Metadata Management in S-OGSA

**Oscar Corcho, Pinar Alper, Paolo Missier,
Sean Bechhofer, Carole Goble**

School of Computer Science, University of Manchester
Oxford Road, Manchester M13 9PL, United Kingdom
{ocorcho,penpecip,pmissier,seanb,carole}@cs.man.ac.uk

Abstract. Metadata-intensive applications pose strong requirements for metadata management infrastructures, which need to deal with a large amount of distributed and dynamic metadata. Among the most relevant requirements we can cite those related to access control and authorisation, lifecycle management and notification mechanisms, and distribution transparency. In this paper we discuss such requirements and propose a systematic approach to deal with them in the context of the S-OGSA architecture.

1 Introduction

Metadata is "structured data about an object that supports functions associated with the designated object" [1]. Metadata can be attached to many different types of objects, including documents, forms, Web services, applications, databases, etc. These objects can be available in different formats and locations. And metadata can be expressed in a wide range of languages (e.g., in natural language, as lists of terms, in formal languages like RDF) and using a wide range of vocabularies (e.g., keyword sets, concept taxonomies, formal axioms). Depending on these aspects, many authors use the terms semantic versus non-semantic, or rich versus non-rich metadata.

There are many technologies available to manage those annotations, such as Jena, Sesame, Boca, Oracle-RDF, Annotea, Technorati, etc. They give a good level of support for current metadata-intensive applications. However, they fall short for some of the metadata management requirements that new applications are posing, such as metadata distribution, lifecycle management and authorisation.

This paper starts by providing a set of requirements for advanced metadata management (Section 2). These requirements apply to metadata-intensive applications in domains like bioinformatics, social sciences, engineering, and market analysis, among others. Existing approaches and technologies do not meet all of the previous requirements[1]. Hence we make a proposal for managing metadata as first-class resources in distributed systems, so that we can deal with the previous requirements (Section 3). Finally we provide some conclusions and future work.

[1] This analysis is out of the scope of this paper, and can be found in [2].

2 Metadata Management Requirements

In this Section we describe some of the advanced requirements for metadata management that are needed in some metadata-intensive applications and are not sufficiently addressed by existing approaches.

2.1 Metadata should be stored and accessible in a distributed manner

.

In the context of Semantic Web and Semantic Grid applications, many systems need to integrate distributed metadata (and ontologies), which may be supplied by multiple parties and with different technologies.

The current situation is that most metadata management systems provide repositories that are designed for centralized use, with metadata consumers and producers acting as local-access clients using specialized APIs. Remote distributed access to metadata and ontologies has received little attention, causing each system to depend on one particular technology, thus reducing interoperability.

A service-oriented approach to metadata and ontology access could improve this situation. However, this should not be done by simply wrapping current metadata management systems with web services that replicate their local APIs, because of the reasons described below.

2.2 Metadata should be accessible with technology-independent service-oriented protocols

.

Metadata can be available in multiple forms (in RDF, as social tags, in natural language, etc.). Hence we require systems that enable the integration and exploitation of this heterogeneous metadata.

Even in the case of a single form of metadata (e.g. RDF(S)) different repositories use different abstractions and means to handle metadata. Therefore there is a clear need for a common abstraction for metadata. Such a model would also enable the systematic development of services that combine other services and applications with varying levels of semantic capabilities for dealing with and interpreting metadata.

2.3 Metadata should evolve together with the resources that it describes and the vocabularies that it uses

.

Metadata evolves due to different reasons. However, the dynamics of metadata is poorly supported by existing technologies, although some work has been done with respect to the semantics of change and its propagation to the related metadata [3–5].

Metadata should be maintained up-to-date in a cost-effective manner. This includes maximising the automation of different aspects of the knowledge life-cycle, managing the evolution and change of metadata and knowledge models in distributed contexts, and synchronising adequately the evolution of all these related entities by means of notification mechanisms.

2.4 Metadata access should be controlled

.

The owner of a piece of metadata may want to define the conditions and rules under which others can access her metadata.

Existing technologies vary in their support and granularity for security, which normally means access-control only. For instance, Sesame provides built-in user/role-based access control per repository, while Jena has no built-in access-control support since it assumes that this can be provided by the underlying database technology. Moreover, all these access control mechanisms have their own specific APIs and conventions, which creates overhead at the client layer (need for multiple usernames and passwords, or multiple sign-ons).

Securing message exchanges, authentication and access-control are well-studied problems in distributed environments like those of web services and service-oriented Grids. Besides, there are standards[2] and open-source reference implementations[3] for global user identification, single sign-on, communication encryption and representation and decision of resource-sharing policies.

3 Metadata Management in S-OGSA

The previous Section presents requirements for the inclusion of advanced metadata management capabilities in metadata-intensive applications. In this Section we describe the approach for metadata management that we propose in the context of the Semantic-OGSA (S-OGSA) architecture [6]. S-OGSA is an extension of the Open Grid Service Architecture [7] for the development of Grid applications that need to use explicit and distributed metadata. Although S-OGSA was first conceived for developing Grid applications, it can be applied to any other type of metadata-intensive distributed system.

We start describing how S-OGSA proposes to maintain the association between resources and their corresponding metadata. The S-OGSA model is driven by the principle that such *associations should themselves be first-class resources that can be distributed and managed in a service-oriented manner*. This gives support to the first two requirements from Section 2, and is the basis for giving support to the rest of requirements.

[2] e.g., XACML: www.oasis-open.org/committees/xacml/, WS-Security: www.oasis-open.org/committees/wss/

[3] e.g., Globus GSI: http://www.globus.org/toolkit/docs/4.0/security/

We have coined the term Semantic Binding (SB) [8] to denote this new type of resource, which constitutes the core of S-OGSA. Semantic Bindings represent associations between any set of resources and any set of knowledge entities (e.g., ontologies, rule sets, controlled lists of terms). These associations contain metadata, which *can be encoded in different formats*: RDF, natural language, as social tags, etc. Besides, *different Semantic Bindings can describe the same set of entities*. For example, different tools or persons may create different annotations for the same resource.

In the following subsections we will describe in more detail the capabilities associated to Semantic Bindings and how they provide solutions for our metadata management requirements.

3.1 Semantic Binding Capabilities

In [6] we describe the model used to represent Semantic Bindings, expressed as an ontology that extends the Grid ontology described in [9]. The main properties of a Semantic Binding are the set of resources to which it refers (that is, the resources for which it contains metadata), the set of knowledge entities that the metadata is based on, and the actual metadata that they store. Besides these properties, others like the Semantic Binding state, creation time, last modification time, etc., are stored, and will be used for managing its lifetime and its notification and authorisation mechanisms, described in the following sections.

Besides the basic properties that describe Semantic Bindings and contain their relevant information, other basic operations are provided by the service suite associated to them (the *Semantic Binding Service*, its corresponding *Factory* and a *Metadata Service* that gives a unified view of the metadata stored by several Semantic Bindings). These operations are shown in figure 1:

- *Create*. It creates a Semantic Binding, given the resources that it describes, the Knowledge Entities used for the description, and the actual metadata to be stored.
- *Update Resource and Knowledge Entity References*. They allow managing the references to Resources and Knowledge Entities of the Semantic Binding.
- *Update Semantic Binding Content*. It updates the metadata stored in the Semantic Binding, due to its reannotation or curation.
- *Destroy*. It destroys the Semantic Binding, together with its content, immediately or at a scheduled point in time.
- *Archive*. It archives the Semantic Binding content so that it is not active but its content can be retrieved in case that it is needed later, such as for provenance reasons.
- *Query*. It executes a query over the metadata stored by the Semantic Binding. Queries will be sent in a query language that the Semantic Binding supports, and can take into account the knowledge entities to which the Semantic Binding refers or not.

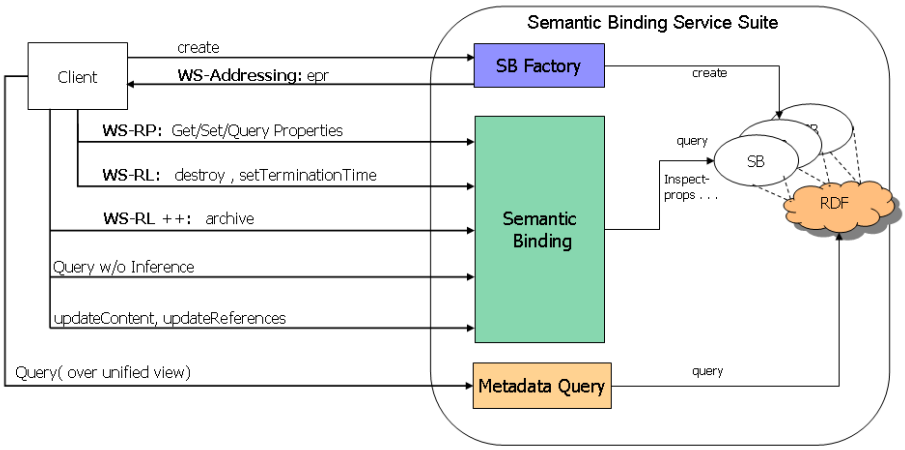

Fig. 1. Functionality of the Semantic Binding Service

These properties and functionalities are implemented in the current S-OGSA reference implementation[4], which complies with the WS-Resource Framework [10] (WSRF) family of specifications. This implementation can be deployed on the Globus Toolkit 4 platform[5] and in Apache Tomcat[6].

In the following sections we give details about how we deal with the rest of requirements, namely the need for managing the lifetime of metadata and the notifications of metadata changes to the interested distributed parties and the controlled access to metadata.

3.2 Semantic Binding Lifetime

In certain applications, metadata can be a dynamic entity subject to frequent changes. The reasons for such changes are diverse: they can be related to changes in the resources that metadata describes or in the referred knowledge entities. Change types are also diverse: resources and knowledge entitites can evolve, become suddenly unavailable, be destroyed, etc.

Some of these changes may cause metadata (and consequently its corresponding Semantic Binding) to become invalid. Therefore in most of the cases the systems that depend on it can no longer rely on it. Other changes may not have an influence on metadata validity (e.g., the removal of a concept in an ontology for which there are no instances in the stored metadata).

[4] Available at http://www.ontogrid.eu/ontogrid/downloads.jsp and at the OntoGrid CVS.
[5] http://www.globus.org/toolkit/
[6] http://tomcat.apache.org/

Finally, the metadata that describes a resource may become invalid after a given period of time. This can happen when a new annotation tool has been made available and the resource has to be reannotated, when a metadata curation process is in place, etc.

In order to deal with all these changes in a principled way, S-OGSA defines SBs as stateful resources with a defined lifetime and identifies the states and state transitions that a SB can go through throughout its lifetime. The corresponding state diagram is presented in the next subsection.

An Extensible State Machine for the Semantic Binding Lifetime The state diagram associated to an SB, shown in Figure 2, includes a set of fundamental states and state transitions, as well as the external events that cause the transitions. The specification of SB lifetime extends the WS-ResourceLifetime specification, a part of WSRF that standardizes the way that resources are destroyed, and defines resource properties for the inspection and monitoring of a resource lifetime. While WS-ResourceLifetime is focused exclusively on resource destruction, we extend it to include any life-changing event that may affect the validity and updates of an SB. Furthermore, the basic state machine presented here can be extended with sub-states if needed in a certain application.

The explanation of the state transition diagram is as follows. When it is first created, a Semantic Binding SB is in the *Valid* state. We denote with Res_{SB} and KE_{SB}, respectively, the set of Resources and Knowledge entities that are part of the association, and with $content_{SB}$ the metadata payload within SB.

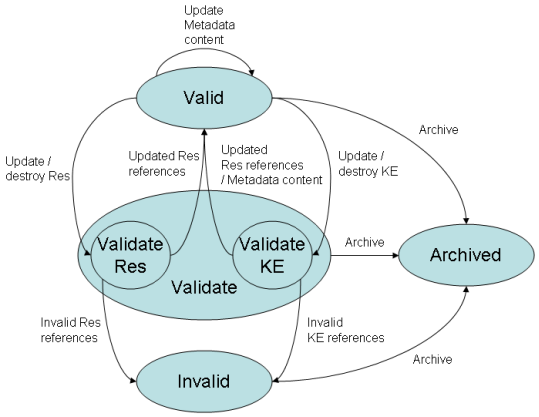

Fig. 2. State transition diagram for a generic Semantic Binding

State transition events are of the following types:

- Changes in the described resources, denoted by $Res_{SB} \rightarrow Res'_{SB}$.
- Changes in the Knowledge entities, i.e., $KE_{SB} \rightarrow KE'_{SB}$

– Updates to the SB content: $content_{SB} \rightarrow content'_{SB}$.

Note the Resources and Knowledge entities can also be destroyed: $Res_{SB} \rightarrow \emptyset$, $Res_{SB} \rightarrow \emptyset$. In addition to these external events, a content expiration date can also be associated to an SB, so that it is automatically considered stale upon expiration.

For a *Valid* SB, these events cause its transition to either one of two possible *Validate* states, *Validate Res* and *Validate KE*. These are interim states in which the SB may be invalid, and is awaiting re-validation. A re-validation process, either manual or automated, is any procedure that updates any or all of Res_{SB}, KE_{SB}, or $content_{SB}$, and which results in a decision as to whether the updated entities represent a new valid combination[7].

– For a *ValidateRes* SB, such procedure determines whether the existing metadata can be associated to the new Resources, and provides an update to the references in SB to Res'_{SB}. For example, following a change in a workflow that is described with a piece of metadata, the procedure determines whether the same metadata can be associated to the new workflow.
– For a *ValidateKE* SB, the problem is to determine whether the new ontology can still be used to interpret the old metadata.

Finally, the *Archived* state indicates that a SB is still available for inspection, but it has been superseded by a more recent version.

3.3 Notification of Semantic Binding Changes

Requirement 3 suggests that the metadata-aware services that use SBs should be informed of any state change for those SBs, since this may affect their behaviour. For this purpose, S-OGSA defines a set of notification mechanisms based on WS-Notification. S-OGSA proposes the use of a set of pre-defined topics associated to the changes described above. Any consumers can subscribe to those topics (or any other set of application-dependent topics) in order to be notified of the changes. Figure 3 shows that a client has subscribed to notifications of state changes in the SB, and that the SB is subscribed to notifications of change in a set of Resources and Knowledge Entities (specified by their last modification time), which will probably make it change its state.

Services that receive any of these notifications will decide, as part of their business logic, how to react to the changes that they are notified about. The S-OGSA specification does not enforce any specific type of behaviour. However, as part of our future work, we are designing a service, called *SB housekeeping service*, which monitors SB lifetime by subscribing to all their topics. This service will be responsible for activating application-dependent re-validation procedures (validation of the SB content, triggering of re-annotation processed, etc.).

[7] In both cases, the SB goes back to the *Valid* state in case of successful validation, and to *Invalid* otherwise.

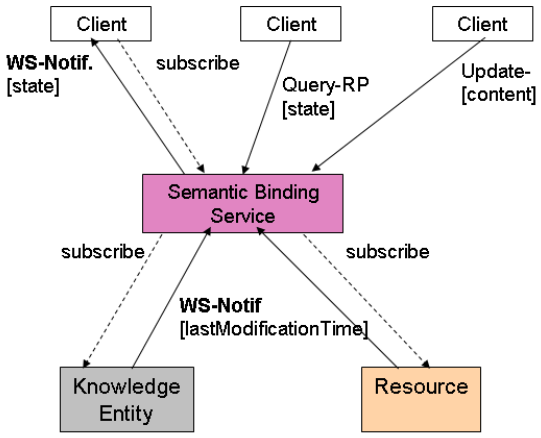

Fig. 3. Notification of Semantic Binding changes

3.4 Security over Semantic Bindings

The last requirement suggests the need to control the access to metadata in more fine-grained ways than what it is currently done with current technology. Furthermore, other security-related aspects like securing message exchanges, authentication, etc., may need to be taken into account in many situations.

S-OGSA provides the framework needed to support security in metadata management. Metadata is treated as a first-class resource; hence standard security mechanisms can be applied to metadata in the same way as it is done with other resources in a distributed system. This includes, among others, the possibility of specifying and enforcing access control policies over each SB. Besides, since the reference implementation of S-OGSA can be deployed on top of Globus Toolkit 4, its associated security mechanisms, such as Globus GSI, can be also applied, ensuring both message and transport level security. These are based on standard X.509 end entity certificates and proxy certificates, which are used to identify persistent entities such as users and servers and to support the temporary delegation of privileges to other entities.

With this framework, it is possible to allow or deny access to the different annotations of an object, stored by different SBs, based on the users or groups that have created them and the access control policies that are defined for each piece of metadata.

4 Conclusions

In this paper we have described the requirements posed by some of the existing and the envisaged metadata-intensive distributed applications. These requirements can be summarised as follows:

- Applications may require their explicit metadata to be encoded in multiple forms, supplied by multiple parties and coming from different contexts. Moreover, metadata may need to be in different physical locations.
- Applications may use heterogeneous metadata storage and query technologies, each of which has specific advantages over the others (e.g. Sesame is good at query performance, Jena has a rich API, etc.). To ease metadata sharing, common technology-independent means for metadata access (metamodels and protocols) become necessary.
- Metadata normally evolves, either because of new metadata generation means or because of the evolution of the resources or knowledge entities that it refers to. Adequate means to deal with this evolution have to be made available.
- Access to metadata may need to be secured with different levels of granularity and different access control policies.

We have shown how the approach followed in S-OGSA complies with the previous requirements without the need to create expensive and difficult-to-maintain ad-hoc solutions for each of them. Basically, S-OGSA proposes to treat metadata as a first-class resource (Semantic Binding) in the application, so that standard resource management and sharing solutions used in the context of distributed applications can be easily applied to it. This includes aspects like service orientation, lifecycle management, security, etc.

The S-OGSA approach is being used in the development of different types of applications, all of which are characterised by being metadata-intensive and by needing support for some of the previous requirements. The S-OGSA reference implementation is being used in several system prototypes: a satellite quality image analysis system [11], an insurance settlement system[8], and an information service for the EGEE Grid [12]. It will be also used in a coral bleaching alert system (Semantic Reef) [13] and in a clinical diagnosis advice system[9].

5 Future Work

Our future work will be devoted to improve our S-OGSA reference implementation, which includes the Semantic Binding Service suite presented in Section 3. We will consider the additional requirements that the early adopters of S-OGSA are providing and create an implementation with industrial standards.

Among the aspects that will be improved, we can cite the following:

- *Security.* We will provide a set of pre-defined security configurations that cover the most common metadata management security aspects used by applications.
- *Naming.* Or current implementation uses WS-Addressing EndPoint References (EPRs) to identify Semantic Bindings and the Resources that they refer to, and URIs to identify the Knowledge Entities that their content

[8] http://www.insurancegrid.org/
[9] http://www.biopattern.org/

uses. There are multiple ways of identifying entities in a distributed environment (URIs, ARK Identifiers, LSIDs). To be able to support all, we will implement a metadata identification model that uses WS-Naming, which builds on WS-Addressing and extends it with use of URIs.
- *Semantic Binding Housekeeping Service*. We will build a configurable SB Housekeeping Service that gives support to the most common behaviours for metadata evolution found in applications.
- *Support for more forms of metadata*. We will give support to several forms of annotation (besides RDF, which is the one currently used), such as social tags, natural language comments, other ontology languages, etc.

Finally, as part of our future work we will also evaluate our system with respect to other metadata management systems, in terms of memory consumption, query execution performance, etc., in distributed settings.

Acknowledgements

This work is supported by the EU FP6 OntoGrid project (STREP 511513) funded under the Grid-based Systems for solving complex problems, and by the Marie Curie fellowship RSSGRID (FP6-2002-Mobility-5-006668). We also thank the other members of the OntoGrid team at Manchester for their helpful discussions: Wei Xing, Ian Dunlop and Ioannis Kotsiopoulos.

References

1. J. Greenberg, "Metadata and the World-Wide-Web," in *Encyclopedia of Library and Information Science*, 2003, pp. 1876–1888.
2. O. Corcho, P. Missier, P. Alper, S. Bechhofer, and C. Goble, "Principled metadata management for next generation metadata-intensive systems," in *4th European Semantic Web Conference (ESWC2007). Submitted*, Innsbruck, Austria, 2007.
3. L. Stojanovic, "Methods and tools ontology evolution," Ph.D. dissertation, Univ Karlsruhe (TH), 2004.
4. M. Klein, "Change management for distributed ontologies." Ph.D. dissertation, Vrije Universiteit Amsterdam, 2004.
5. M. Klein and N. F. Noy, "A component-based framework for ontology evolution," in *Proceedings of the Workshop Ontologies and Distributed Systems at the International Joint Conference on Artificial Intelligence (IJCAI'03)*, Acapulco, Mexico, 2003.
6. Ó. Corcho, P. Alper, I. Kotsiopoulos, P. Missier, S. Bechhofer, and C. A. Goble, "An Overview of S-OGSA: A Reference Semantic Grid Architecture," *Journal Web Semantic*, vol. 4, no. 2, pp. 102–115, 2006.
7. I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, and J. V. Reich, *The Open Grid Services Architecture, Version 1.5*, gfd-i.080 ed., GGF, July 2006, http://forge.gridforum.org/projects/ogsa-wg.
8. P. Missier, P. Alper, O. Corcho, I. Kotsiopoulos, I. Dunlop, W. Xing, S. Bechhofer, and C. Goble, "Managing semantic grid metadata in S-OGSA," in *Cracow Grid Workshop 2006. Submitted*, Cracow, Poland, 2006.

9. M. Parkin, S. van den Burghe, O. Corcho, D. Snelling, and J. Brooke, "The Knowledge of the Grid: A Grid Ontology," in *Proceedings of the 6th Cracow Grid Workshop*, Cracow, Poland, October 2006.
10. K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke, and W. Vambenepe, "Web Services Resource Framework (WSRF)," Globus Alliance and IBM," Technical report,, March 2005.
11. M. Sánchez-Gestido, L. Blanco-Abruña, M. de los Santos Pérez-Hernández, R. González-Cabrero, A. Gómez-Pérez, and O. Corcho, "Complex data-intensive systems and semantic grid: Applications in satellite missions," in *Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing (e-Science 2006), Amsterdam, The Netherlands*, December 2006.
12. W. Xing, O. Corcho, C. Goble, and M. Dikaiakos, "Active ontology: An information integration approach for highly dynamic information sources," in *Submitted to ESWC2007*, May 2007.
13. T. S. Myers, "The Semantic Reef: Managing complex knowledge to predict coral bleaching on the great barrier reef." in *In proceedings of AusGrid 2007*, 2007, http://eprints.jcu.edu.au/1131.