Designing Ontologies and Distributed Resource Discovery Services for an Earthquake Simulation Grid

Mehmet S. Aktas[1,2], Marlon Pierce[2], and Geoffrey C. Fox[1,2]
Indiana University, Bloomington, IN, USA
{maktas, marpierc, gcf}@indiana.edu

Abstract: Locating resources of interest is a fundamental problem within large scale resource-intensive environments. An effective way of solving this problem is providing and managing metadata about resources. In this paper, we describe an initial ontology design and the architecture of a distributed information system that we are implementing. We introduce an efficient notification based caching mechanism to make metadata available in open hypermedia peer-to-peer systems.

1 - Introduction

This paper describes our initial work designing and developing a metadata management and discovery system using Semantic Web [1] technologies, focusing particularly on the problems and needs of the SERVOGrid (Solid Earth Research Virtual Observatory Grid) [2][3][4] project. SERVOGrid integrates measured and calculated earthquake data (such as fault models, seismic events, GPS measurements) with simulation codes, all of which are accessed through various Web Services. SERVOGrid resources are located at various institutions across the country, with growing interest in participation from international ACES [5] partners.

SERVOGrid requires Grid services for remote code execution and data access, as in traditional science Grids. However, our challenges also include modeling and managing metadata information generated by the Grid services. We must also assume that resources are volatile and not suitable for centralized management: Grid resources tend to evolve over time, may include very distributed partners, and may be temporarily inaccessible. Thus information management is crucial.

This paper introduces our efforts to represent and facilitate the use of metadata in distributed systems using Semantic Web technologies. The Semantic Web adopts the Open Hypermedia [6] model. In this model, resource information, that points users to resources, is indicated in separate documents. This information is the semantic metadata. We consider the SERVOGrid environment as an open hypermedia peer-to-peer system. In this paper, we describe our initial ontology design and the architecture of a distributed publish/subscribe system for managing metadata discovery and access in the SERVOGrid environment.

2 - SERVOGrid Resources Ontology

In this section, we describe our effort to create ontology aided metadata instances for SERVOGrid resources. The SERVOGrid project has a collection of codes, visualization tools, computing resources, and data sets that are distributed across the grids. Instances of a well-defined ontology will be used to describe specific resources as metadata. We outline the steps of creating a SERVOGrid ontology as follows.

Defining Classes: The first step is to group together related resources in order to create an ontology. There are three major groups of SERVOGrid resources that need to be classified. These groups are *ServoCodes*, such as simulation and visualization codes, *ServoData*, such as fault and GPS data,

[1] **Computer Science Department,** Lindley Hall, Room 215 150 S. Woodlawn Ave Bloomington, IN 47405

[2] **Community Grids Labs,** 501 N. Morton Street, Suite 222, Bloomington, IN, 4740

and *ComputingPlatforms*, such as computers and web services. We observe the following hierarchical classification of classes to group together SERVOGrid resources in Figure-1.

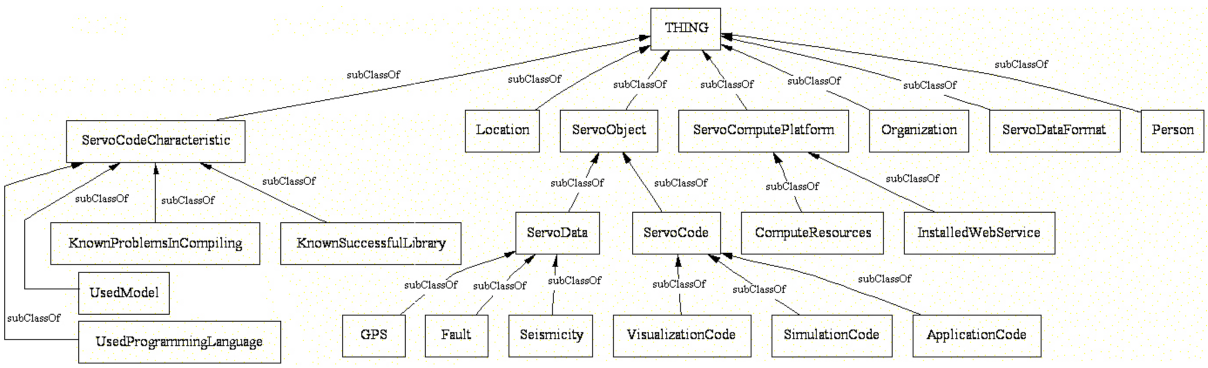

Figure -1 Class Hierarchy of SERVOGrid Ontology

For lack of space, the full descriptions of the ontology classes are not given here but they can be found from the preliminary schemas available at [7].

Defining Properties: Ontology classes are created to group together similar resources. Since the class hierarchy does not give any information about the classes themselves, we defined various meaningful properties for each class of the ontology. In addition to these properties, we have also used DC (Dublin Core) [9] and vCard [10] metadata definition standards in our ontology. In Figure-2, we sketch the properties that link SERVOGrid Resources. The detailed descriptions of the properties are available at [7].

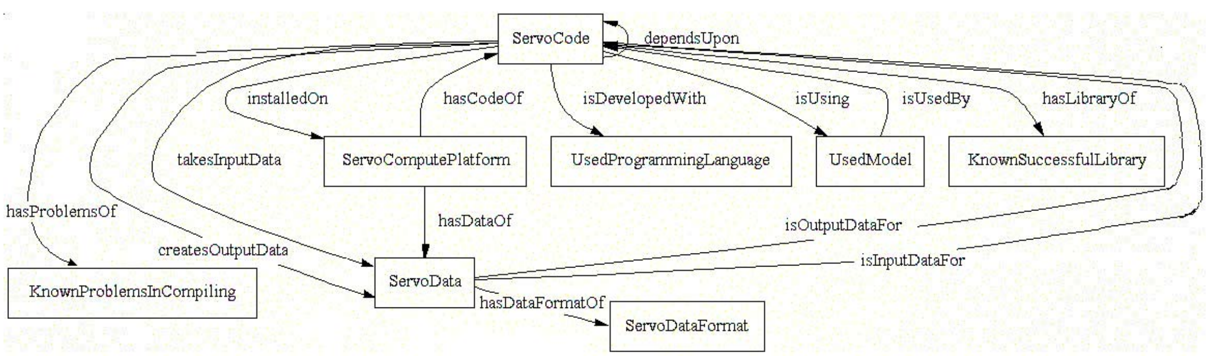

Figure -2 Properties of SERVOGrid Ontology

Generating the Ontology and Metadata Instances: As a next step, the description of these classes and properties are written in the semantic markup languages RDF/RDFS [11] and OWL [12]. The full ontology and examples of the metadata instances are available from [7][8].

As mentioned above, a decentralized approach to metadata management is desirable. Decentralization of metadata management, however, leads to potentially low search performance than centralized approaches. To improve response time and lower network traffic, we introduce a caching mechanism which is based on notification-based publisher/subscriber interactions. We discuss the details of our proposal in the following section.

3 - Notification based caching approach in managing metadata

Our system is designed to run as a standalone Java program on distributed servers. These servers might accommodate a metadata repository or might be interacting with a separate database repository through Web Services. The principal feature of the system is that after an initialization phase achieved through crawling, it establishes updatable caches that subscribe to remote resources and listen for updates. We introduce proxy caching to distribute the copies of the metadata across the peers in the network. When a user posts a query to a proxy cache, starting from this cache, at each node, the cache is queried for the requested metadata and then the query is passed to nodes that are semantically connected. Results are returned to the proxy cache that initiated the query and the proxy cache returns results to the client. At each step, each cache relays the query only to the nodes that it is aware of. We use breadth first search to simply explore the next nodes if the requested metadata is not already cached.

When using cached copies across the nodes of the network, information consistency becomes an issue. To avoid information inconsistency, we utilize a notification based interaction pattern between the source cache and other caches where remote copies exist. The cashing system is based on a notification-based brokering system such as NaradaBrokering [13]. In this scenario, each server interacts with the NaradaBrokering system. Our system is designed to work with one or multiple NaradaBrokers. Each server can be both a subscriber and/or publisher to different topics. Subscribers listen to updates of the metadata instances, so that the cached local copy remains consistent with the original metadata. Publishers are responsible for publishing any changes in the metadata to the related topics to keep the cached copies up-to-date. In this scheme, RDF predicates serve as topics. Subscriber nodes must subscribe to predicate topics to receive the updates for the resources that are in the range of that predicate. We can illustrate this in the following example which is a triple regarding a simulation code "disloc.c".

Example:	Subject	<http://www.servogrid.org/servo/instances/servoCode#disloc_instance>
	Predicate	<servo:installedOn >
	Object	<http://www.servogrid.org/servo/instances/servoPlatform#grids_instance>

In this example, the topic would be <servo:installedOn >. Remote cache copies that carry the instances of the "grids_instance" metadata (e.g. set of triples describing the grids instance) would listen to the topic <servo:installedOn> for any updates. The origin cache for the "disloc_instance" resource metadata publishes any changes to the <servo:installedOn> topic to keep the remote copies up-to-date. The predicates are uniquely defined by the ontology. The range of each predicate is defined based on the SERVOGrid ontology. This provides an automatic way of subscribing to topics (since we know which resources can subscribe which topic), that is simple to implement.

4 - Related Work

Finding resources by stepwise exploration has been first studied in the hypertext community [14], [15]. The Lagoon Cache [16] is similar to our approach, using the proxy caching method to improve search performance and bandwidth consumption. Our main difference is that we utilize the idea of "caching by enforcement", meaning the resource provider is expected to propagate the updates to all remote cached copies (by using a notification system) to keep the remote copies up-to-date. The Globus Alliance [17] has released WSRF Specifications [18] which include WS-Notifications [19]. We find the following similarities with our approach. Both approaches give the responsibility of propagating the updates to the service providers. Also, both approaches have notification based publish/subscribe interactions to notify the corresponding nodes regarding updates. There are differences however. WS-Notification Specifications define the topics with XML syntax and the specifications do not define any rules or restrictions regarding becoming a publisher or a subscriber to a topic. We define each topic as predicate of a triple. The SERVOGrid ontology defines the rules regarding which resources can subscribe which topics. Also, our approach does not need a discovery system to find out what available topics exist. The topics (predicates) are well defined with the namespaces used in SERVOGrid ontology, whereas the WS-Notification approach needs to provide mechanisms to discover the available topics.

5 - Conclusions and Future Work

This paper has described the ontology development work for describing SERVOGrid resources. We are coupling this information representation to a notification-based caching system that will allow stepwise exploration of metadata instances by using Semantic Web technologies. In future work, we plan to further study fast and efficient ways of exploring open hypermedia peer-to-peer systems for the SERVOGrid project. Our goal is to provide an efficient metadata management system where semantic metadata is used to describe resources in a distributed fashion.

6 - Acknowledgement

The work for this project was funded through NASA Ames Research Center through Grant 1247333. The SERVOGrid project is funded through the NASA Computational Technologies program under Grant 1253656.

References

[1] W3C Semantic Web Site: http://www.w3.org/2001/sw
[2] "Building Problem Solving Environments with Application Web Service Toolkits", Choonhan Youn, Marlon Pierce, and Geoffrey Fox, ICCS03 Australia June 2003
[3] "Interoperable Web Services for Computational Portals", Marlon Pierce, Choonhan Youn, Ozgur Balsoy, Geoffrey Fox, Steve Mock, and Kurt Mueller. SC02 November 2002
[4] "Interacting Data Services for Distributed Earthquake Modeling", Marlon Pierce, Choonhan Youn, and Geoffrey Fox, ACES Workshop at ICCS June 2003 Australia
[5] ACES - APEC Cooperation for Earthquake Simulation Web Site http://www.aces.org.au
[6] "Open Hypermedia: Systems, Interoperability and Standards". Wiil, U.K. Journal of Digital Information, Vol.1, No.2 (1997)
[7] Link to the SERVOGrid ontologies http://ripvanwinkle.ucs.indiana.edu:4780/examples/download/schema
[8] Link to the SERVO ontology instances http://ripvanwinkle.ucs.indiana.edu:4780/examples/download/data
[9] Dublin Core Meta-data Initiative Web Site http://dublincore.org/documents/dces
[10] The vCard version 3.0 Specifications: http://www.ietf.org/rfc/rfc2426.txt
[11] W3C - Resource Description Framework Site: http://www.w3.org/RDF/
[12] W3C – Web Ontology Language Overview http://www.w3.org/TR/2004/REC-owl-features-20040210
[13] NaradaBrokering Project: http://www.naradabrokering.org
[14] "Searching for Arbitrary Information in the WWW: the Fish-Search for Mosaic.", P. De Bra, R. Post. WWW Conference, 1994. http://citeseer.ist.psu.edu/debra94searching.html
[15] "The Shark Search Algorithm – An application: Tailored Web site mapping", Michael Hersovici, et al. In WWW7, 1998.
[16] "Information retrieval in the World Wide Web: Making client-based searching feasible", P. De Bra, R. Post, in: *Proceedings of the 1st WWW Conference, Geneva*, 94.
[17] Globus Project Website: http://www.globus.org
[18] "The WS-Resource Framework", Czajkowski K., et al. http://www-106.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf
[19] "Publish-Subscribe Notification for Web Services", Graham S., et al. http://www-106.ibm.com/developerworks/library/ws-pubsub/WS-PubSub.pdf