of the domain knowledge and could be further extended to obtain a complete ontology.

Considered as a large repository of quasi-structured knowledge about language and the world, MRDs illustrate the building stone for the generation of conceptual structures ranging from concept hierarchies, thesauri to ontologies (Jannink, 1999). Indeed, as noted in (Hirst, 2009), word senses can be seen as the equivalent of ontological categories, and lexical relations (e.g., *synonymy*, *antonymy*, *hyponymy*, *meronymy* and so on) would correspond to ontological relations (for example, *hypernymy* would stand for *subsumption*). Nevertheless, since the MRDs are oriented towards human reader, much information is not well-structured and therefore its machine interpretation might not be evident. Consequently, systems relying on MRDs incorporate two major problems, one of which is their need for massive human intervention and the other is their confinement to limited relations, in almost all cases, the taxonomic ones.

From another standpoint, being a newly emerging standard for the creation and use of computational lexicons, LMF has recently been defined (Francopoulo and Georges, 2008). Its meta-model allows the representation of NLP and MRD lexicons in a systematic organization. Indeed, such model contains much explicit linguistic information as well as a lot of implicit information included in the definitions and examples.

After the introduction of LMF standard, a good deal of active work, among which we can mention LexInfo (Buitelaar et al., 2009), LIR (Montiel-Pensoda et al., 2008) and (Pazienza and Stellato, 2006) models, has been undertaken in response to the need of increasing the linguistic expressivity of

given ontologies (Buitelaar et al., 2009). The proposed models try to associate lexical information with ontological entities, which is a heavy and time consuming activity considering the plurality and the heterogeneity of the resort sources. In addition, some complexity rises when linguistic information is involved in ontology reasoning (Ma et al., 2010).

3 METHODOLOGY FOR CORE DOMAIN ONTOLOGY GENERATION

3.1 Basic Idea

Thanks to its encompassing of both ontological and lexical information, an LMF standardized dictionary offers a very suitable primary knowledge resource to learn domain ontologies (Baccar et al., 2010). Accordingly, we have proposed an approach consisting in firstly building the core of the target ontology, taking advantage of the LMF standardized dictionary structure. Secondly, it consists in enriching such core starting from textual sources with guided semantic fields available in the definitions and the examples of lexical entries.

Within our context, the core provides all possible sets of basic objects in a specific domain that could be directly derived from systematic organization of linguistic objects in an LMF standardized dictionary. But before proceeding to the core building, we have to create a dictionary fragment by extracting the relevant part of the whole dictionary. It gathers lexical entries of related senses to the domain of interest as well as their semantically related words. This dictionary fragment represents then the privileged initial source for generating the target ontology. Besides, when handling the obtained ontology, conceptual nodes always keep reference to lexical information included in the dictionary fragment (henceforth dictionary).

In order to identify the concepts of a particular domain, we consider the domain information given in the dictionary by the *SubjectField* instances. Since a concept corresponds to a meaning of a word, we can directly deduce the concepts of the domain ontology from particular instances (e.g., *Context*, *SenseExample*) attached to the *Sense* class. With regard to concepts properties, the generic LMF meta-model allows for defining **any** type of semantic relationship (e.g. *synonymy*, *hypernymy*, *meronymy*) between the senses of two or several lexical entries by means of the *SenseRelation* class. Consequently, a relation that connects two or several senses in the dictionary leads to an ontological relation linking the corresponding concepts.

3.2 The Proposed Methodology

For the construction of core domain ontology, we propose an automatic and incremental process. It consists of three main stages (Figure 1). Firstly, we identify candidates of concepts and relations relying on some identification rules that we defined in advance. Secondly, we check for duplicated candidates by means of two lists of previously identified and validated concepts and their relationships. Finally, through a validation stage supported by some validation rules, we check whether the current change is still coherent after the current change.

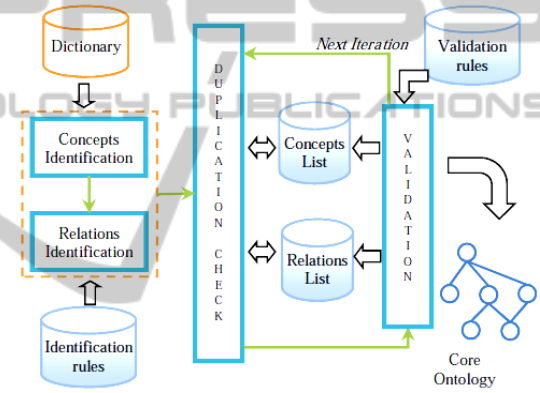

Figure 1: The core domain ontology generation process.

3.2.1 Concepts and Relations Identification

In this stage, we identify the concepts and their relationships from the lexical entries in the LMF standardized dictionary. According to our investigation on LMF structure we managed to define a set of identification rules allowing for the elicitation of ontological entities. As a result of this stage, we acquire all candidates of the core elements; each one is assigned to a given signature. In the present work, we formally define a signature of a candidate concept and a candidate relation as follow:

Definition 1. *A **concept**, denoted by **C**, is defined as a couple, **C** = (N, S), where N is the name of C and S denotes its binary tag whose value is equal either "**0**" if **C** has no relation with other concepts, or "**1**" if **C** is linked to another concept.*

Definition 2. *A **relation**, denoted by **R**, is defined by a triplet, **R** = (N, CD, CR), where N is the name of the relation, CD is the domain of **R** and CR is the range of **R**.*

3.2.2 Duplication Check

As its name indicates, the goal of this stage is to check for duplicated candidates. Such verification is very important since the identification task may imply several inter-related lexical entries per iteration. In order to detect duplicated candidates, the proposed process is based on two lists of concepts and relations, which are initially empty. They are intended to contain the concept as well as the relation signatures, emanated from valid introduction of new concepts and/or relations into the resulting core. In addition, we distinguish three types of duplication: exact duplication, quasi-exact duplication and implicit duplication. Let $\mathcal{L}_C$ and $\mathcal{L}_R$ be the lists of concepts and relations, respectively.

Exact duplication. It refers to the identification of the same copy of a previously identified candidate. This type of duplication is denoted by the ' ≡ ' symbol which is formally stated as follow:

Let $< C2 = (Name\_2, 0) >$ be a concept candidate,

$<C2 = (Name\_2, 0) > \equiv <C1=(Name\_1, ?) >$ **Iff**
$\exists < C1 = (Name\_1, ?) > \in \mathcal{L}_C$ **such that**
(Name_1=Name_2)

Let $< R_q = (relation_q, C1_q, C2_q) >$ be a relation candidate and the concepts $<C1_q=(Name\_1_q,1) >$ and $<C2_q=(Name\_2_q, 1) >$ be its two arguments,

$< R_q = (relation_q, C1_q, C2_q) > \equiv < R_p = (relation_p, C1_p, C2_p) >$ **Iff**
$\exists < R_p = (relation_p, C1_p, C2_p) > \in \mathcal{L}_R$ **;**
$\exists <C1_p=(Name\_1_p, 1) > \in \mathcal{L}_C$ **and**
$<C2_p = (Name\_2_p, 1) > \in \mathcal{L}_C$ **such that**
$< C1_p =(Name_p, 1) > \equiv < C1_q = (Name_q, 1) >$ **and**
$< C2_p =(Name_p, 1) > \equiv < C2_q = (Name_q, 1) >$ **and**
$relation_p = relation_q$

Quasi-exact duplication. This duplication denoted by the '≅' symbol concerns only relation candidates. A quasi-exact duplicated relation candidate might not be identical to an already identified candidate but it represents an equivalent. Formally,

Let $< R_q=(relation_q, C1_q, C2_q)>$ be a relation candidate and $< C1_q=(Name\_1_q, 1)>$ and $< C2_q = (Name\_2_q, 1) >$ be its two arguments,

$<R_q = (relation_q, C1_q, C2_q) > \cong <R_p = (relation_p, C1_p, C2_p) >$ **Iff**
$\exists < R_p = (relation_p, C1_p, C2_p) > \in \mathcal{L}_R$**;**
$\exists < C1_p = (Name\_1_p, 1) > \in \mathcal{L}_C$ **and**
$< C2_p= (Name\_2_p, 1) > \in \mathcal{L}_C$ **such that**
$< C1_p =(Name_p, 1) > \equiv < C2_q = (Name_q, 1) >$ **and**
$< C1_q =(Name_q, 1) > \equiv < C2_p = (Name_p, 1) >$ **and**
$(relation_p = relation_q = relation)$ **and**
symmetric **(**relation**)**

To illustrate the quasi-exact duplication with a concrete example, we consider the case of "*married-to*" symmetric relationship, for instance **R1** = (*married-to*, *Man*, *Woman*). Hence, a candidate relation with the **R2** = (*married-to*, *Woman*, *Man*) signature is considered as a quasi-exact duplicated relationship and should be ignored.

Implicit duplication. It also concerns only relation candidates. An implicit duplicated candidate is a completely different candidate but whose knowledge can be inferred from existing core elements. Formally, let $< R_u = (relation , C1, C2) >$ be a relation candidate and $< C1= (Name\_1 , 1) >$ and $< C2 = (Name\_2 , 1) >$ be its arguments,

$< R_u = (relation, C1, C2) >$ is an implicit duplicated relation **Iff**
$\exists < C3 = (Name\_3 , ?) > \in \mathcal{L}_C$**;**
$\exists <R_p=(relation, C2, C3)> \in \mathcal{L}_R$ **and**
$<R_q=(relation, C3, C1)> \in \mathcal{L}_R$ **such that**
(***transitive*** (relation)) **and (** R_p and $R_q \Rightarrow R_u$)

For example, if we have two identified relations, **R1** = (*is-a*, *Dog*, *Pet*) and **R2** = (*is-a*, *Pet*, *Animal*), then we can derive the **R3** = (*is-a*, *Dog*, *Animal*) relation candidate. Hence, a candidate with **R3** signature is an implicit duplicated relationship that must be removed from the relations list.

In all duplication types, the duplicated candidates should be ignored. Therefore, the constructed core domain ontology does not store unnecessary or useless entities. This quality criterion is also called *conciseness* (Gómez-Pérez, 2004).

It is worth mentioning that the final lists of concepts and relations are very helpful not only for the core construction, **but also** for its enrichment. Particularly, in the enrichment task, we will consider only the orphan concepts (i.e., whose binary tag is equal to 0) in order to link them to either old or new concepts. Indeed, the first stage of this process may introduce a good number of concepts that are not involved in any relations. Likewise, the list of relations is needed for the enrichment stage so as to check the coherence of the whole ontology.

Once duplication check is performed, a further validation stage is required to verify whether the resulting core remains coherent when the candidate is added to it.

3.2.3 Validation Stage

The automatic addition of non-duplicated candidates to the ontology core could bring about errors. In order to maintain the coherence of the built core, the

integration of a validation stage into the proposed process is necessary. In other words, a concept or a relation is automatically added to the output core structure only when the latter is still coherent. Gómez-Pérez has identified different kinds of errors in taxonomies: inconsistency, incompleteness, and redundancy errors (Gómez-Pérez, 2004).

Incompleteness Error. It occurs if the domain of interest is not appropriately covered. Typically, an ontology is incomplete if it does not include all relevant concepts and their lexical representations. Furthermore, partitions are incompletely defined if knowledge about disjointness or exhaustiveness of a partition is omitted.

Redundancy Error. It is a type of error that occurs when redefining expressions that were already explicitly defined or that can be inferred using other definitions.

Inconsistency Error. This kind of error can be classified in circularity errors, semantic inconsistency errors, and partition errors.

- *Circularity Errors*. A circularity error is identified, if a defined class in an ontology is a specialization or generalization of itself. For example, the concept *Woman* is a subclass of the concept *Person* which is a subclass of the concept *Woman*.
- *Semantic Inconsistency Errors*. It refers to an incorrect semantic classification. For example, the concept *Car* is a subclass of the concept *Person*.
- *Partition Errors*. A class partition error occurs, if a class is defined as a common subclass of several classes of a disjoint partition. For example, the concept *Dog* is a subclass of the concepts *Pet_Animal* and *Wild_Animal* which are disjoint subclasses of the concept *Animal*.

In the current stage, we are interested in kinds of errors that can be automatically detected (i.e., without human expert involvement). Redundancy verification has already been dealt with in the second stage of this process. As for the completeness assessment, it could not be done at this early stage of domain ontology development. Therefore, only inconsistency errors, particularly those of circularity and partition types, are addressed in the present work. After the check of the resulting core, we proceed to the update of the concepts and relations lists as well as the ontology core.

4 IMPLEMENTATION DETAILS

The methodology for core domain ontology generation from LMF-standardized dictionaries is implemented by a Java-based tool that enables users to automatically build the core structures formalized in OWL-DL, a sublanguage of OWL (Dean and Schreiber, 2004). Indeed, an OWL-DL formalized ontology can be interpreted according to description logics, and DL-based reasoning software (e.g., RacerPro or Pellet) can be applied to check its consistency or draw inferences from it. To take advantage of this, we have decided to incorporate the Pellet reasoner into our system. Indeed, it is an open-source Java-based OWL-DL reasoner tool (Sirin et al., 2007). Its consistency checking ensures that an ontology does not contain any contradictory facts. After the loading of the built OWL file, Pellet determines if the ontology is actually consistent by calling the `isConsistent()` method, whereby its `boolean` return shall decide whether the addition operation could be performed in the resulting core.

5 EXPERIMENTATION AND EVALUATION

The assessment of the high performance of the developed system as well as the good quality of the obtained ontologies is shown through the experiment carried out on the Arabic dictionary. The latter's standard model (Baccar et al., 2008) and experimental version has been worked out by our research team. This dictionary is covering various domains, of which animals, plants, astronomy and sports are but a few. Besides, thanks to LMF meta-model, our dictionary would certainly be an extendable resource that could be incremented with entries and lexical properties, extracted from other sources (e.g., Arabic lexicons, text corpora).

As far as the evaluation of the obtained results is concerned, we can obviously see that besides the fully automated level, many important benefits are noticeable in the proposed approach. Indeed, we can firstly point out that all concepts and relations represented in the core domain ontology are relevant to the considered domain. In fact, the LMF-standardized dictionaries are undeniably widely-accepted and commonly-referenced resources; thereby they simplify the task of labeling concepts and relationships. Moreover, there is no ambiguity insofar as we check the duplication of core ontology elements before their construction. In addition, no

inferred knowledge is explicitly represented. Finally, there are no consistency errors since we managed to check the coherence of the generated ontology with a specialized tool. Furthermore, a series of statistical studies were conducted on various domains toward the comparison of the obtained core ontologies with the corresponding handcrafted expected domain ontologies. We found out that about 80% of all concepts and 30% of all relations can be deduced and formalized without human expert involvement.

6 CONCLUSIONS AND FUTURE WORK

The main contribution of the current research work is to propose a novel approach for the domain ontology generation starting from an LMF standardized dictionaries (ISO-24613). Firstly, it consists in building an ontology core. Secondly, the constructed core will be further enriched with additional knowledge included in the text available in the dictionary itself. The originality of this approach lies in the use of a unique, finely-structured source and rich in lexical as well as conceptual knowledge.

Both qualitative and quantitative evaluations have shown that the constructed core elements stand for basic structures of a good quality, prone to be further fleshed out with the additional information. We expect to at the end create rich and valuable semantic resources that are suitable for NLP tasks.

The next challenges deal with how to exploit the wealth of information in the handled dictionary and preserve in the same time the good quality of yielding ontologies. Indeed, although systematic organization provided by LMF structure, much implicit information still needs to be analyzed toward digging out more ontological knowledge. That is why, ongoing work deals with the investigation on words bearing other relationship to the dictionary entry. We also plan to support the enrichment mechanism with rules maintaining the coherence of domain ontologies throughout their construction process.

REFERENCES

Almeida, M. B., 2009. A proposal to evaluate ontology content. *Applied Ontology*, 245-265.

Aussenac-Gilles, N., Despres, S., Szulman, S., 2008. The TERMINAE Method and Platform for Ontology Engineering from texts. Bridging the Gap between Text and Knowledge. IOS Press, 199–223.

Aussenac-Gilles, N., Kamel, M., 2009. Ontology Learning by Analyzing XML Document Structure and Content. *In KEOD'09,* 159-165.

Baccar, Ben Amar, F., Khemakhem, Gargouri, B., Haddar, K., Ben Hamadou, A., 2008. LMF standardized model for the editorial electronic dictionaries of Arabic. *NLPCS'2008*, Barcelona, Spain, 64-73.

Baccar, Ben Amar, F., Gargouri, B., Ben Hamadou, A., 2010. Towards Generation of Domain Ontology from LMF Standardized Dictionaries.*SEKE 2010*, Redwood City, San Francisco Bay, USA, 515-520.

Buitelaar, P., Cimiano, P., Haase, P., Sintek, M., 2009. Towards Linguistically Grounded Ontologies. *ESWC2009*, Heraklion, Greece.

Chrisment C., Haemmerlé, O., Hernandez N., Mothe J., 2008. Méthodologie de transformation d'un thesaurus en une ontologie de domaine. *Revue d'Intelligence Artificielle* 22(1): 7-37.

P. Cimiano, A. Mädche, S. Staab, J. Völker, 2009. Ontology Learning. In: S. Staab & R. Studer. *Handbook on Ontologies*. Springer.

Dean, M., Schreiber, G., 2004. OWL Web Ontology Language reference. W3C recommendation, W3C.

Francopoulo, G., George, M., 2008. Language Resource Management-Lexical Markup Framework (LMF). Technical report, ISO/TC37/SC4 (N330 Rev.16).

Gómez-Pérez, A. 2004, Ontology evaluation. In Staab, S., Studer, R. (eds.), *International Handbooks on Information Systems*.

Hirst, G., 2009. Ontology and the Lexicon. In: S. Staab & R. Studer. *Handbook on Ontologies*. Springer.

ISO 24613. Lexical Markup Framework (LMF) revision 16. ISO FDIS 24613:2008.

Jannink, J., 1999. Thesaurus entry extraction from an on-line dictionary. *In Proceedings of Fusion '99*.

Kurematsu, M., Iwade, T., Nakaya, N., Yamaguchi, T., 2004. DODDLE II: A Domain Ontology Development Environment Using a MRD and Text Corpus. IEICE(E) E87-D(4) 908-916.

Ma, Y., Audibert, L., Nazarenko, A., 2010. Formal Description of Resources for Ontology-based Semantic Annotation. *In LREC 2010*, Malta.

Montiel-Ponsoda, E., Peters, W., Auguado, de Cea, G., Espinoza, M., Gómez Pérez, A., Sini, M., 2008. Multilingual and localization support for ontologies. Technical report, D2.4.2 Neon Project Deliverable.

Na, H.-S., Choi, O.-H., Lim, J.-E., 2006. A Method for Building Domain Ontologies based on the Transformation of UML Models. *(SERA'06)*, 332-338.

Pazienza, M. T., Stellato, A., 2006. Exploiting Linguistic Resources for building linguistically motivated ontologies in the Semantic Web. *(OntoLex2006)*.

Rigau, G., Rodríguez, H., Agirre, E., 1998. Building accurate semantic taxonomies from monolingual MRDs. *COLING-ACL'98*, Montreal, Canada.

Sirin E., Parsia B., Cuenca Grau B., Kalyanpur A., Katz Y., 2007. Pellet: A practical OWL DL reasoner. Journal of Web Semantics, 5(2):51-53.