OTAGen: A tunable ontology generator for benchmarking ontology-based agent collaboration

F. Ongenae, S. Verstichel, F. De Turck, T. Dhaene, B. Dhoedt, P. Demeester
Ghent University - IBBT

Department of Information Technology
G. Crommenlaan 8/201, 9050 Ghent, Belgium
tel.: +32 9 331 4900 fax: +32 9 331 4899

Email: Femke.Ongenae@intec.UGent.be, Stijn.Verstichel@intec.UGent.be

Abstract

On the one hand, agent-based software platforms are commonly used these days, while on the other hand Semantic Web technologies are also maturing. It is obvious that the combination of these two technologies can bring added value through the creation of Semantic Agent-based frameworks. However, it is also known that these Semantic Web technologies, and the reasoning on ontologies in particular, can rapidly become resource intensive. In order to get a clear view on this problem, we have developed OTAGen, a highly tunable tool to generate customized ontologies and corresponding queries. The generated ontologies can then be used to evaluate at design-time the performance of the Semantic Agent-based platform as a function of the number of ontologies, users and queries.

1. Introduction and Motivation

The Semantic Web technology is gaining more and more importance and its usage in application platforms is also increasing. As depicted in Figure 1, the example platform already contains eight different ontologies, each of them additionally supported by a reasoning engine. Depending on their complexity and also on the potential large number of users of these services, the overall ontology-based process can rapidly become resource intensive. In this particular scenario, the agenda of a potentially large number of end-users is combined with their profiles, the traffic situation and on-board information services to create a personalized context-aware ontology-based platform.

Other benchmarking tools have already been developed previously, such as the Lehigh University Benchmark (LUBM) [1] or University Ontology Benchmark (UOB) [2]. The LUBM features a university domain ontology, of which

Figure 1. Example scenario for an ontology-based agent platform involving 8 different ontologies and illustrating the complex and resource intensive collaboration.

the T-Box is statically defined. Also a set of 14 different queries is included. The size of the dataset, the A-Box, can be specified during the generation process. UOB aims to increase the reasoning complexity, by introducing OWL-DL constructs. However, still a more or less static T-Box is used. Neither of these gives us the desired flexibility of specifying e.g. in detail the depth of class-hierarchies or the amount of logically defined concepts and their relationships.

2. Functional description

The workflow of OTAGen is illustrated in Figure 2. It also indicates which parameters are used in each part of the workflow. A deterministic property is added to the generation process by using a seed. Each time a set of parameters is used with the same seed, the same ontology is generated.

A user first specifies the characteristics of the conceptual level (T-Box). A number of properties of the instance level (A-Box) can also be adjusted. Finally the characteristics of the queries are identified. As a result the conceptual (T-Box) and instance (A-Box) level of an ontology are randomly and

Figure 2. Workflow of the ontology generation process

automatically generated. For this ontology also the queries are generated.

We start by generating the T-Box. First the necessary simple (not logically defined) classes are created and organized into a hierarchy. Next the logically defined classes are generated. There are 5 categories that can be created: union, intersection, complement, restriction and enumerated classes. The creation of classes of the last two categories is postponed until the necessary individuals and properties are available. In the following step the datatype and object properties are generated for the simple and union classes. The properties of the other classes need to be computed by a reasoner. Finally the object properties to connect the future enumerated classes are generated. We do this here so that they can also be used for the generation of the restriction classes. In the last step the restriction classes are created. There are 5 categories of restrictions: all values from, some values from, maximum cardinality, minimum cardinality and strict cardinality.

The generation of the A-Box starts by establishing a number of clusters. Clusters are defined as a number of classes which more strongly relate to one another, then with other classes in the ontology. Next, the actual A-Box instances of the T-Box concepts are generated. The individuals of the concepts belonging to the same cluster, which are strongly connected, have to be linked by instantiating appropriate object properties. The same procedure is repeated for linking individuals breaching the borders of the clusters. The connectivity has to be less for these individuals. As a

last step the datatype properties are defined.

We generate enumerated classes as collections of randomly chosen individuals. We already created the necessary object properties for these classes, we just need to fill in their domain and range. The same algorithms are used for linking these extra enumerated classes and relationships.

As an added value for OTAGen, situated in our overall research concerning collaborating agents, a feature for generating queries has been implemented as well. These queries are generated in the SPARQL[3] query language.

3. Applicability

OTAGen was created to support our research in Semantic Agent collaboration frameworks. It is envisaged that in such frameworks, the agents internally use ontologies. The ontologies used in these agents will not only be used as a data representation, but will be augmented with a supporting reasoning engine as well. Such platforms can potentially have a large number of agents, and thus ontologies, and also serve a large number of users. Using OTAGen, we can quickly generate a large number of different ontologies with varying complexity and also corresponding query-sets. The performance analysis of the platform and the included algorithms can therefore easily be supported.

4. Conclusions and Future Work

In this paper we have presented our developments on OTAGen, a highly-tunable ontology generator. An extensive number of characteristics can be configured to generate ontologies according to the needs of the user. In the presented example scenario, OTAGen has been used to generate customized ontologies for these agents to measure and analyze the behaviour of the overall platform as a function of different characteristics of the used ontologies. Using OTAGen, a large number of different ontologies and query-sets can quickly be generated so as to estimate at design-time the performance of the platform under different circumstances.

References

- [1] Y. Guo, Z. Pan, and J. Heflin. Lubm: a benchmark for owl knowledge base systems. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2-3):158–182, October 2005.
- [2] L. Ma, Y. Yang, Z. Qiu, G. Xie, Y. Pan, and S. Liu. Towards a complete owl ontology benchmark. *Lecture Notes in Computer Science, The Semantic Web: Research and Applications*, Springer Berlin/Heidelberg, 4011:125–139, June 2006.
- [3] E. Prud'hommeaux and A. Seaborne. Sparql query language for rdf. *W3C Recommendation*, January 2008.

**32nd Annual IEEE International
Computer Software and Applications Conference**

COMPSAC 2008

**28 July - 1 August 2008
Turku, Finland**

**COMPSAC 2008 INFORMATION
SEARCH**

**SESSIONS
GETTING STARTED**

**AUTHORS
TRADEMARK**