Grid Metadata Management: requirements and architecture

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 annotations of Grid resources can potentially be used for a number of purposes, including accurate resource allocation to jobs, discovery of services, and precise retrieval of information resources. In order to realize this potential on a large scale, various aspects of metadata must be managed. These include uniform and secure access to distributed and independently maintained metadata repositories, as well as management of metadata lifecycle. In this paper we analyze these issues and present a service-oriented architecture for metadata management, called S-OGSA, that addresses them in a systematic way.**

I. Introduction

Successful, large-scale management of Grid resources, i.e., data and services, increasingly involves the use of annotations to describe various aspects of those resources, be it the availability of computing or memory resources, or the function and interface to a Grid service. With the term "metadata" we denote, in a broad sense, any data that describes resources, and more specifically, "structured data about an object that supports functions associated with the designated object" [1]. Thus, metadata is structured according to some schema, and it is used to provide a functional or behavioral description of objects, or resources.

Annotations of resources may potentially serve a multitude of purposes, from the correct allocation of computational resources to jobs, to discovery of services, to accurate retrieval of information resources. The adoption of the Semantic Web paradigm in particular with its associated standard languages and technologies for metadata and knowledge representation (i.e. RDF(S) [2] and OWL [3]) has been viewed as the key enabler of the annotation of Grid resources and the exploitation of this mark-up. According to this vision, dubbed the Semantic Grid [1], describing various aspects of Grid resources in terms of agree-upon formal ontologies makes it possible to generate "semantic" annotations that can be interpreted using concepts from those ontologies. This results in annotations that are predictable both in structure and in content, without being too rigid. This makes them more easily interoperable than free-format metadata with arbitrary content. Furthermore, formal techniques for automated reasoning may sometimes be leveraged to enhance the effectiveness of resource management tasks, like those listed earlier. A number of annotation tools (a survey can be found in [4]) are currently available to produce metadata, and various technologies can be used to manage those annotations, including Jena, Sesame, Boca, Oracle-RDF, Annotea, Technorati, etc.

With this recognition of the importance of metadata annotations in the Grid environment, however, also come new issues of large-scale metadata access, interoperation, and reuse. In this paper, we argue for a metadata management architecture that takes into account the specific requirements of (i) uniform access to distributed metadata produced by independent organizations; (ii) metadata lifecycle management; and (iii) uniform authorisation mechanisms, in order to support a new generation of metadata-intensive applications.

Firstly, we present some of the known metadata management issues in the context of the ^{my}Grid project[2], which provides a rich service-based middleware infrastructure for the bioinformatics domain (Section II). Secondly, we analyze some of the existing approaches and technologies that can be used to address these issues (Section IV). Our main contribution is a novel proposal for managing metadata as first-class resources in distributed systems, known as S-OGSA (Section V). Designed as a non-disruptive extension to the OGSA architecture, S-OGSA provides a service-oriented approach to large-scale, uniform metadata management on the Grid. After presenting the core S-OGSA service, called "Semantic Binding Service" (SBS), we conclude by arguing that the proposed architecture addresses the management issues listed above. A prototype version of the SBS has been implemented, and is deployed as a Grid service within the Globus Toolkit v.4 service container.

II. A Motivational Example: Metadata Management in the ^{my}Grid project

The ^{my}Grid project provides a good example for the type of metadata management requirements that we address in S-OGSA. ^{my}Grid provides bioinformaticians with a suite of middleware tools and services for assembling and executing scientific workflows that involve access to a variety of databases and data analysis tools.

In this context, metadata is pervasive: on one hand, genes and gene products are annotated, possibly by human curators, in order to provide a rich description of function or structure.

[1] http://www.semanticgrid.org

[2] http://www.mygrid.org.uk/

1-4244-1560-8/07/$25.00 © 2007 IEEE

On the other hand, annotation of services and workflows provides the necessary metadata to support tasks such as resource discovery and the management of workflow results [5].

A number of metadata management requirements emerge from the ^{my}Grid experience. Firstly, just as resources are managed by independent organizations, so *metadata is created by multiple parties*, i.e., service providers, expert curators and biologists, and comes in different formats, ranging from free text, Javadoc, curated RDF(S)-based semantic descriptions. In order to make resource discovery possible, semantic metadata is made accessible through public registries, searchable via web services, using NG4J[3] and Sesame2[4]. Textual metadata is collected on human-readable web pages.

Secondly, *referring to resource and its metadata requires a uniform naming scheme*. This is not always available, however. For non-semantic metadata, for instance, resources are mostly referred to using web service end-points or URLs to workflow scripts. For semantic metadata, ^{my}Grid adopts internal Life Science Identifiers (LSIDs) [6], i.e., Uniform Resource Names (URNs) with well-defined resolution protocols.

Thirdly, there is an *issue of change*, not only in the availability of resources, but also in knowledge. Domain ontologies in particular are subject to change, reflecting the ongoing knowledge discovery process in biology and the life sciences. This imposes a high maintenance effort on the associated metadata. This dynamicity calls for a general bookkeeping framework for resources, knowledge and metadata, that is able to propagate change events through the middleware and cause appropriate maintenance actions to be taken.

Finally, access control to metadata resources is an emerging issue, prompted by the increasing volume of published scientific results, the visibility of which should be determined by the controlling organization.

In ^{my}Grid, most of these issues are currently addressed through ad hoc solutions, including customised ways to associate metadata with resources, specialised metadata management stores and services that extend existing RDF storage and querying technologies, customised access control services, etc. In the rest of this paper, we propose a generalized, technology-independent metadata abstraction layer that allows for a more flexible integration, and thus better reuse, across providers.

III. New requirements for Metadata Management

Metadata-intensive middleware platforms like ^{my}Grid help us highlight some of the critical issues in metadata management:

1) **Distribution, uniformity of access.**
   Just as data is naturally distributed and managed by independent organizations, so is its associated metadata, i.e., in the form of data annotations. Currently, most metadata management systems, some of which are described in the next section, provide repositories that are designed for centralized use and offer proprietary APIs, with metadata consumers and producers acting as local-access clients.
   There is therefore an initial need to offer uniform access to metadata repositories and ontologies, as a pre-requisite to providing integrated views over large collections of distributed metadata. We are going to address this requirement using a service-oriented approach, by defining a new, single service interface that can be implemented by most providers.

2) **Metadata should be accessible using service-oriented protocols.**
   Metadata can be structured in different ways and take up many different formats, even in the case of a single form of metadata (e.g. RDF(S)). Although it would not be feasible to imagine a common data model, some common denominator for managing metadata is needed. As we will see, minimally we need to maintain the association between resources and metadata, and to support query languages on metadata that are model-specific, for instance SPARQL for RDF metadata.

3) **Management of metadata lifecycle.**
   Metadata is dynamic in nature, and its lifetime may vary greatly by orders of magnitude. 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 related metadata [7], [8], [9]. Metadata should be maintained uo-to-date in a cost-effective manner. This includes maximising the automation of different aspects of the knowledge lifecycle, managing the evolution and change of metadata and knowledge models in distributed contexts, and synchronising the evolution of all these related entities by means of notification mechanisms.
   Thus, this requirement concerns the automation of all aspects of metadata lifecycle management. In our model, this is addressed by assuming that metadata is stateful and that a manager can detect state changes and propagate them using a notification infrastructure, as described in Section V-C.

4) **Granular and uniform access control to metadata.**
   There are two main issues regarding access control: firstly, permission to access and manipulate metadata elements should be defined at different levels of granularity, depending on the metadata format. When using RDF, for example, it should be possible to define access control rules for a single statement, as well as on an entire named graph, i.e., an atomic unit consisting of a graph of interconnected RDF statements. Secondly, access control mechanisms should be uniform across metadata repositories, regardless of differences in the access mechanisms.
   With respect to these issues, current access mechanisms to metadata repositories are still limited. For instance,

[3] http://sourceforge.net/projects/ng4j/
[4] http://www.openrdf.org/

Sesame provides built-in user/role-based access control at the level of an entire repository, while Jena has no built-in access-control support since it assumes that this can be provided by the underlying database technology. Furthermore, all these access control mechanisms adopt proprietary security APIs and conventions. When trying to access multiple repositories, the need to manage multiple sign-ons creates an obvious overhead in the client.

We note that service-oriented and Grid architectures face similar issues, and that current solutions are available to address them. To this end, we propose to investigate the suitability of current standards[5] and open-source reference implementations[6] for global user identification, single sign-on, communication encryption and representation and decision of resource-sharing policies.

IV. Current Approaches to Metadata Management

We now review some of the approaches and technologies available for metadata management in the Web and on the Grid, including mechanisms for linking metadata to resources, as well as storage and retrieval functionality and access protocols.

A. Metadata and Resource linkage

On the Web, the association of metadata to the entities that it describes is mostly ad-hoc or implicit. The commonly adopted approach is to either embed metadata in HTML or XHTML documents [10], i.e., by making use of the `<meta>` tag, or to link [11] to it. Using this approach, an extensible set of properties including author, expiration date, keyword lists, etc. can be used to annotate documents. In the linking approach, the `<link>` tag is used to refer to files containing metadata about a particular XHTML document. The `rel`, `about` and `href` properties of this tag are used to specify the metadata document's location, type and its relationship to the subject XHTML document.

B. Metadata Storage and Retrieval

There is a large body of work in the area of RDF(S)-based metadata storage and querying technologies, with Jena[7] and Sesame among the most well-known and widely-used systems. These technologies provide rich, fine-grained APIs for manipulating and accessing RDF(S) data, as well as querying it with different RDF query languages, including the emerging SPARQL[8].

Sesame also supports access to remote repositories through Java RMI, and it also exposes a subset of its API within a REST-based web service interface, so as to allow for platform and programming language independence. Work is also under way on enhancing RDF storage and querying techniques in the areas of contextualization, distribution and scalability. NG4J and Sesame2 provide extensions for grouping RDF statements based on contextual information (e.g. ownership). In [12] a mediator layer on top of Sesame is described, that provides clients with a global and location-transparent view of RDF data in multiple repositories. In [13], [14] the data structures and protocols of P2P networks are used to store and query large amounts of RDF data in a fault tolerant manner.

The storage and retrieval of social tags about Web resources is not standardised, although some specific database schemas and technologies are gaining acceptance and becoming *de-facto* standards[9].

C. Metadata Access Protocols

The notion of serving ontologies and metadata has so far received little attention in the Semantic Web community, with only a few proposals in this direction. Knowledge and metadata are usually treated as a kind of Web resource, and made available through Web servers using the HTTP protocol [15]. In some cases the metadata is generated dynamically at request time from a repository and delivered to the client by means of "HTTP 303" re-directs. This approach appears to be too simplistic to support the needs of future semantic aware applications, where protocols for collective, and controlled and interoperable access and manipulation of knowledge and metadata are needed. In fact, the lack of such systems has motivated many recent research efforts[10].

More recently, access to knowledge and metadata is also considered in the SPARQL protocol, a light-weight protocol for querying RDF sources with the SPARQL language. The protocol is independent of any transport bindings, although SOAP and HTTP bindings are provided. This work solely focuses on querying and does not address RDF data manipulation; neither does it address issues of security, lifetime and others.

The Grid community has also produced work on protocols for accessing and managing knowledge and metadata. The work of the Open Grid Forum DAIS-RDF(S) group is focused on the standardization of stateful web-service access and querying over RDFS ontologies [16] and RDF instance data [17]. The set of specifications proposed by this working group provide more capabilities than those of SPARQL, although they do not address yet issues such as security or the association of metadata with resources.

D. Metadata Advanced Management Capabilities

None of the aforementioned systems manages metadata lifetime and notification of updates in distributed environments. Besides, the service-oriented metadata delivery aspect is mostly neglected in all the current approaches. A recent effort from IBM, Boca RDF[11], provides more advanced capabilities related to service-based access, named graphs, versioning

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

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

[7] http://jena.sourceforge.net/

[8] http://www.w3.org/TR/rdf-sparql-query/

[9] http://forge.mysql.com/wiki/TagSchema

[10] See for example the work being performed in the projects NeOn (http://www.neon-project.org/)and OntoGrid (http://www.ontogrid.eu/)

[11] http://ibm-slrp.sourceforge.net/2006/11/20/boca-the-rdf-repository-component-of-the-ibm-semantic-layered-research-platform/

support, notifications, authorized access, and scalability. Boca is an attempt to push RDF data management facilities to the level of maturity of well-developed data representations such as relational and XML data. Therefore it additionally provides replications and transactions over RDF data. The Boca client stack is based on the Jena API and its notification system is based on Java Messaging Service.

V. Metadata Management using Semantic Bindings

Our approach to addressing the metadata management requirements is set in the context of the Semantic Grid architecture S-OGSA, described in [18]. The Open Grid Service Architecture (OGSA) defines a core set of capabilities and behaviors for Grid systems. Semantic-OGSA (S-OGSA) extends OGSA to support the explicit handling of semantics, and defines the associated knowledge services to support a spectrum of service capabilities.

The key principle in the S-OGSA model is that *the association of Grid resources to metadata is itself a first-class Grid resource*. S-OGSA introduces a novel type of Grid resource called the *Semantic Binding* (SB), which encapsulates metadata and thus allows for its management. A SB maintains the association between one or more Grid Entities (resources or services), and one or more metadata elements. The latter can be anything that can be referenced on the Grid using a unique identity. This includes, potentially, a named graph of RDF statements, chunks of natural language, social tags, and more. Each of these associations can optionally be further contextualized with references to other meta-information, called *Knowledge Entities*. These can be ontologies, controlled vocabularies, and any other type of formal knowledge that can be used to interpret the metadata.

Consider for example the workflow in Figure 1, on the left, denoted with gs (a Grid-retrievable resource), and suppose that multiple annotations, each independently maintained from the others, have been created, as shown in the central part of the figure. Each annotation is regarded as a whole metadata element md_i and takes part in a SB of the form $SB_i = (wf, md_i)$. Furthermore, one or more knowledge entities ke_j (on the right) can optionally be included in the SB, to form a complete SB: (wf, md_i, ke_j). This arrangement accounts for natural associations between, for example, an RDF graph and an ontology, whereby the objects in the RDF statements are individual that belong to the ontology classes.

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.

A. Semantic Binding Capabilities

In [18] we describe the model used to represent Semantic Bindings, expressed as an ontology that extends the Grid ontology described in [19]. 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 2:

- *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.

These properties and functionalities are implemented in the current S-OGSA reference implementation[12], which complies with the WS-Resource Framework [20] (WSRF) family of specifications. This implementation can be deployed on the Globus Toolkit 4 platform[13] and in Apache Tomcat[14]. The implementation is being used in several system prototypes: a satellite quality image analysis system [21], an insurance settlement system[15], and an information service for the EGEE Grid [22]. It is also being integrated into the workflow provenance framework of the ^{my}Grid project.

In the following sections we give details about how we deal with the rest of requirements presented in Section III, namely the need for managing the lifetime of metadata and the notifications of metadata changes to the interested distributed parties (requirement 3) and the controlled access to metadata (requirement 4).

B. Semantic Binding Lifetime

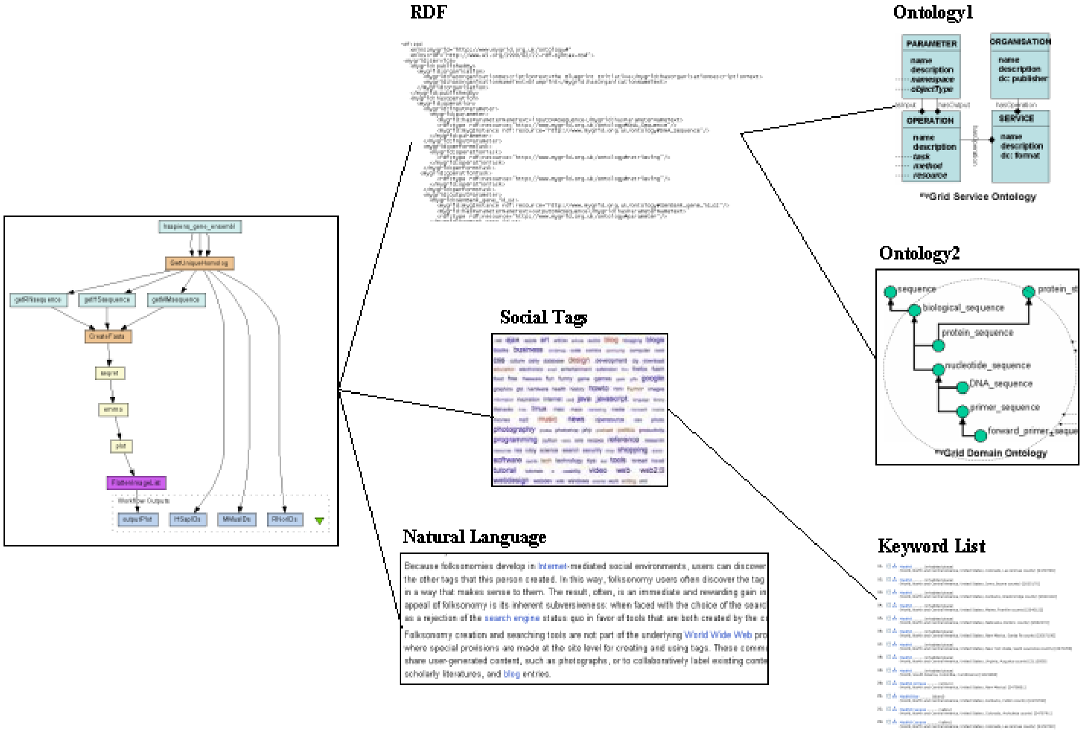
The dynamic nature of a SB is easily recognized by noting that each of the elements comprising the association are subject to changes in time: Grid resources may be updated

[12] Available at http://www.ontogrid.eu/ontogrid/downloads.jsp and at the OntoGrid CVS (instructions in the same Web page).

[13] http://www.globus.org/toolkit/

[14] http://tomcat.apache.org/

[15] http://www.insurancegrid.org/



Because folksonomies develop in Internet-mediated social environments, users can discover
the other tags that this person created. In this way, folksonomy users often discover the tag
in a way that makes sense to them. The result, often, is an immediate and rewarding gain in
appeal of folksonomy is its inherent subversiveness: when faced with the choice of the searc
as a rejection of the search engine status quo in favor of tools that are both created by the c

Folksonomy creation and searching tools are not part of the underlying World Wide Web pro
where special provisions are made at the site level for creating and using tags. These commu
share user-generated content, such as photographs, or to collaboratively label existing conte
scholarly literatures, and blog entries.

Fig. 1. Multiple Semantic Bindings for a single Grid entity (a workflow)

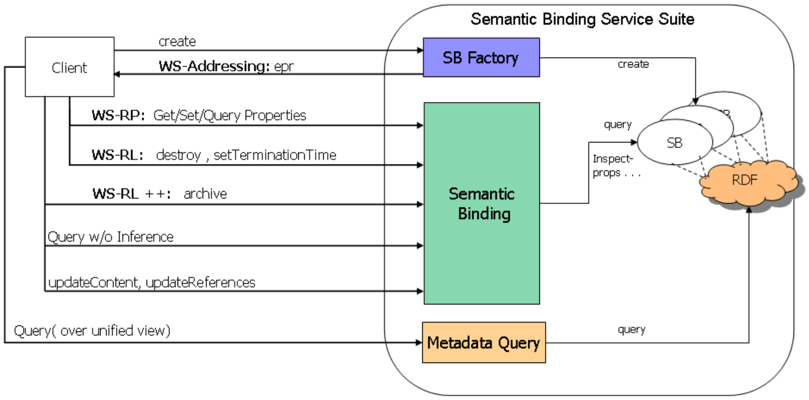

Semantic Binding Service Suite
Client
create
WS-Addressing: epr
SB Factory
create
WS-RP: Get/Set/Query Properties
WS-RL: destroy , setTerminationTime
WS-RL ++: archive
Query w/o Inference
updateContent, updateReferences
Query(over unified view)
Semantic Binding
query
Inspect-props . . .
SB
RDF
Metadata Query
query

Fig. 2. Functionality of the Semantic Binding Service

or removed, the annotations may change, and the Knowledge entities themselves may be upgraded, for example to reflect new knowledge about a particular domain, or a restructuring of existing knowledge.

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).

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 state

TABLE I

EVENTS, STATE TRANSITIONS, AND VALIDATION ACTIONS

Events	State after event	States after validation
$Res_{SB} \rightarrow Res'_{SB}$	*ValidateRes*	*Valid / Invalid*
$Res_{SB} \rightarrow \emptyset$	*ValidateRes*	*Invalid*
$KE_{SB} \rightarrow KE'_{SB}$	*ValidateKE*	*Valid / Invalid*
$KE_{SB} \rightarrow \emptyset$	*ValidateKE*	*Invalid*
$content_{SB} \rightarrow content'_{SB}$	*Valid*	N/A
$obsolete(SB)$	*Invalid*	*Valid / Invalid*
$content_{SB} \rightarrow \emptyset$	N/A	N/A
$archive(SB)$	*archived*	N/A
$destroy(SB)$	N/A	N/A

diagram associated to an SB, shown in part (a) of Figure 3, 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, as shown later.

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.

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: $Res_{SB} \rightarrow \emptyset$, $Res_{SB} \rightarrow \emptyset$. These transitions are listed in the second column of Table I. 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. In the table, this is indicated as the event $obsolete(SB)$.

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[16].

- 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. The problem of assessing the impact of ontology evolution on an existing knowledge base has been addressed in the literature [7], [9].

The possible outcomes of the validation procedures are listed in the last column of Table I. As a particular case, when the Resources or Knowledge entities are destroyed, the validation procedure is always assumed to fail, leading to an *Invalid* state. Note also that according to the table, an update to valid metadata, i.e., $content_{SB} \rightarrow content'_{SB}$ when the state is *Valid*, always results in new valid metadata. Finally, the *Archived* state indicates that a SB is still available for inspection, but it has been superseded by a more recent version.

The basic SB state machine can be extended by introducing sub-states, resulting in finer-grain definition of the behaviour of specific types of metadata. Part (b) in figure 3 shows the extended metadata state diagram in the ^{my}Grid project. The new sub-states within *Valid* allow a distinction to be made between metadata that has been reviewed by human experts (i.e., "Quality Assured"), and metadata that is awaiting QA. Note that in both cases, the metadata is indeed valid, and the sub-states add explicit information regarding the *quality* of the annotation, which some applications may want to take into account. This is important for instance when annotations are automatically generated, as in [23], requiring experts' inspection prior to their release.

Along with the state diagram, the transition table shown in Table I can be refined with specific state transition rules, as follows:

- $Res_{SB} \rightarrow \emptyset$ results in a transition to the *archived* state;
- $KE_{SB} \rightarrow KE'_{SB}$ triggers the invocation of a change detection tool, which analyzes the SB content and issues a report to the annotator;
- a transition to the *Awaiting QA* state triggers a notification to the annotator, to carry out the QA task.

C. Notification of Semantic Binding Changes

Requirement (3) above 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 includes a notification mechanism based on WS-Notification, along with a set of pre-defined topics associated to the state changes described in Table I. Application-dependent topics can be added to the set. Consumers who subscribe to a topic are notified of state changes.

Upon receving a notification, services must implement logic for reacting to the changes. Although S-OGSA services, which are domain-independent, cannot be responsible for application-specific logic, it is possible to imagine a single monitoring service that mediates between the Semantic Binding Service

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

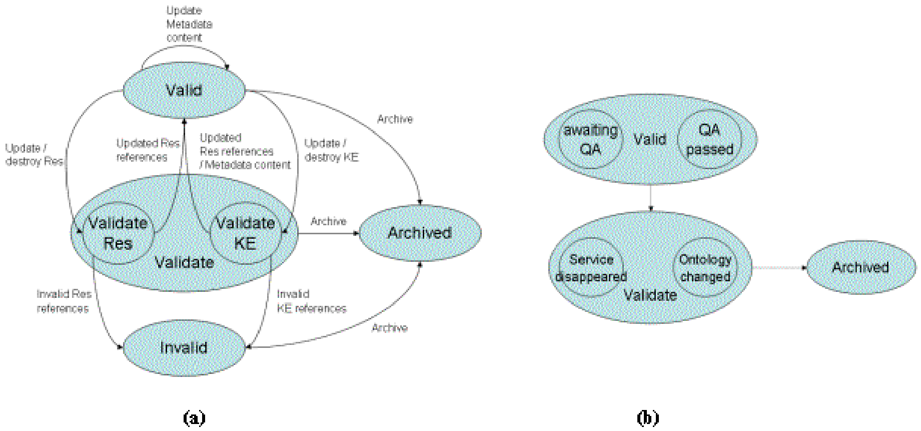


Fig. 3. State transition diagram for a generic Semantic Binding and and extended state model from myGrid

and applications that are reactive to change. Specifically, this *SB housekeeping service*, currently under development, subscribes to all standard state change topics, and is responsible for activating application-dependent re-validation procedures, including validation of the SB content, triggering of re-annotation processed, etc.. For example, in the context of ^{my}Grid, when a new annotation has been generated for a workflow, the corresponding SB changes its state to *Awaiting QA*. The SB housekeeping service is subscribed to changes in the state of SBs, and when it receives the notification of this transition, it is in charge of contacting an annotation curator (e.g., by e-mail) so that the metadata can be curated and the SB can move into a *QA passed* state.

D. Semantic Bindings Security

Requirement (4) calls for a fine-grained access control to metadata. Furthermore, other security-related aspects like securing message exchanges, authentication, etc., may need to be taken into account.

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, in the ^{my}Grid example it would be possible to allow or deny access to different annotations of a workflow, 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.

VI. Conclusions

In this paper we have outlined a few key requirements for metadata management, and described aspects of a service-oriented architecture, called S-OGSA, that addresses the requirements. 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 (meta-models and protocols) become necessary.
- Metadata is dynamic and subject to change in time. The lifetime of metadata should be managed explicitly.
- Access to metadata may need to be secured with different levels of granularity and different access control policies.

The proposed S-OGSA architecture provides a generic and application-agnostic solution to address these requirements. At its core is the notion of a Semantic Binding, a first-class Grid resource that encapsulates metadata and its association both to Grid resources and to Knowledge Entities that can be used to interpret the metadata. We have shown how the Semantic Binding Service, a new type of Grid service, can manage the lifecycle of Semantic Bindings, as well as their access control by leveraging the standard Globus GSI security framework.

The S-OGSA approach is being used in the development of different types of applications, all of which are characterized by being metadata-intensive and by needing support for some of the previous requirements.

VII. Future Work

Our future work will be devoted to improve our reference implementation, which includes the Semantic Binding Service suite presented in Section V. 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*. In addition to adopting the Globus GSI security framework, we are working on a set of pre-defined security configurations that cover the most common cases needed so that the metadata management security aspects of the applications can be easily configured.
- *Naming*. Since there are multiple ways of identifying entities in a distributed environment (URIs, ARK Identifiers, LSIDs), it is necessary to implement a unified naming scheme; WS-Naming will be used for the purpose.
- *Semantic Binding Housekeeping Service*. As mentioned, this service provides a single management point for reactive service logic that is triggered by Semantic Bindings lifecycle change events. A part of the configuration, it will be possible to launch annotation services, check the validity of semantic metadata, etc.
- *Support for more forms of metadata*. In the current implementation, the Semantic Binding Service supports RDF metadata only. This will be generalized to support additional forms of metadata, including social tags, natural language comments, other ontology languages, etc.

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] D. Brickley and R. G. (editors), "RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation," February 2004, http://www.w3.org/TR/rdf-schema/.

[3] P. Patel-Schneider, P. Hayes, and I. Horrocks, *OWL Web Ontology Language Semantics and Abstract Syntax*, World Wide Web Consortium, February 2004.

[4] Ó. Corcho, "Ontology based document annotation: trends and open research problems." *IJMSO*, vol. 1, no. 1, pp. 47–57, 2006.

[5] C. Wroe, R. Stevens, C. Goble, A. Roberts, and M. Greenwood, "A suite of daml+oil ontologies to describe bioinformatics web services and data," *Special issue of the International Journal of Cooperative Information Systems*, vol. 12, no. 2, pp. 197–224, 2003.

[6] T. Clark, S. Martin, , and T. Liefeld, "Globally distributed object identification for biological knowledgebases," *Briefings in bioinformatics*, vol. 5, no. 1, pp. 59–70, 2004.

[7] L. Stojanovic, "Methods and tools ontology evolution," Ph.D. dissertation, Univ Karlsruhe (TH), 2004.

[8] M. Klein, "Change management for distributed ontologies." Ph.D. dissertation, Vrije Universiteit Amsterdam, 2004.

[9] 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.

[10] B. Adida and M. Birbeck, "RDFa Primer 1.0: Embedding RDF in XHTML," W3C Working Draft, Tech. Rep., May 2006, http://www.w3.org/TR/xhtml-rdfa-primer/.

[11] J. Axelsson, M. Birbeck, M. Dubinko, B. Epperson, M. Ishikawa, A. N. Shane McCarron, and S. Pemberton, "XHTML 2.0," W3C Working Draft, Tech. Rep., July 2006, http://www.w3.org/TR/xhtml2.

[12] G. Adamku and H. Stuckenschmidt, "Implementation and evaluation of a distributed RDF storage and retrieval system," in *WI '05: Proceedings of the The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 393–396.

[13] M. Cai and M. Frank, "RDFPeers: A Scalable Distributed RDF Repository based on A Structured P2P Network," in *In proceedings of the World Wide Web Conference 2004*, 2004.

[14] Z. Kaoudi, I. Miliaraki, M. Magiridou, A. Papadakis-Pesaresi, and M. Koubarakis, "Storing and querying RDF data in Atlas," demo presentation, European Semantic Web Conference 2006.

[15] A. Miles, T. Baker, and R. Swick, "Best Practice Recipes for Publishing RDF Vocabularies," March 2006, http://www.w3.org/TR/swbp-vocab-pub/.

[16] M. Esteban-Gutirrez, A. Gmez-Prez, and O. Moz-Garca, "Ontology Access in Grids with WS-DAIOnt and the RDF(S) Realization," in *European Semantic Web Conference 2006, Poster presentation*, 2006.

[17] I. Kojima, "Design and Implementation of OGSA-DAI-RDF," in *GGF16 Semantic Grid Workshop*. Athens, Greece: Global Grid Forum, February 2006.

[18] 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.

[19] 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.

[20] 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.

[21] M. Snchez-Gestido, L. Blanco-Abrua, M. de los Santos Prez-Hernndez, R. Gonzlez-Cabrero, A. Gmez-Prez, 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.

[22] W. Xing, O. Corcho, C. Goble, and M. Dikaiakos, "Active ontology: An information integration approach for highly dynamic information sources," in *Poster, Procs. ESWC2007*, May 2007.

[23] K. Belhajjame, S. M. Embury, N. W. Paton, R. Stevens, and C. A. Goble, "Automatic annotation of web services based on workflow definitions." in *International Semantic Web Conference*, 2006, pp. 116–129.