Reasoning with multiple ontologies in OWL language.

Liliana Ibeth Barbosa Santillan, Juan Jaime Sanchez Escobar, and Claudia Rosa Maria Muñoz Cano

Knowledge Management Institute of Jalisco,
Centro de Enseñanza Técnica Industrial(CETI), UNAM Chicago (National Autonomous University of México)
ibarbosa@inaoep.mx,
jjsanchez@ceti.mx,
cmunoz@cepe.unam.mx

Abstract. Over the past seven years, ontologies have proved to be one of the best logical representations of knowledge bases, since they can be represented in multiples levels and topics. However, reasoning can challenge the speed ontologies provide with output. This paper presents a convenient starting point to reasoning in OWL language with multiple ontologies in order to get a better result based on reasoning time.

Keywords: reasoning, full ontologies, logic, knowledge representation, owl

1 Introduction

According to Ontoprise [1] an Inference engine is based on a data processing system or a computer system which a purpose is to store data. The data processing system has a query unit to determine output variables by accessing the stored data. The data is allocated to predetermined classes, which are part of at least one stored class structure forming an ontology.

Gruber [2] defines an ontology as an explicit specification of a conceptualization.

There are several standard languages commonly used to define and instance Web Ontologies [4] in use, including the OWL (Web Ontology Language). OWL vocabulary is defined on top of the RDFS (Resource Description Framework Schema) vocabulary.

It is important to mention that efforts to measure the size of each ontology varies by the number of concepts and instances (sometimes is more than 1,000 of concepts), which can cause problems with reasoning time.

In addition, is not acceptable for the client to wait more than one hour to recover data. Thus, the research question is: Is it possible with the technology available up to now to recover data from one o more reasoner in less than one hour with more than 1,000 concepts into an ontology?.

A basic scenario for a reasoner with one or more data representation is shown in Fig. 1. Content providers are represented by entrepreneurs, academics, and citizens who store and retrieve data. This representation data is generated by business charts, workflows, flowcharts, mind maps, uml diagrams, and their stored data are in several formats like HTML, PDF, SWF, DXF, EPS, XML, SVG, PPT, etc. The ultimate goal of content providers is to make the representation data accessible to the expressions used to query a reasoner.

The main problem involves a common data format. A possible solution is the use of ontologies in the supporting semantic interoperability. In the last years, ontologies have been used to provide a fixed set of concepts whose meanings and relationship are stable and have been previously determined by the content providers.

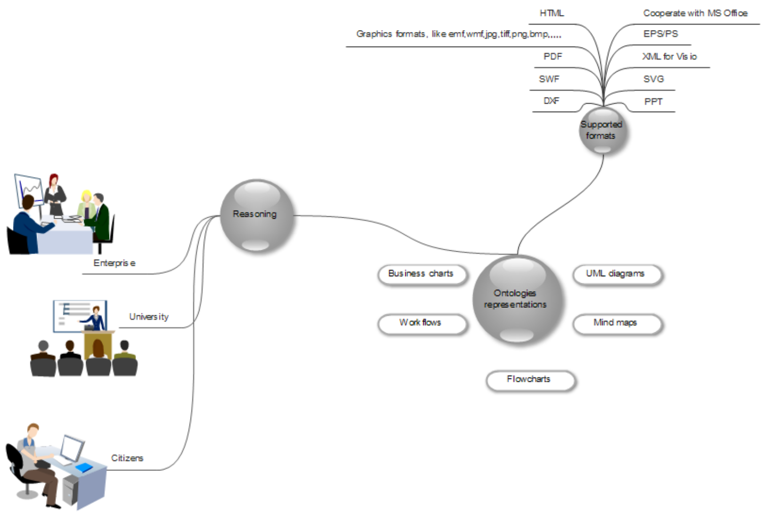

Fig. 1. A basic scenario for a reasoner with one or more data representation.

This paper presents a convenient starting point to scalable reasoning in OWL language with several ontologies in detail. The main contribution of this paper is the use of a summary ontology to access different ontologies described related domains. The remainder of the paper is organized as follows. Section 2 briefly discusses the background information of the reasoning tools. In Section 3, the detailed functionalities of reasoning with multiple ontologies are presented. Section 4 briefly evaluates the performance of the model. Finally, Section 5 discusses future projects and concludes this paper.

2 Background

Among the most used reasoning tools in the market are FACT (Fast Classification of Terminologies), Pellet [3] and RACER (Renamed ABox and Concept ExpressionReasoner). FaCT [5] , was developed at University of Manchester, supports automated TBox (concept-level) reasoning, namely class subsumption, consistency and classification reasoning. It does not support ABox (instance-level) reasoning. It can classify a DAML (Darpa Agent Markup Language)+OIL (Ontology Inference Layer)/OWL ontology by performing subsumption reasoning so as to reduce redundancy and detect any inconsistency within it. FaCT is a new generation of the FaCT reasoner. It is an OWL Lite reasoner that features a new internal architecture and new optimization.

On the other hand, RACER, the Renamed ABox and Concept Expression Reasoner[6], supports both TBox and ABox reasoning for the description logic ALCQHIR [7]. RACERs functionalities include developing ontologies, querying, retrieving and evaluating the knowledge base. It supports RDF, DAML+OIL and OWL. In addition, OilEd [8] is an ontology editor that can use both FaCT and RACER as background reasoner.

From the previous reasoning tools the authors have been choosen RACER since it supports TBox, ABox and monotonic reasoning. The reasoning problems that our research consider are time.
For our case study, seven ontologies have been developed. The Publication ontology is defined as follows:

```
(define-class Publication (?publication)
   "The act of publishing a book, paper"
:axiom-def
 (and (Superclass-Of Publication Research)
      (Template-Facet-Value Cardinality
            researchDate Publication 1)
      (Template-Facet-Value Cardinality
            publicationDate Publication 1)
      (Template-Facet-Value Maximum-Cardinality
            singlePage Publication 1))
:def
  (and (researchDate ?publication Date)
       (publicationDate ?publication Date)
       (singlePage ?publication Number)
       (researchAssistantName ?publication String)))
```

3 Reasoning with multiple ontologies

The architecture of reasoning with multiple ontologies is formed by three components: a client, a reasoner, and a data is shown in Fig. 2.

Client is an interface that a) gets a question from the user, b) comprehends the question, c) searches the answer in an efficient way and the presents that answer to the user.

Reasoner is an interface that uses a summary ontology to represent the knowledge and develop the knowledge base on T-Box and A-Box reasoning. The purpose is to control the matching of the words in the question sentence to the concepts and instances.

One of the biggest problems with reasoning with multiple ontologies in OWL language is time. The reasoner solves the problem by accessing the summary ontology, only. It does relate to the other ontologies by matching their concepts. The reasoner can solve the problem by accessing the proposed summary ontology, which in turn is related to other ontologies by a mutual matching of concepts. The summary ontology comes from the probabilistic ontology that can be obtained through the frequency of concepts. The most important characteristics of the summary ontology are: average, number of concepts, root concepts, leaf concepts, attributes and subclasses, which are the sum of annotated concepts.

The role of the summary ontology in the reasoning process is to provide a common core of concepts.

Data is an item of information represented by a set of ontologies instances.

Workflow The tasks of the workflow of the architecture of reasoning with multiple ontologies are: to aggregate and update data across multiple ontologies with the purpose of getting information and presenting it to the client.

4 Use case about research in organizations.

The context of research in organizations offers the perfect scenario to develop the case study: research in organizations. The scenario has three components as our architecture: client, reasoner and data.

Client Organizations collect data, address questions and problems, experience processes and events within established time frames, and have people and resources with particular characteristics and varying accessibility.

Reasoner Examples of some of the possible questions that a client wants to know from the data are:

1. $publicationOwner \equiv researchAssistant \cap \ni hasPublication.paper$
2. $Organization \subseteq \ni hasPublication.paper \cap \forall hasPublication.book$
3. $hasPublication \subseteq (organization, person)$
4. $Liliana \in female \cap researchAssistant$
5. $hasPublication(Liliana, ReasoningwithOntologies)$

Data An abstraction of the use case is represented through the following ontologies: Person, Research, Publication, Research Group, Organization, Research, and Summary as shown in Fig. 3. All of them are at the level data within the propouse architecture. The seven ontologies used in the approach

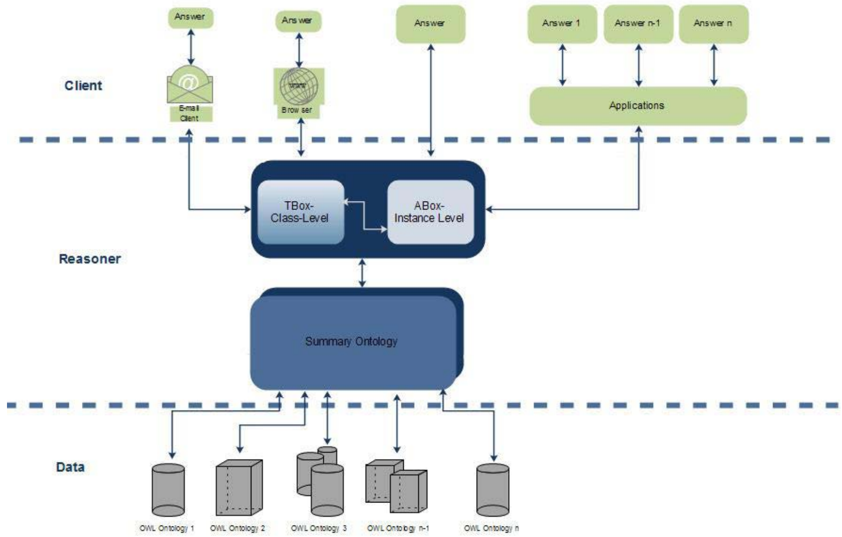

Fig. 2. The architecture of reasoning with multiple ontologies

-with a size of 1000 concepts each, have been made semi-automatically by the knowledge base of the organization. The ontology match is done through a comparison of related concepts. The inconsistencies among the seven ontologies are managed by a log and a record of up-to-date changes.

5 Conclusions and future work.

The results for this proposed architecture provided a convenient starting point to reasoning in OWL language with multiple ontologies.

The authors have showed one use case that represent the structure to real-life ontologies on the domain of research in organizations. Our analysis has shown, that reasoning time have to consist of several types of ontologies is not homogeneous as to derive parameters. The same analysis has shown that one type of summary ontology can be identified, generally. In addition, an approach to generate instances and concepts of the most important reasoning systems has been developed. Likewise, the authors are now able to provide a probabilistic ontology as the basis to generate a summary ontology. Nevertheless, there is still much more that has not been fully explored, such as how to develop concrete strategies to unify reasoning and searching in a parallel environment.

Furthermore, after running the purpose architecture the answers to those questions were not provided in one hour. Therefore, the authors propose a parallel architecture that can improve the reasoning speed while producing more

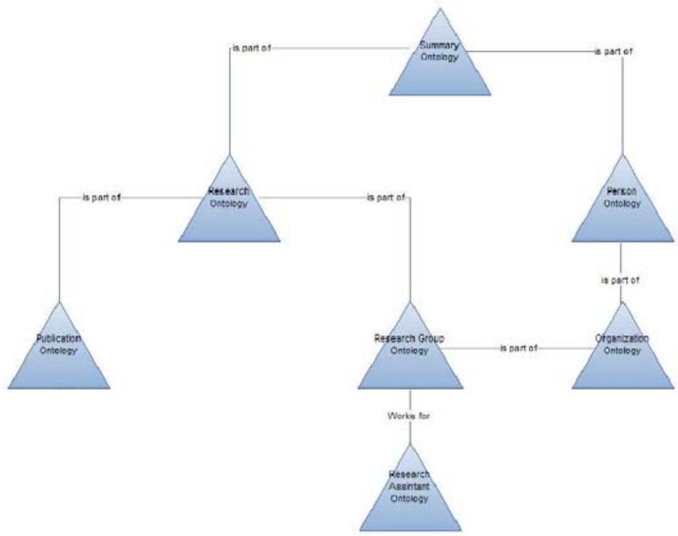

Fig. 3. The Summary ontology about Research in organizations.

accurate reasoning results to meet different clients needs. The paper described first steps towards a systematic performance evaluation for speed on reasoners. In future work, the authors are concerned into deeper discussion and concrete implementations of the parallel architecture.

Acknowledgments. This study was supported by the CONACYT.

References

1. Ontoprise. Ontostudio. 2007
2. Thomas R. Gruber. A Translation Approach to Portable Ontology Specifications. Technical Report KSL 92-71 pp 1-22
3. Parsia, Bijan and Sirin, Evren. 3rd International Semantic Web Conference. Pellet: An OWL DL Reasoner.2004.
4. Alexander Maedche and Valentin Zacharias. 13th European Conference on Machine Learning.
5. Clustering ontology-based metadata in the semantic web. 2002. Ontologies similarities wordnet-similarity.
6. Jan Hladik. Description Logics. Diego Calvanese. CEUR-WS.org. CEUR Workshop Proceedings. Reasoning about Nominals with FaCT and RACER. 81,2003.
7. V. Haarslev and R. Möller. International Joint Conference on Automated Reasoning, RACER System Description.
8. Giuseppe De Giacomo and Maurizio Lenzerini. On Instance-level Update and Erasure in Description Logic Ontologies. 19, 2009.