OntoWebML: A Knowledge Base Management System for WSML Ontologies

Christina Tziviskou and Marco Brambilla

Politecnico di Milano, Dipartimento di Elettronica e Informazione
Piazza L. Da Vinci 32, 20133 Milano, Italy
{tzivisko, mbrambil}@elet.polimi.it

Abstract. This paper addresses the topic of defining a knowledge base system for representing and managing ontologies according to the WSMO conceptual model. We propose a software engineering approach to this problem, by implementing: (i) the relational model for ontologies that corresponds to the WSML representation of WSMO; (ii) the usage of a well known Web modeling language called WebML, extended by a set of new components for exploiting ontological contents in Web services and Web applications design; and (iii) a Web-based content management system for ontologies editing and reasoning, implemented using the abovementioned software engineering approach.

1 Introduction

The Semantic Web aims at solving all the interoperability problems of Service-Oriented Architectures through the use of ontologies as the common infrastructure for data description. However, ontology management and related application development is still based on empirical design and development. In this paper, we propose a framework for specifying ontologies and applications that make use of them, based on Software Engineering techniques, such as Model Driven Development, model transformations, and a well-defined development process:

- Our modeling approach is based on visual design primitives and comprises (i) an information meta-model of the WSMO [6] specification for ontologies; and (ii) some new primitives of the WebML conceptual design language [1, 4], for exploiting ontological contents in Web services and Web applications design;
- By using the proposed approach, we describe a set of Web-based software tools allowing the developer to define ontologies, and the final user to explore them.

In the extended version of this paper [3], we investigated two main categories of related works: the Software Engineering methodologies applied to the context of Semantic Web, and the existing GUIs for the definition and editing of ontologies.

Our work is based on Web Modeling Language (WebML), a conceptual language for specifying dynamic large-scale Web applications. The WebML specification of an application consists of: (i) a *data schema* represented by a standard E-R model or UML class diagram, describing the application data; and (ii) one or more *hypertexts*, expressing the Web interfaces (*pages*) used to publish this data. *Units* in pages publish atomic pieces of homogeneous information and *operations* perform arbitrary business actions. The language currently supports business process specifications [1] and Web services capabilities [2]. WebML is supported by the CASE tool WebRatio [5]. The language is extensible, allowing for the definition of customized primitives.

The Web Service Modeling Ontology (WSMO) is a well-known conceptual model for describing Semantic Web services. It consists of four components, describing semantic aspects for: ontologies, Web services, goals, and mediators. In this work, we adopt and present the WSMO specification for ontologies [6] in terms of concepts, relations, instances of concepts and relations, and axioms.

2 Model-driven Design of Semantic Web Applications

Our approach is composed of the relational model of ontologies compliant to WSMO conceptual model, and the WebML primitives for managing this a model.

2.1 Ontology Metadata Modeling

The data model of Fig. 1 describes WSMO components for an Ontology: *Concept*, *Relation*, *Function*, *Axiom*, *Instance*, and *Relation Instance*. For reasons of limited space, a detailed description of the data structure is available in [3]. The main entity *Concept* contains defined concepts, possibly organized in a hierarchy (*HasSuperConcept* relation), defined as *Basic* or *Internal Concept*. Entity *Attribute* denotes properties of concepts (*HasAttribute* relation), having as range a set of concepts (*HasRange* relation). The actual instantiation of a concept and of its attribute values are managed by entities *Instance* and *Attribute Value* respectively [3].

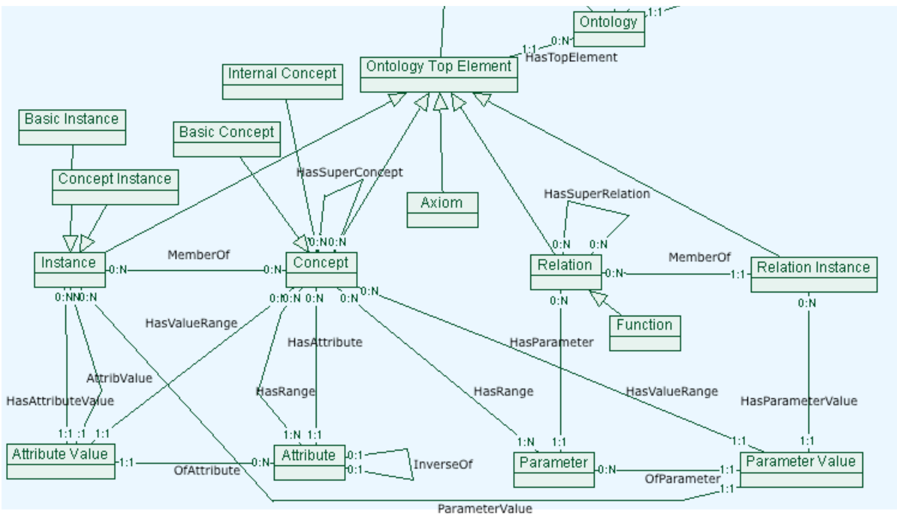

Fig. 1. Ontology metadata schema

2.2 Ontology Management Primitives

Besides the described metadata model, our approach extends the WebML language by means of new primitives that access the information in the relational representation of the Ontology and provide the designer with a simplified access to the ontological content. The new primitives are: *Subsume unit* (it returns the list of the more generic concepts/relations of a given concept); *Plugin unit* (it returns the sub-concepts/sub-relations of a given concept/relation); *Exact unit* (it returns the more specific concept/relation among two or more concepts/relations); *Subsume attributes/parameters unit* (it returns the list of the attributes/parameters of a concept/relation, that are inherited from its more generic concepts/relations); *Export and Import Ontology units* (they export the internal relational representation of an ontology in the WSML format, and update its internal relational content respectively). The primitives have been implemented in a prototype that extends the CASE tool WebRatio [5], a visual IDE for WebML. Some usage examples can be seen in [3].

3. A Web-based Tool for Ontology Editing and Browsing

Based on the instruments presented above, we implemented a set of end user tools based on web interfaces for the insertion and management of ontologies, and for the browsing of the ontological repository by the final user. The implementation of the tools is natively multi-user and distributed, thanks to the Web environment where it is deployed and to the access control mechanisms provided by WebML. The designer can decide to add at the modeling level the specification of logging and change tracking with small effort. Fast prototyping and evolution is indeed one of the main advantages provided by the conceptual modeling and software engineering methods.

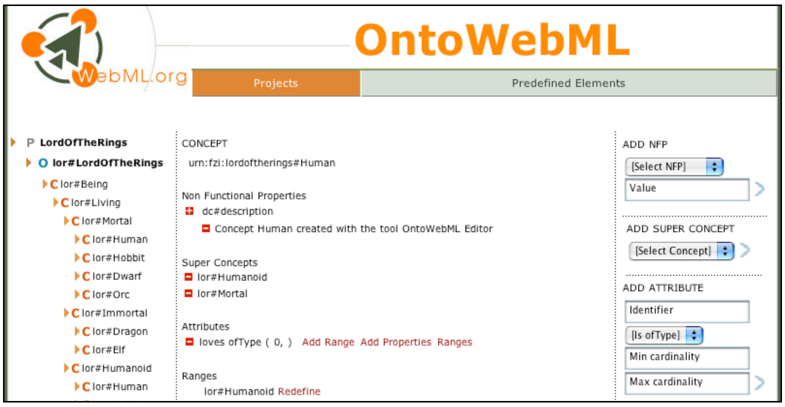

Fig. 2. *Concept Element Page* interface automatically generated by WebRatio from WebML.

The **OntoWebML Editor** allows users to specify WSMO ontologies through WebML hypertexts that rely on the structure of the underlying metadata. The tool populates entities and attributes of the metadata schema for the syntactically correct insertion of values, and optionally interacts with external reasoning tools through Web service calls for the validation of the ontology consistency. Fig. 2 is a snapshot of a page interface specified in WebML for the definition of a concept *Human* (part of the “Lord of the Rings” ontology). It presents its attributes (*Loves* of type *Humanoid*), its super concepts (*Humanoid*, *Mortal*), its inherited attributes, its non-functional properties, its namespace, and the allowed operations on these elements.

The design of similar hypertext pages can be achieved by WebML visual specifications, extended with the presented metadata schema and plug-in components. Fig. 3 presents the WebML hypertext model for ontology validation. In the *Ontology Reasoning Page*, the user receives the ontology details. By navigating the link exiting the *Ontology Details data* unit, he requests the validation of the ontology by an external reasoning tool. The ontology relational data is exported to WSML format through the *Export Ontology* component, and passed as input to the *Request Reasoning* Web service call. The latter invokes an external Web service that provides reasoning facilities; the response of the Web service invocation is used as input to the Import Ontology Component for inserting/updating the validated ontology into the database.

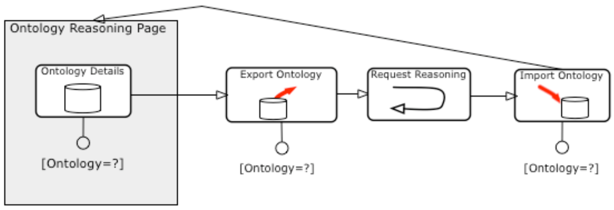

Fig. 3. WebML hypertext modeling for invoking an external reasoner.

The **OntoWebML Explorer** allows users to browse the ontology repository without the overhead of understanding the details of the WSMO descriptions. Through Web interfaces the user may navigate from concepts and relations to their instances without having any knowledge of the WSMO specification. [3] includes an example of WebML hypertext allowing browsing and search upon the ontological repository.

4 Conclusions

In this paper we presented a conceptual modeling approach to Semantic Web application design, and a set of tools for ontology browsing and management. The extended version of the paper [3] presents the benefits of our approach inherited from the model-driven implementation of applications, the back-end data storage in relational databases, and the use of the Web for distributed contents management. Future work will include the implementation of realistic industrial cases of ontology management.

References

1. Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M.: Designing Data-Intensive Web Applications, Morgan-Kaufmann, USA, 2002.
2. Manolescu, I., Brambilla, M., Ceri, S., Comai, S., Fraternali, P.: Model-Driven Design and Deployment of Service-Enabled Web Applications, ACM TOIT, Vol. 5(2), May 2005.
3. OntoWebML: A Knowledge Base Management System for WSML Ontologies, extended www.webml.org/webml/upload/ent5/5/SWESE06_BrambillaTziviskou_longversion.pdf
4. WebML Web site http://www.webml.org
5. WebRatio site http://www.webratio.com
6. Web Service Modeling Ontology (WSMO) site http://www.wsmo.org/