Ontology Access in Grids with WS-DAIOnt and the RDF(S) Realization

Miguel Esteban Gutiérrez
Ontology Engineering Group
Campus de Montegancedo s/n
28660, Boadilla del Monte, Madrid Spain
+34913363670
mesteban@fi.upm.es

Asun Gómez-Pérez
Ontology Engineering Group
Campus de Montegancedo s/n
28660, Boadilla del Monte, Madrid Spain
+34913367439
asun@fi.upm.es

Óscar Muñoz García
Ontology Engineering Group
Campus de Montegancedo s/n
28660, Boadilla del Monte, Madrid Spain
+34913363670
omunoz@delicias.fi.upm.es

ABSTRACT
This paper presents WS-DAIOnt, a framework for defining ontology access service interfaces in terms of the WS-DAI specification, extending it with the patterns, properties and behaviours needed for providing ontology access in a Grid environment. We also present WS-DAIOnt-RDF(S): a realization of WS-DAIOnt for accessing RDF(S) ontologies.

Categories and Subject Descriptors
D.2.0 [**General**]: Standards, H.4.m [**Information Systems**]: Miscellaneous

General Terms
Design, Standardization.

Keywords
Grid computing, ontologies, RDF(S).

1. INTRODUCTION
The increasing use of semantic technologies has reached almost all computer science related fields, including the Grid computing. The next generation Grid should make virtual the notion of distribution in computation, storage, and communication over unlimited resources with well defined computational semantics. A Grid node may provide new services, functions, and even new concepts that are now unknown to clients. The semantics of such services are defined by means of Ontologies, which are formal and explicit specifications of a shared conceptualization.

Ontologies can be used in the Grid for several purposes: for describing policies and sharing information, services and computing resources in Virtual Organizations; for describing formal and non formal properties of Grid resources and services; for accessing data catalogs in a conceptual and multidimensional way, etc. Right now, few Grid applications use ontologies but the access to these ontologies is not integrated in the Grid infrastructure, and the current OGSA architecture [1] does not consider ontology usage, does not define protocols and does not tackle this issue. Therefore, the provision of proven value mechanisms for accessing and managing ontologies in Grid environments is the main priority if the Grid wants to make profit of the semantic technologies already available in other areas such as the Semantic Web.

The Semantic Grid community [2] does not start from scratch. The Semantic Web community has already developed languages and tools for building and using ontologies. The W3C has recommended three languages to be used for implementing ontologies in the Semantic Web (RDF(S) and OWL) and several ontology development tools (i.e., Protégé, WebODE, KAON) for supporting the creation of ontologies in such languages. The languages recommended differ in their expressiveness (the kind of knowledge that can be represented) and in their inference mechanisms (the kind of reasoning they carry out). However, the diversity of ontology languages and tools causes translation problems, which appear when an ontology developer decides to reuse anontology with a tool/language different than the one used in its development. At the same time, several APIs can access ontologies implemented in a given language and the ontology user should know how to retrieve the ontology content, i.e. the Sesame, Jena or 3Store APIs in the case of RDF(S). At present, the Semantic Web community does not have a standard mechanism or protocol for accessing ontologies implemented in a given ontology language in a storage-and-retrieval system independent fashion, and this leads to severe interoperability problems.

In order to be able to apply semantic technologies in the Grid, we must first face and solve these interoperability issues. Thus, to provide the appropriate means for accessing and using ontologies in the Grid is a crucial issue if semantic technologies are to be used, as it is crucial the transition from monolithic, centralized ontology services to a virtual organization of Grid compliant and Grid aware ontology services that can coordinate and cooperate with each other to progress towards the Semantic Grid

2. WS-DAIOnt: a framework for specifying ontology access services
One of the main goals of the OntoGrid project is to explicitly share and deploy knowledge to be used for the development of innovative Grid infrastructure and for Grid applications. To address this challenge, the OntoGrid project is developing a Semantic Grid reference architecture (named Semantic OGSA, a.k.a. S-OGSA) and the technological infrastructure for the rapid prototyping and development of knowledge-intensive distributed open services for the Semantic Grid. A key module of this Semantic OGSA reference architecture is the component that provides access to Ontologies, being our main goal in OntoGrid to develop, build on, adapt and extend existing ontology services to be Grid compliant. This high level, very general objective can be refined as follows: to provide seamless access to heterogeneous and distributed ontology sources created according to different knowledge representation formalisms by means a uniform ontology access mechanism.

With the goal of avoiding the proliferation of different access mechanisms for ontologies implemented in languages of the Semantic Web, the OntoGrid project is specifying and designing an ontology access mechanism for the Grid, whose formal name is WS-DAIOnt. It provides a WS-DAI [3] based framework for defining ontology access service interfaces; it uses the standard grid data access vocabulary, and extends the data access mechanisms with the patterns, properties and behaviours needed for providing ontology access. The WS-DAIOnt specification and the accompanying realizations (WS-DAIOnt-RDF(S), …) define the data access services that are needed for dealing with ontologies in Grid environments. The specification is fully compliant with S-OGSA (so with OGSA) and WS-DAI and is based on up-to-date Web Services standards such as WS-RF and WS-Addressing.

WS-DAIOnt is built over four pillars:

- *Unified basic terminology*. WS-DAIOnt defines a neutral vocabulary for naming the ontology elements to be used when dealing with ontologies in Grid environments.
- *Ontology elements usage patterns*. WS-DAIOnt defines how the messages, methods, interfaces, and services must be specified in order to provide functionalities in a standard way.
- *Ontology elements possible relationships*. WS-DAIOnt defines how to specify how each ontology element is related to each other.
- *Ontology access services behaviors*. WS-DAIOnt defines the expected behaviour of the predefined common components and functionalities.

By extending WS-DAI and, therefore, the OGSA architecture, with WS-DAIOnt and the accompanying realizations, we provide the current Grid architecture with a standard way for supplying ontology access and management capabilities, thus enabling the future integration of semantic technologies in the Grid architecture. A preliminary detailed version of the WS-DAIOnt specification is available in the deliverable D3.1 of the OntoGrid project [4], available at the project's site: http://www.ontogrid.net.

3. WS-DAIOnt-RDF(S): accessing RDF(S) in the Grid

To test the correctness of our approach, we decided to start the implementation of WS-DAIOnt with ontologies implemented in RDF(S). The WS-DAIOnt-RDF(S) realization offers a framework for defining ontology access service interfaces using the WS-DAI vocabulary and for defining the set of messages, properties and behaviors needed to provide ontology access to ontologies implemented in RDF(S) (see figure below)

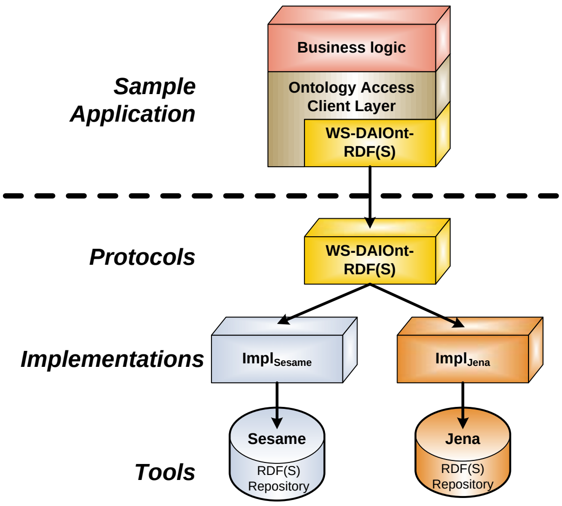
The infrastructure chosen for developing WS-DAIOnt-RDF(S) is the Globus Toolkit 4, which provides the Grid infrastructure, specifically the Java WS-Core. Regarding the ontology repository, we selected Sesame, using SeRQL for accessing the ontologies stored in it.

In the WS-DAIOnt-RDF(S) framework architecture we can distinguish three service layers: the upper layer allows selecting the repository to be used; the intermediate layer allows interacting with the selected repository directly; finally, the lower layer includes the services that permit interacting directly with the RDF(S) knowledge components. Up-to-date the following services have been fully developed:

`RDFSRepositorySelectorService`. This service allows selecting the specific repository to be used in case that multiple repositories are available. Each repository is identified by a unique identificator.

`RDFSRepositoryService`. This service provides access to all the knowledge components (properties, statements, classes, etc.) in the repository. Some of the operations provided are, for instance, to get all the classes of the repository (`getAllClasses`) or to get a class by its URI (`getClass`).

`RDFSClassService`. This service provides access to a given RDFS class, i.e., get the sibling classes of the class (`getSiblings`), get the related subclasses (`getSubClasses`) or superclasses (`getSuperClasses`).



T

The RDFSClassService needs to interact directly with the Sesame repository to execute its methods. As we are trying to provide storage-independent access to RDF(S), we have to decouple the repository from the service implementation.

Generally speaking, those services which have to interact with the repository must do it in a loosely coupled way. This is achieved by an extra data access layer, namely RDFSConnector, which adapts the access to the repository to the services requirements.

4. ACKNOWLEDGMENTS

This work is supported by the OntoGrid project (FP6-511513) and by a U.P.M. PhD student grant.

5. REFERENCES

[1] I. Foster (Editor), D. Berry, A. Djaoui, A. Grimshaw, B. Horn, H. Kishimoto (Editor), F. Maciel, A. Savva (Editor), F. Siebenlist, R. Subramaniam, J. Treadwell, J. Von Reich; The Open Grid Services

[2] D. De Roure, N.R. Jennings, N.R. Shadbolt; The Semantic Grid:Past, Present and Future. In: Proceedings of the IEEE Volume 93, Issue 3, March 2005, pp 669-681.

[3] M. Antonioletti, M. Atkinson, S. Malaika, S. Laws, N. W. Paton D. Pearson and G. Riccardi; Web Services Data Access and Integration (WS-DAI). DAIS-WG Informational Draft, 13th Global Grid.

[4] M. Esteban Gutiérrez (ed), S. Bechhofer, O. Corcho, M. Fernández-López, A. Gómez-Pérez, and more; Specification and Design of Ontology Grid Compliant and Grid Aware Services. D3.1 OntoGrid Project, 27th April 2005.