

An overview of S-OGSA: a Reference Semantic Grid Architecture

Oscar Corcho, Pinar Alper, Ioannis Kotsiopoulos, Paolo Missier,
Sean Bechhofer and Carole Goble

School of Computer Science
The University of Manchester,
Manchester, UK

<ocorcho, penpecip, ioannis, pmissier, seanb, carole>@cs.man.ac.uk

Abstract

The Grid's vision, of sharing diverse resources in a flexible, coordinated and secure manner through dynamic formation and disbanding of virtual communities, strongly depends on metadata. Currently, Grid metadata is generated and used in an ad hoc fashion, much of it buried in the Grid middleware's code libraries and database schemas. This ad hoc expression and use of metadata causes chronic dependency on human intervention during the operation of Grid machinery, leading to systems which are brittle when faced with frequent syntactic changes in resource coordination and sharing protocols.

The Semantic Grid is an extension of the Grid in which rich resource metadata is exposed and handled explicitly, and shared and managed via Grid protocols. The layering of an explicit semantic infrastructure over the Grid Infrastructure potentially leads to increased interoperability and greater flexibility.

In recent years, several projects have embraced the Semantic Grid vision. However, the Semantic Grid lacks a reference architecture or any kind of systematic framework for designing Semantic Grid components or applications. The Open Grid Service Architecture (OGSA) aims to define a core set of capabilities and behaviours for Grid systems. We propose a Reference Architecture that extends OGSA to support the explicit handling of semantics, and defines the associated knowledge services to support a spectrum of service capabilities. Guided by a set of design principles, Semantic-OGSA (S-OGSA) defines a model, the capabilities and the mechanisms for the Semantic Grid.

We conclude by highlighting the commonalities and differences that the proposed architecture has with respect to other Grid frameworks

Author Keywords

Grid; Semantic Grid; Architecture; Explicit Metadata, Semantics

1 Introduction

The Grid aims to support secure, flexible and coordinated resource sharing by providing a middleware platform for advanced distributed computing [1]. Consequently, the Grid's infrastructural machinery aims to allow collections of any kind of resources—computing, storage, data sets, digital libraries, scientific instruments, people, etc—to easily form Virtual Organizations (VOs) that cross organizational boundaries in order to work together to solve a problem.

Computational Grids are the most mature kinds of Grids, harnessing available compute power to support compute-intensive analysis applications. Whereas these Grids present the illusion of a single virtual computer to an application, Data Grids present a single virtual data store that is in reality distributed and multi-located. Portals provide a way for application developers and users to submit their compute job or their query. On top of these middleware-oriented Grids, reside the Application Grids, which aim to present the illusion that applications work together when in reality they do not. To ease the development of all these types of Grids, the Grid community, through the Global Grid Forum, has developed a reference architecture, the Open Grid Service Architecture (OGSA). This architecture addresses the need for standardization by defining a set of core capabilities and behaviours that address key concerns in Grid systems.

A Grid depends on understanding the available resources, their capabilities, how to assemble them and how to best exploit them. Thus Grid middleware, and the Grid applications they support, thrive on the metadata that describes resources in all their forms, the VOs, the policies that drive them and so on, together with the knowledge to apply that metadata intelligently.

In current Grids, this metadata is commonly managed in an ad hoc implicit way or buried in code libraries. The arbitrary expression and use of knowledge causes Grid middleware to be more prone to syntactic changes, less (if at all) interoperable, more dependent on extensive human effort for deployment configuration and maintenance, and less shareable. This seriously hampers the progress towards flexible, adaptable and interoperable Grid computing as envisaged in [1].

The Semantic Grid [2] is a recent initiative to expose semantically rich information associated with Grid resources to build more intelligent Grid services. In the last few years, several projects [3, 4] have embraced this vision and there are already successful pioneering applications that combine the strengths of the Grid and of semantic technologies, as described in [2]. The idea is to make structured semantic descriptions real commodities and visible first class citizens with an associated identity and behaviour. We can then define mechanisms for their creation and management, and protocols for their processing, exchange and customization. The languages used to encode the semantic descriptions (from natural language text right through to logical-based assertions) and the structure and content of the descriptions themselves may vary from application to application.

Semantic Web, Semantic Web Services and Semantic Grid

In practice, work on Semantic Grids to date has primarily focused on the introduction of technologies from the Semantic Web and Semantic Web Services initiatives to the Grid.

On the one hand, Semantic Web (SW) research has produced a substantial body of work focused on providing the essential language and tooling support to build conceptual models (i.e. ontologies, rules) and structured web resource metadata that adheres to these models. The interest of the SW community regarding what could be modelled using conceptual models has been on a case study basis where modelling the knowledge of certain domains has been seen as test-cases for SW languages and tools. As to how the conceptual models and accordingly typed metadata could be used, the Semantic Web has investigated smart discovery of web resources, specifically smart web search.

On the other hand, the Semantic Web Services (SWS) community has been more involved in investigating what form of conceptual models are needed to describe best a particular type of resource, namely web

services, and how these conceptual models could be used to undertake increase mechanisation of the activities related to web services, namely discovery, composition, negotiation and enactment.

These technologies could be applied to the Grid in a variety of ways. For instance, the background knowledge and vocabulary of a Grid middleware component could be captured in ontologies (for example a model of a VO [5, 6]). Metadata could be used to label Grid resources and entities with concepts, for example describing a data file in terms of the application domain where it is used. Rules and classification-based reasoning mechanisms could be used to generate new metadata from existing metadata, for example describing the rules for membership of a VO and reasoning that a potential member's credentials are satisfactory. Furthermore, activities like Grid service discovery or negotiation of service level agreements, among others, can be potentially enhanced using the functionalities provided by Semantic Web Service technologies.

The Semantic Grid activities have been in a phase of exploratory experimentation rather than one of systematic investigation and architectural design. Thus, the Semantic Grid currently lacks a reference architecture, or a systematic approach, for designing Semantic Grid components and applications. Furthermore, there is a lack of principled delivery mechanisms that would enable adoption, use and deployment of SW and SWS techniques and technologies in the Semantic Grid. As part of the EU-IST project OntoGrid¹, we aim to fill this gap by proposing a reference architecture called Semantic-OGSA (or S-OGSA for short), which is based on the existing OGSA. This proposal is in line with the the SOKU concept (Service-Oriented Knowledge Utilities), expressed in [7] as the current European vision and research directions for future Grids.

The rest of the paper is organised as follows. In Section 2 we describe the Open Grid Service Architecture, which S-OGSA is based on, and outline the requirements and design principles behind S-OGSA. In Section 3 we describe our reference architectural proposal in detail introducing S-OGSA by focusing on its characterizing entities and their inter-relationships and the category of services related to these entities. In Section 4 we describe the mechanisms with which S-OGSA conceptual elements could come into existence in an infrastructure specific instantiation of the architecture. In Section 5 we survey related work on architectural proposals within Semantic Grid and Semantic Web Service projects. We conclude in Section 6 with a summary of our contributions and outline current and future work needed in adopting and evaluating the architecture.

2 Grid Reference Architectures: OGSA and S-OGSA

“The Grid” is a heavily overloaded term to refer to anything from high-performance computing to cycle harnessing or even web-based application integration. Motivated by this trend, Grid researchers have made attempts to provide precise definitions for the Grid [8, 9]. [9] outlines the **characterizing** aspects of the Grid as follows:

1. Grid focuses on sharing distributed resources in a well-controlled and mutually fair manner by constructing a **virtual pool**. The users of the pool have very little or no a priori knowledge about the actual (i.e. physical) state, type and features of resources. Furthermore, this virtual pooling of resources should allow users to **utilize temporarily** resources, which they might not directly access otherwise.
2. In order to achieve (1), abstracting the two major entities in the environment, namely **Users** and **Resources**, at the virtual pool level is necessary. Mapping these abstracted notions of Users (e.g. Global identities represented with Certificates) to local identities (e.g. a database role), and Resources (e.g. a virtual data item) to actual resources (e.g. a file on disk) at the time of resource utilization is also needed.

In addition to the above definitions, [8] identifies the following requirements for Grid systems, each stemming from a group of use-cases [10] from e-Science and e-Business domains: interoperability, optimized allocation, adaptivity, manageable task execution, scalability, secure operation, high-availability, extensibility and ease of use.

¹ <http://www.ontogrid.net/>

The Open Grid Services Architecture (OGSA) [8] is the result of a standardization effort, now being sustained by the Grid's standards body, namely the Global Grid Forum (GGF). OGSA aims to define a core set of capabilities and behaviours for Grid systems. For this, OGSA takes the Grid definition of [8] and the Web Service based realization of the Grid [11] as a base and maps the aforementioned requirements to the following capabilities (i.e. categories of services) that could be put together to develop Grids. These are: Infrastructure Services; Data Services; Resource Management Services; Execution Management Services; Security Services; Self Management Services; and Information Services (blue boxes in Figure 1).

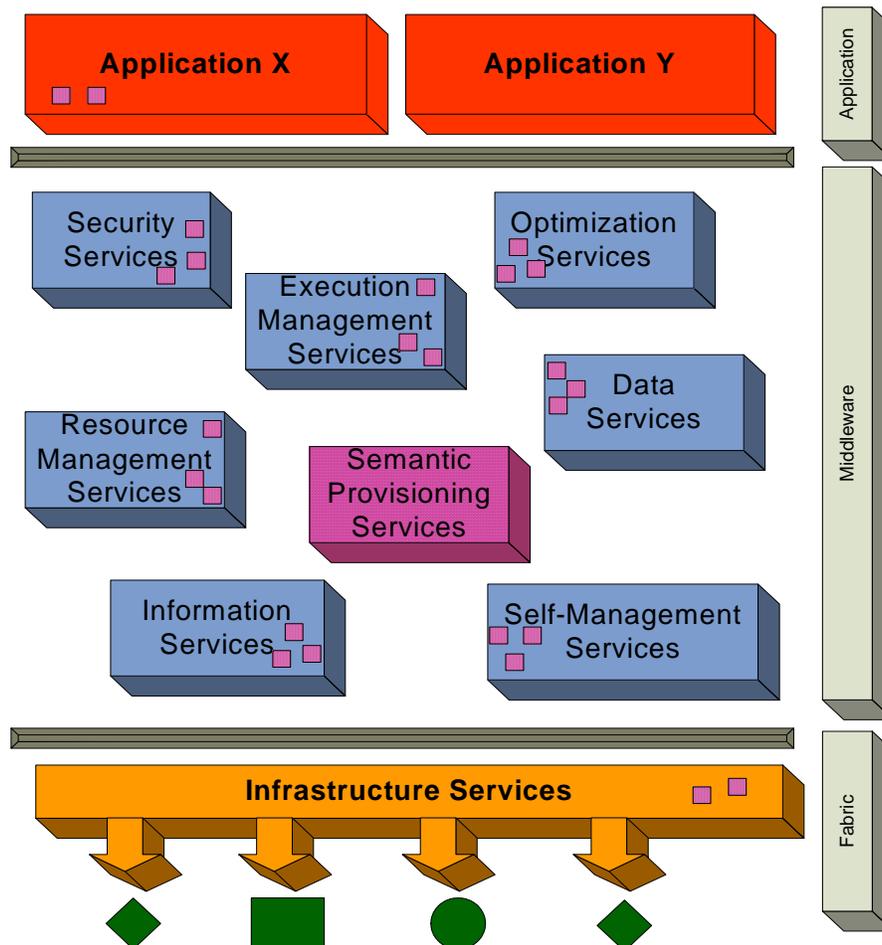


Figure 1. A Simplified View of the Architecture Proposed by OGSA (blue), with the additional components from S-OGSA (pink), described in section 3.

OGSA provides a logical 3-tiered view of distributed environments realized by the use of Grid systems. This view is also given in Figure 1. The base layer is the *fabric* where different types of resources are virtualized through web-services. OGSA covers the middle layer, which is composed of core categories of services also known as the *Grid middleware*. The middleware services are not organized in a layered architecture; instead they interact with each other while delivering their associated capability. The top layer is where the *applications* reside. Applications make use of the Grid middleware to undertake their activities.

OGSA proposes the use of Web services as the method for virtualising Grid resources. While there is a consensus on service-based virtualization, there are variations in the realizations of this view. Major service-oriented middleware providers EGEE², OMI³, and Globus⁴ supply frameworks [8, 12] for

² EGEE: Enabling Grids for E-science in Europe. <http://public.eu-egee.org/>

deploying web services with the port types through which resources are accessed. By either using or introducing new web service specifications, or by introducing their own conventions into existing ones, these frameworks attempt to address certain requirements that appear due to the association of a Web service with a Grid resource. First, and foremost, grid resources exposed via services are not restricted to application software as is the case with regular web services. Examples of different types of resources are a hard disk or even a telescope. Furthermore these resources have state and properties, which need to be exposed, for example the remaining quota of a disk or the default steering direction of a telescope. Moreover, resources also have a lifetime, which, through well-defined manageability interfaces, could be controlled. In terms of representation of descriptive information about resources (i.e. to deliver resource properties, state and lifetime information), these frameworks are deeply oriented towards the use of XML Schema based conceptualizations and XML based representations for delivery of resource property, state and lifetime.

2.1 Design Principles for a Reference Semantic Grid Architecture

As outlined in the introduction, the Semantic Grid currently lacks a Reference Architecture or any kind of systematic framework for designing Semantic Grid components or applications. OGSA aims to define a core set of capabilities and behaviours for Grid systems [8]. Our approach extends OGSA by defining a lightweight mechanism that will allow for the explicit use of semantics along with the associated knowledge services to support a spectrum of service capabilities.

Semantic-OGSA (S-OGSA) is guided by six general design principles we have devised based on our observations on fundamental issues in Semantic Grid research [13]. These are:

1. Parsimony of architectural elements. The architectural framework should be as lightweight as necessary and should minimise the impact on legacy Grid infrastructure and tooling. We believe this is crucial to the adoption of our approach. Furthermore, it should not impose the vocabulary or the structure to be used in the semantic descriptions, since these will be application or middleware dependent, though a basic set of reusable vocabularies can be provided, related to different aspects of the model.

2. Extensibility of the framework. Rather than defining a complete and generic architecture, define an extensible and customisable one.

3. Uniformity of the mechanisms. Semantic Grids are Grids, so any S-OGSA entity⁵ included in the architecture will be OGSA-observant. OGSA observance brings about the following expectations:

- a) Similar to the Grid resources they are associated with, knowledge and metadata should exhibit manageability aspects. Semantic descriptions could have state and soft state characteristics – they have a lifetime and may change during their life.
- b) S-OGSA must encapsulate both stateless and stateful Grid services, as OGSA does.
- c) Knowledge services in S-OGSA are OGSA-observant Grid services. For instance, metadata stores and ontology services are just special kinds of data services, hence we propose the adoption of the OGSA-DAI specification for their deployment, so that they can potentially exploit other data grid capabilities.

4. Diversity of semantic capabilities. A dynamic ecosystem of Grid services ranging over a spectrum of semantic capabilities should coexist at any one time. Grid entities do not *need* to be Semantic Grid entities. Semantic capability may be possible for some Grid resources all of the time, and maybe all Grid resources some of the time, not all resources all of the time. Entities in the Semantic Grid are thus classified as:

- a) *Ignorant* of the explicit semantics associated to another entity.
- b) *Aware* that another entity has explicit associated semantics but *incapable* of processing it.
- c) *Aware* that another entity has explicit associated semantics and *capable* of processing it, partially or completely.

³ Open Middleware Infrastructure Institute. <http://www.omii.ac.uk/>

⁴ The Globus Alliance. <http://www.globus.org/>

⁵ We use the terms Semantic Grid entity and S-OGSA entity interchangeably.

5. Heterogeneity of semantic representation. Any resource's property may have many different semantic descriptions, and each of them may be captured (or not) in different representational forms (text, logic, ontology, rule).

6. Enlightenment of services. Services should have a straightforward migration path that enables them to become knowledgeable. The cost involved in the migration to the Semantic Grid must be minimised in order to improve the impact and uptake of Semantic Grid, and to take advantage of current tooling and services. Thus:

- a) S-OGSA should have minimal impact on adding explicit semantics to current Grid entity interfaces or on Grid services that are ignorant of Semantic Grid entities;
- b) Grid entities should not break if they can consume and process Grid resources but cannot consume and process their associated semantics (that is, if they are aware of the semantics but incapable of processing it);
- c) If a Grid entity understands only part of the knowledge it consumes it should be able to use it as a best effort (that is, there are different degrees of awareness and semantic processing capabilities).
- d) During their lifetime, Grid entities can incrementally acquire, lose and reacquire explicit semantics.

Finally, since our aim is to develop S-OGSA as a conceptual reference architecture, it should apply equally with a grounding to WSRF⁶ [14], to WSDM [15], to Microsoft's WS-Management stack⁷, etc.

S-OGSA has three main aspects: the *model* (the elements that it is composed of and their interrelationships), the *capabilities* (the services needed to deal with such components) and the *mechanisms* (the elements that will enable delivery when deploying the architecture in an application, grounded to a Grid platform). In the following two sections we describe the architecture, relating our design decisions to the list of desiderata presented in this section. The division into two sections is due to the fact that in the first section we concentrate on the platform independent – conceptual - parts (the model and capabilities), while in the second we concentrate on the architecture grounding (the mechanisms).

3 S-OGSA Model and Capabilities

3.1 S-OGSA Model

Explicit semantics may be used to represent knowledge in the Grid environment, the source of which could come from each of the 3 tiers (i.e. application, middleware, fabric) of OGSA. Application specific knowledge is out of the scope of our S-OGSA model proposal, as it depends on the application domain (e.g., scientific data) and cannot be easily generalized. The knowledge of the middleware and the fabric layers, however, are of importance in S-OGSA. A standardized overall model of the Grid and its basic concepts (e.g. VOs and resources) are not currently available. There exist, however, project-specific modelling efforts [16], [17] capability focused models emerging from the Global Grid Forum (e.g. CIM [18, 19], DFDL [20], JSDL [21]) and a vocabulary associated with OGSA.

A definition of the Semantic Resources that are supplied and consumed amongst the services extends the general model of the Grid. With S-OGSA we introduce the notion of Semantics into the model of the Grid (see Figure 2 for a graphical depiction):

- **Grid Entities** (G-Entities) are anything that carries an identity on the Grid, including resources and services [22].
- **Knowledge Entities** (K-Entities) are special types of Grid Entities that represent or could operate with some form of knowledge. Examples of Knowledge Entities are ontologies, rules, knowledge bases or even free text descriptions that encapsulate knowledge that can be shared. Knowledge services are those that provide access to or operate over those knowledge resources, examples could be rule engines, automated reasoners and so on.
- **Semantic Bindings** (S-Bindings) are the entities that come into existence to represent the association of a Grid Entity with one or more Knowledge Entities. Existence of such an association transforms the

⁶ http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf

⁷ <http://msdn.microsoft.com/ws/2004/10/ws-management/>

subject Grid entity into a **Semantic Grid Entity**. In analogy with the Semantic Web approach, Semantic Bindings represent metadata assertions on web resources. In our model Semantic Bindings are first class citizens as they are modelled as Grid resources with an identity and manageability features as well as their own metadata. This is done according to our design principle of *uniformity*, where the new entities in the model must exhibit manageability aspects. As shown in the cardinalities of the relationships in figure 3, Semantic Bindings refer to one Knowledge Entity or a group of Knowledge Entities, and to one Grid Entity or a group of Grid Entities. At the same time, Knowledge Entities and Grid Entities can exist without the need to be related to any Semantic Binding.

- **Semantic Grid Entities** (SG-Entities) are those Grid Entities that are either the subject of a semantic binding, are themselves a semantic binding, or a Knowledge Entity (this definition includes all the entities shown in Figure 3). In keeping with some of our design principles (namely *diversity*, *heterogeneity*, and *enlightenment*), Grid entities can be associated with zero, one or multiple knowledge entities of different forms and capabilities, simultaneously, and can acquire and discard associations with knowledge entities through their lifetime. It should be noted that S-OGSA does not prescribe any specific technology for the realisation of these.

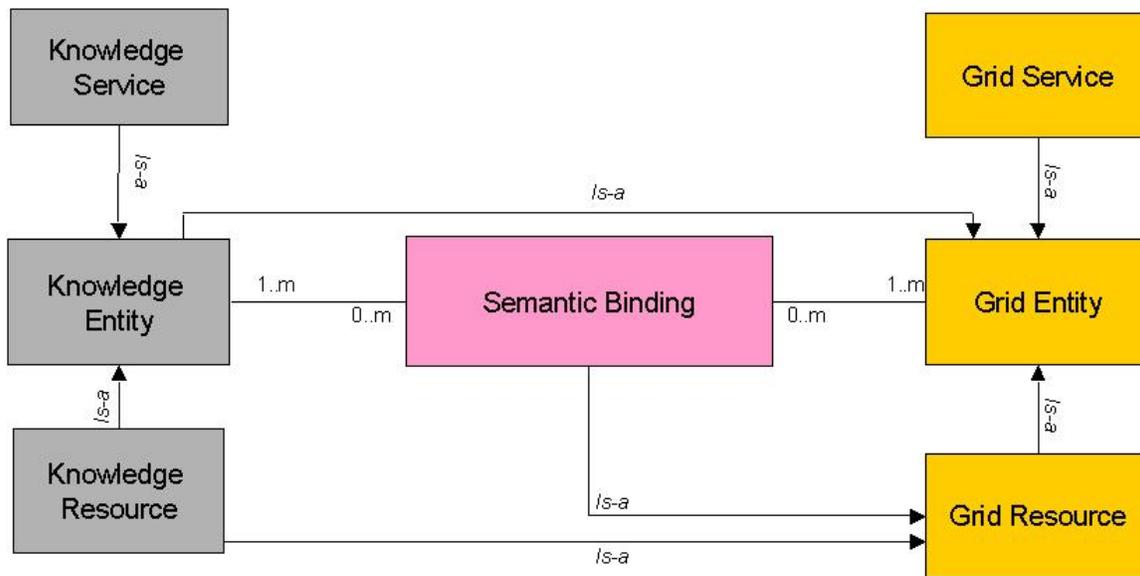


Figure 2. Entities in S-OGSA and their relationships

In the next section, Figure 3 gives a broader view of the S-OGSA model, including the capabilities and the entities.

3.2 S-OGSA Capabilities

According to our design principle of *diversity*, S-OGSA is a mixed economy of services with varying degrees of semantic capabilities. To achieve this goal, we extend the set of capabilities that Grid middleware should provide to include *Semantic Provisioning Services*⁸ and *Semantically Aware Grid Services* (SAGS). This extension is shown in Figure 1 with pink boxes (for semantic provisioning services) and dotted pink squares in the OGSA capability services (for Semantically Aware Grid Services). Semantic Provisioning Services are those responsible for the *provisioning* and management of explicit semantics and its association with Grid entities. SAGS are those enhanced Grid services that deliver OGSA enumerated capabilities but differ from others by having an affiliation with, or operating using, explicit semantics. Next we describe both types of services in more detail.

⁸ We use the term “provisioning” explicitly to make a clear distinction between those services that give support to the knowledge delivery and storage from those that are implemented using a semantic approach. These services could also have been called semantic services.

3.2.1 Semantic Provisioning Services

Semantic Provisioning Services are the services that give support to the provision of semantics, by allowing the creation, storage, update, removal and access of different forms of knowledge and metadata (i.e. Knowledge Entities and Semantic Bindings of the S-OGSA model). The semantics provisioned by these new categories of services apply to knowledge and metadata both in the Grid (i.e. related to the operation Grid middleware) and on the Grid (i.e. related to the Application domain).

Semantic provisioning services are further classified into two major categories (see Figure 3), namely Knowledge Provisioning Services and Semantic Binding Provisioning Services, reflecting the S-OGSA model.

Knowledge provisioning services include *ontology services*, which are in charge of the storage and access to the conceptual models of representing knowledge, and *reasoning services*, in charge of computational reasoning with those conceptual models.

An ontology service provides access to the knowledge stored in ontologies, in the form of concepts, relationships between concepts (including taxonomic relationships) and constraints on relationships. It thus deals with information related to the ontology's conceptual model (ontology instances will be dealt with by the metadata service defined below). Ontologies are accessed via a generic query language or through the use of a predefined ontology API. The exposure of ontologies to the Grid as managed resources through OGSA-DAI services is currently being investigated in the OntoGrid project [23], and the creation of a GGF working group on this topic is currently under discussion. Descriptions and comparisons of existing systems that can be used as ontology services are given in [24] and proposals for the provision of ontology services in Grid applications are given in [23].

Reasoning services allow inference of new information and the checking of constraints taking into account the knowledge stored in ontologies, working in close collaboration with the ontology service; in fact some of the functions of the ontology service often involve taxonomic reasoning [23].

Semantic binding provisioning services include *metadata services*, in charge of the storage and access to semantic bindings, normally considered as sets of ontology instances, and *annotation services*, in charge of generating metadata from different types of information sources, like documents, databases, provenance information, credentials, etc.

A metadata service is responsible for storing and providing access to Semantic Bindings. As with the Ontology Services, access to Metadata can be through a generic query language based on the metadata representation formalism or could be through the use of a predefined metadata API. There is a tight relationship between the metadata and ontology services, as the data stored by the metadata service will normally be based on the conceptual models that are stored in the ontology service. Metadata Services may also use the ontology and reasoning services to reason with the metadata they store.

Like all "Semantically Enhanced environments" the Semantic Grid comes with certain costs associated to its benefits. The cost is mainly related to development of knowledge (conceptual) models and metadata that adheres to these models. While generation of conceptualizations could be seen as a one-time job, metadata generation is a process that needs to be sustained during the operation of the Grid. Annotation services aim to act as the infrastructure that enables this through the (semi-)automation of the task of rich metadata generation. The annotation service aims to *create Semantic Bindings* (i.e. metadata) from existing information sources.

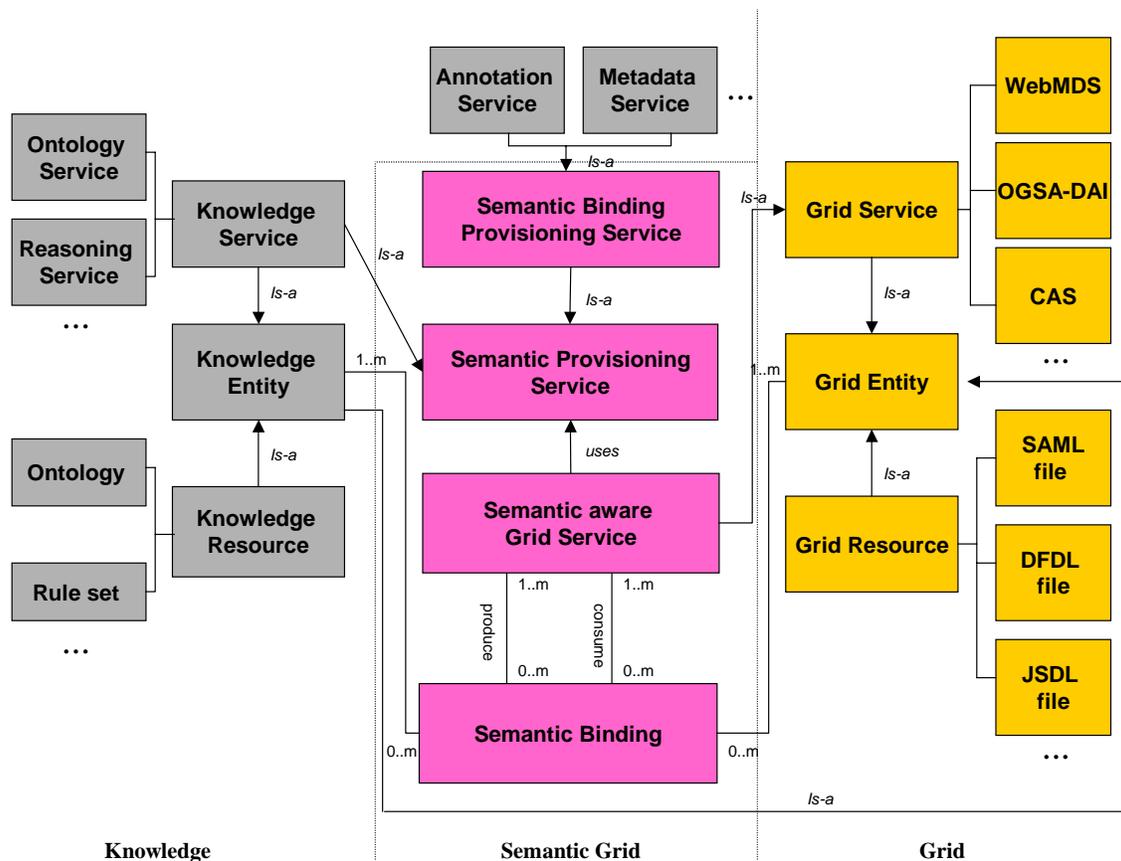


Figure 3. The Semantic Grid Information Model (an extended view). Components in yellow come from traditional Grid; components in grey can be found in Semantic Web platforms, and finally components in pink belong to the Semantic Grid

3.2.2 Semantically Aware Grid Services

Certain classes of middleware services in the Grid could exploit knowledge technologies to deliver their functionality. In Figure 3 we have identified these enhanced Grid services as Semantically Aware Grid Services. Semantic awareness here means being able to consume Semantics Bindings and being able to take actions based on knowledge and metadata. Examples of such actions are

- Metadata aware authorization of a *given identity* by a **VO Manager service**;
- Execution of a *search request* over entries in a **semantic resource catalogue**;
- Incorporation of a *new concept* in to an ontology hosted by an **ontology service**;
- Reduction of an annotated *scientific data set* to a smaller subset by a **scientist**.

SAGS allow for sharing of community-wide knowledge and may outsource knowledge-related activities. The explicit expression of knowledge in formalisms with well-defined interpretation mechanisms allows for representation of a common understanding of the environment among components both in and on the Grid. Sharing this knowledge brings flexibility to components and increases interoperability. Furthermore, the reasoning tasks can be outsourced to other specialised components (e.g. inference engines or rule engines).

3.3 Example: the S-OGSA Model and Capabilities in the context of a semantic-enabled access control system

In this section we will show how the notions described above apply to the creation of a semantically enabled access control system in a VO. This same example will be used in the next section to provide more insight about S-OGSA mechanisms.

VOs supply a context for operation of the Grid that can be used to associate users, their requests, and a set of resources. Resource providers and consumers define clearly and carefully what is shared, who is allowed to share, and the conditions under which sharing occurs [22]. That is, each organization that participates in the VO (including institutions, individuals, groups, etc.) defines a set of policies, which essentially consist of access rules for resources they own. The policy files are defined independently and are distributed in several locations within the VO, as shown in Figure 4. When a user wants to access a service in the VO, it will contact the service that controls the access to resources (the Policy Enforcement Point in the figure). This service reconciles these diverse policies so that the Policy Decision Point (authorization service / PDP in the figure) is able to reason with them and decide whether or not the user can have access to the resource.

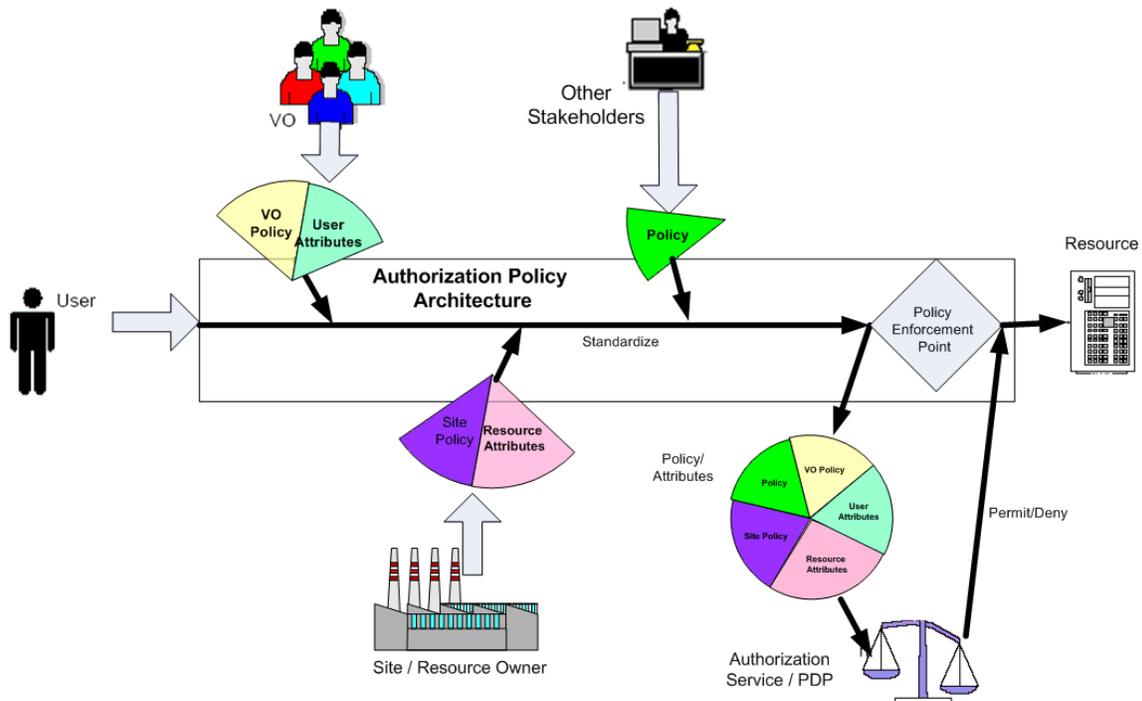


Figure 4. Authorisation policy architecture (adapted from [11]).

Figure 5 gives an example of the different types of S-OGSA entities that were described in section 3.1, in the context of this access control scenario.

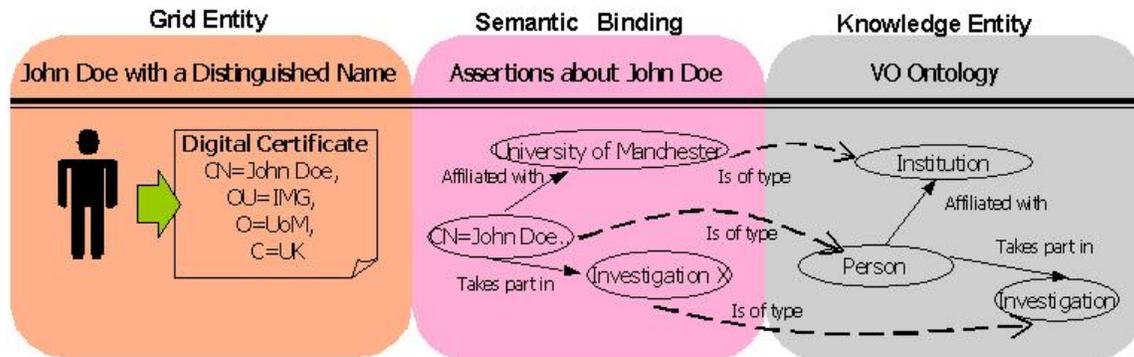


Figure 5. Examples of S-OGSA Entities

In OGSA (and consequently in S-OGSA) any entity with identity is defined as a Grid entity. Based on this, users/subjects within a VO are also **Grid Entities** generally identified by their Distinguished Names –DN- in certificates issued to them (see the DN CN=John Doe, OU=IMG, O=UoM, C=UK for John Doe within the digital certificate in the figure).

Early Semantic Grid approaches to modelling VOs and their sharing rules have been through the use of various SW technologies, viz. ontologies and rules [5, 6]. These VO Ontologies are examples of the **Knowledge Entity** concept of S-OGSA. VO ontologies model generic aspects, which could be used to characterize nearly every VO (e.g. Institutions, Persons, Resources) and problem/application specific aspects such as domain specific resource types (e.g., scientific data sets). A small extract of a generic VO ontology is given in the figure. Furthermore, VO Ontologies are functional not only in representing the entities in the environment but also the *VO formation and operation* policies.

- Policies for VO establishment are used to designate who can be a member under what conditions (that is, to specify role membership). An example could be as follows: *VO member is a user that is affiliated with an organization that is a member of the VO.*
- Resource Sharing policies are expressed through the concepts of Roles, Actions and Resources and the simple authorization pattern: *Role is authorized to perform Action on Resource.* We should note that there might be different technology specific methods (such as rules, axioms, defined classes, etc.) for modelling these policies, which are later exploited for making access control decisions at the time of resource utilization. An example of a resource sharing policy could be *Role X can perform a read operation on a resource (e.g. a job submitted to a Job Execution Manager) if (a) the VO member in that role is the job owner or (b) the member is the job owner's manager.*

The Knowledge entities in the Semantic Grid provide the essential conceptualizations, which can be used to structure metadata assertions about Grid entities. Within S-OGSA this structured metadata is represented by the **Semantic Binding** entity. Figure 5 depicts an example of a Semantic Binding as a group of assertions about the Grid Entity John Doe. In this example metadata assertions are structured with respect to the schema in the VO Ontology, though they could be also related to a set of rules or even textual descriptions. The semantic bindings could come into existence and evolve both during the formation and operation of the VO. For example the Semantic Binding on John Doe's institutional affiliation could be generated at formation time, whereas the Semantic Binding expressing John Doe being the owner of a submitted job could be generated when the Grid entity representing the job comes into existence.

The role of S-OGSA Capabilities in the context of VO management will be analysed in the example provided in the next section.

4 S-OGSA Mechanisms: Grounding S-OGSA to concrete Grid Platforms

Section 3 has presented the first two strands of our Semantic Grid Reference Architecture, namely the model and capabilities. The third strand in S-OGSA is a set of mechanisms through which we ground our

conceptual definitions regarding the use of metadata in the Grid into concrete Grid modelling elements. Our S-OGSA grounding is defined at a level that is independent from any Grid implementation system, which is a necessity for a *conceptual* and *parsimonious* Semantic OGSA specification.

Section 4.1 describes the set of mechanisms devised for treating Semantic Bindings as Grid resources in S-OGSA, following the design principle of *uniformity*. Section 4.2 focuses on how Semantic Bindings can be delivered by Grid services, in what we call S-Stateful Services.

4.1 Treating Knowledge Entities and Semantic Bindings as Grid Resources

Knowledge Entities and Semantic Bindings are treated as first class citizens in S-OGSA, as described in section 3. Consequently, they have to be reflected in the technology and paradigm specific layers. This is because we want to explicate the existence of semantics at appropriate abstraction levels: viz. the abstraction levels at which Grid implementation systems operate. This could also be seen as a reflection of the *uniformity* design principle of S-OGSA, where entities in our architecture are treated in the same way Grid entities are.

We have chosen the Common Information Model (CIM) Resource Model [18] to implement these S-OGSA extensions regarding Knowledge Entities and Semantic Bindings. CIM is an object-oriented, technology-independent and comprehensive schema for defining real world managed objects that occur in computer and network environments. CIM has been conceived by an industrial collaboration called the Distributed Management Task Force (DMTF)⁹, and has been endorsed by the GGF to be used as a base for developing the Grid Information Model, which would underpin interoperability among different Grid systems. The managed objects within CIM are not only defined by their attributes but also by the specific management operations that they support.

The CIM Model is currently represented in UML [25], although other representations are also being developed. Here we demonstrate our proposition to implement basic S-OGSA entities as an extension of the CIM Resource Model in UML. We have chosen the candidate standard CIM in order to have minimal impact and a smooth migration from Grid to Semantic Grid (according to our design principle of *parsimony*). Besides, our operation at the UML level enables us to remain at a conceptual level. Figure 6 depicts our extensions, which can be outlined as follows:

- Grid Entities are represented with the class `CIM-ManagedElement` in the CIM Model.
- Knowledge Entities are represented with the new class `S-OGSA-KnowledgeEntity`, which is an indirect subclass of `CIM-ManagedElement` (that is, Knowledge Entities are Grid Entities).
- Finally, the association between a Grid Entity (`CIM-ManagedElement`) and a Knowledge Entity (`S-OGSA-KnowledgeEntity`), which in our model is a Semantic Binding, is represented with the new class `S-OGSA-SemanticBinding`.

⁹ <http://www.dmtf.org/>

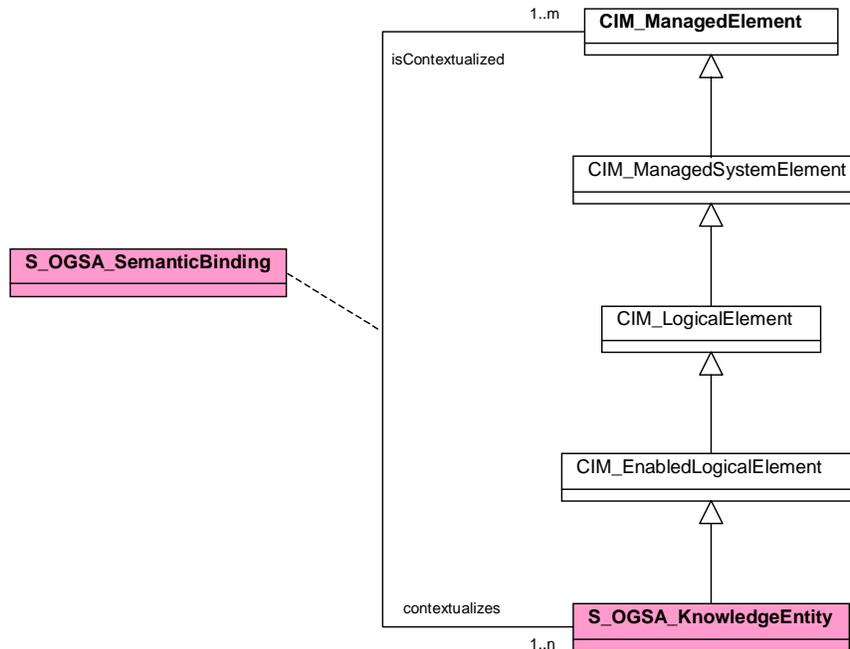


Figure 6. S-OGSA Model Space as an extension of the CIM model.

We use a mapping process to reflect the UML-based definitions of S-OGSA entities on to infrastructure-specific resource representations, following an idea that is similar to the one presented in [26]. CIM entities, including S-OGSA Knowledge Entities and Semantic Bindings, are transformed into OWL classes in a Resource Ontology, and the actual Knowledge Entities and Semantic Bindings are represented as instances of those classes. We anticipate that Grid middleware providers will define similar mappings in the near future in order to improve interoperability of their systems with others.

4.2 S-Stateful Services: Delivery of Semantic Bindings by Grid Services

We define S-Stateful Services to be those which virtualise Grid resources that are coupled with explicit metadata (that is, which have Semantic Bindings). Our S-Stateful Service specification is a set of mappings, with which our S-OGSA model can be grounded to the specific implementations of service oriented Grids. The S-Stateful Services is a delivery mechanism rather than a descriptive framework - It is not our intention to provide any exhaustive listing of what the content of semantic metadata about the Grid Resources would be, but rather we prescribe **mechanisms** for the delivery of Semantic Bindings for resources. Consequently the activities that could be undertaken based on the content of Semantic Bindings, such as discovery and composition, is out of the scope of our work in S-OGSA.

Our design decisions in S-Stateful Services have taken into account the list of design principles that were described in section 2, specifically those of *parsimony* and *extensibility* (the proposed framework must be as lightweight as necessary, and extensible and customisable versus complete and generic).

We have chosen the Web Services Resource Framework (WSRF) specification to devise a sample grounding of S-Stateful Services. WSRF [14] is a suite of specifications that define web service interfaces (with their associated messages) that could be used to create/destroy virtual representatives of Grid Resources, manage their lifecycle and inspect, aggregate and disseminate their properties. As described in [27], WSRF can be seen as an instruction set for the Grid, similar to HTTP PUT, GET, POST and DELETE operations being the instruction set of the Web.

WS-Resources have WS-ResourceProperties associated, which are the pieces of information defined as part of the state model of the resource, and which may reflect a part of the resource's state, metadata, manageability information, etc. Hence in our grounding we devise the existence of WS-ResourceProperties to provide information about the semantically-encoded metadata about a resource, so that it can be

retrieved and queried. Figure 7 depicts one possible pattern of interaction regarding the delivery of this semantically-encoded metadata, which can be summarized as follows:

- Metadata (Semantic Bindings) is treated as a resource on its own, managed via the metadata service, as shown in the figure. Services provide access to resources with properties typed with the corresponding XML Schema Datatypes generated by the mapping process described in the previous section. Knowledge Resources are treated similarly.
- Upon request, Grid resources can provide end point references of their associated Semantic Binding Resources (that they know about), through infrastructure specific metadata delivery operations (see steps 1 and 2 in Figure 7). For example, in the case of WSRF-based implementations, these operations are `getProperties` and `queryProperties`.
- Clients interested in exploiting Semantic Bindings can interact with the metadata service through any operation that a Semantic Binding resource might support (e.g., query, retrieve value, etc.), as shown in steps 3 and 4 of the figure. The query evaluation process may involve interaction with Knowledge Services (e.g. ontology servers, reasoners, etc.), as shown in step 5.

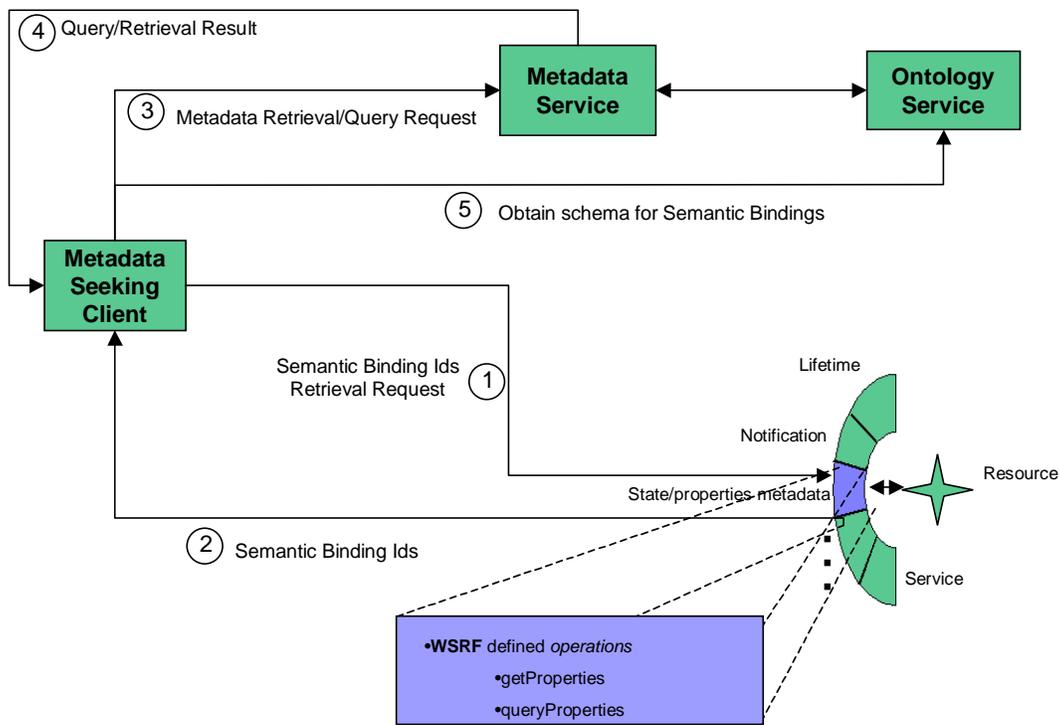


Figure 7. Retrieving and Querying Semantic Bindings of Resources

The provider of Semantic Bindings related to a Grid Entity may not necessarily be the provider of the entity itself. Third parties may also generate Semantic Bindings for Grid Entities and publish them via metadata services.

4.3 Example. S-OGSA Grounding in the Context of a semantic-enabled access control system

To illustrate S-OGSA Mechanisms in the context of our example, we depict in Figure 8 a Semantically Aware Authorization Service and its interaction with the S-OGSA Semantic Provisioning Services¹⁰.

¹⁰ The scenario depicted in this example is based on the pull-based Authorization scenario of GGF's conceptual access control framework.

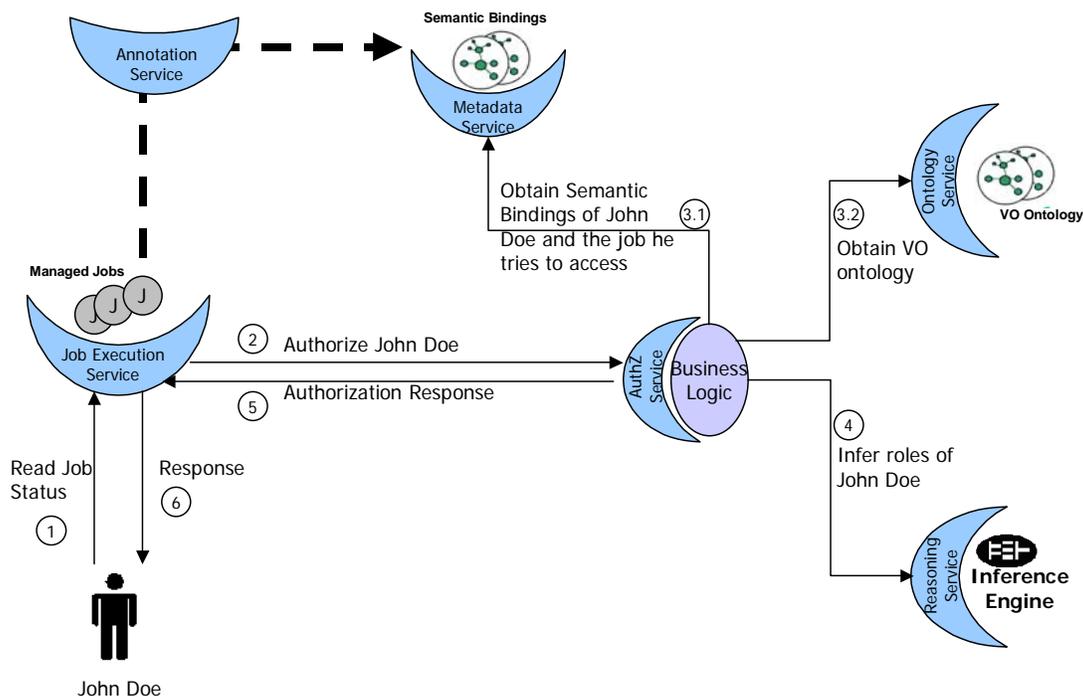


Figure 8. A Semantic Aware Authorization Service consuming an input that could in relation with S-OGSA Semantic Provisioning Services

According to the scenario depicted in Figure 8,

1. The subject John Doe contacts a job execution service (obviously through appropriate tooling such as a Grid portal or similar which is not shown in the figure for brevity) to obtain the results of a job that was submitted by one of his interns.
2. The job execution service contacts the authorization service to check whether John Doe is authorized to perform the required action according to the sharing rules of the VO. The semantically aware authorization service, requires the ID of the Actor to be authorized (e.g. the Distinguished Name –DN- of John Doe), the ID of the requested resource (e.g. the ID of the job that one of John Doe’s interns has submitted) and the requested action type (e.g. obtain job status info).
3. Upon receipt of these, the authorization service contacts the Semantic Provisioning Services to retrieve the Semantic Bindings and Knowledge Entities that will be used for the decision making process:
 - 3.1 It contacts the metadata service to obtain the Semantic Bindings associated with the Subject and Object Grid Entities in the authorization question posed to it.
 - 3.2 Optionally, it contacts the ontology service to obtain Knowledge Entities that it needs to know about in order to interpret the Semantic Bindings of the subject and/or the object.
4. The authorization service then contacts the reasoning service for the deduction of the roles that John Doe plays at the time of authorization request using the role definitions in the VO ontology and the metadata assertions about John Doe
5. Based on John Doe’s roles obtained from the Reasoning Service, the sharing rules in the VO Ontology and the Semantic Bindings of the job that he is trying to access, the authorization service evaluates an access control function, which leads to an authorization response to be sent to the requestor (the job execution service).

6. Based on the result of authorization the Job submission service either undertakes the desired function or returns an authorization failure message to the user.

In this interaction pattern, we assume the existence of Semantic Bindings of Grid entities (i.e. John Doe and the job he is trying to inspect). With S-OGSA Capabilities we have appointed the Annotation Service category in S-OGSA for creating Semantic Bindings for resources as they come into existence. In the context of our example this necessary link between Grid entities and their Semantic Bindings can be modelled with an additional set of rules, triggered by Grid events. For instance, we may postulate that an individual, who successfully submits a job to a resource, becomes the owner of that resource. This rule is triggered by a new job created event, generated by the Job Execution Service.

5 Related work

S-OGSA is not the only attempt to provide an architecture for the development of Semantic Grid applications or simply Semantic Aware Grid Services. In [28] we can find a high-level architecture for knowledge-oriented Grids, as well as descriptions of some case studies where Semantic Grid ideas can be applied. Besides, projects like InteliGrid¹¹ and ^{my}Grid¹² have made significant attempts to provide either such architectural principles or to show how explicit metadata can be used in the context of existing Grid applications, respectively.

[28] already discusses the fact that semantics in Grid applications cannot be placed in a separated layer, different from other resources like data, computational resources, etc. On the contrary, semantics permeate the full virtual vertical extent of Grid applications and infrastructure. This proposal distinguishes several macrocomponents that work together: knowledge networks (our Knowledge Entities); knowledge-generating services (our Semantic Binding Provisioning Services); knowledge-aware, knowledge-based or knowledge-assisted Grid services (our Semantic Aware Grid Services); and Grid knowledge services (our Ontology and Reasoning Services). However, the proposal does not go into more detail about the actual mechanisms to be used to deliver and consume semantics.

The ^{my}Grid [3] project is a pioneering Semantic Grid effort, which has developed a suite of tools and services to enable workflow based composition of diverse biological data and computational resources. Within the project Semantic Web technologies have been applied to the problems of resource discovery and workflow results management. A characterizing aspect of these solutions is their focus on user-orientation, which has resulted in 1) semantic aware decision-support components instead of decision making components and 2) user-facing tools that keep the human in the loop during metadata generation and querying. From an S-OGSA perspective, the ^{my}Grid approach to exploiting semantics in a service-oriented resource-sharing environment is rather principled. The project has identified the need for specialized components for storage of ontologies and metadata (Semantic Provisioning Services), and for service discovery (an example of a Semantically Aware Grid Service).

InteliGrid [29] proposes an architecture based on three layers: conceptual, software and basic resource. At the conceptual layer we find descriptions of knowledge entities such ontologies, notions, graphs, etc. while the software layer consists of software that consumes the knowledge entities found in the conceptual layer (this is equivalent to the Semantic Aware Grid Services from our architecture). The basic resource layer includes the low level infrastructure and resembles the notion of the Grid fabric. Ontology services, situated in the software layer, play a central role in this architecture as they are considered as interoperability services that support multiple functionalities such as data consistency, service discovery, VO set-up management, etc., while in S-OGSA they are meant to allow storage and access (and reasoning, by means of the reasoning services) of knowledge entities, leaving such semantic-aware tasks to the developer of a Semantic Grid application.

¹¹ <http://www.inteligrid.com/>

¹² <http://www.mygrid.org.uk/>

If we analyse the characteristics of both proposals with respect to the list of design principles that we outlined in section 2¹³, we obtain the following interesting results:

- The architecture described in [28] conforms with the design principles of *parsimony*, *extensibility* and *diversity*. However, it does not provide enough detail to be able to determine what is the approach with respect to the other principles (*uniformity*, *heterogeneity* and *enlightenment*).
- The ^{my}Grid approach has strong similarities with S-OGSA with respect to the design principles of *parsimony*, *extensibility* and *uniformity*. Both approaches share the vision that there should be a simple migration path from Grid applications to Semantic Grid applications, so that the uptake of semantics in application development can be easily done (hence the need for a lightweight and extensible approach). ^{my}Grid also investigated how semantics can be used to tackle knowledge-intensive activities, such as service discovery, while this is not considered by S-OGSA, which is only focused on providing the mechanisms that allow the performance of such knowledge-intensive tasks. ^{my}Grid also considers the coexistence of Grid and Semantic Grid services, though they can just be semantics ignorant or semantics aware and capable of processing. However, ^{my}Grid does not consider the rest of our design principles, related to the fact that knowledge may be represented in multiple forms, that there could be several interpretations of a component, and that the semantics associated to a Grid entity can be managed and can have a lifecycle.
- InteliGrid also complies with the criteria of *extensibility*, *uniformity* and, to some extent, *diversity* (as ^{my}Grid does). However, the heavyweight use of ontology services as interoperability services that are used throughout all the applications developed with this architecture goes against the *parsimony* principle and may have a negative impact in the uptake of this architectural proposal. Finally, InteliGrid does not address the possibility of having *multiple* forms of knowledge or multiple interpretations, and *enlightenment*.

In both cases, the proposals are grounded in a specific platform (Web services for ^{my}Grid and Globus Toolkit for InteliGrid), although the same ideas could be easily applied to other platforms.

There are also other ongoing projects where the use of explicit metadata is identified as a key issue in the next generation of Grid application development. However, they do not provide specific proposals or mechanisms on how to expose and deliver this metadata in application development, or they focus on very specific advanced knowledge-intensive functions like service discovery. Furthermore, none of those projects addresses the issue of manageability, soft-state aspects related to metadata. Examples of such projects are NextGrid¹⁴, AkoGrimo¹⁵, KWfGrid¹⁶ [30], or KnowledgeGrid [31]. We believe that all of these projects could benefit from the application of the architecture described in this paper, and we are aiming at providing support to them as part of EU Grid concertation activities.

From the Semantic Web Service (SWS) perspective, we can also perceive some similarities and differences with respect to the work that has been presented in this paper, some of which were already pointed out in the introduction. In general, we can argue that SWS approaches like WSMO¹⁷ or OWL-S¹⁸ are mainly focused on solving complex problems like discovery, composition, and negotiation. If we analyse SWS approaches with respect to our design principles, we can conclude that these approaches do not focus on parsimony, extensibility and uniformity, but mainly on the aspects of diversity (Semantic Web Service execution environments can execute both Web services and Semantic Web services) and *heterogeneity* (the problems of language, content and process heterogeneity are overcome by mediation techniques). That is,

¹³ The use of our own list of design principles to analyse other approaches could be seen as an unfair set of criteria to perform the comparison. However, we are not aiming to benchmark the different approaches but rather to provide a deeper description of each of them according to a known set of criteria, so that we can outline the main similarities and differences between approaches.

¹⁴ <http://www.nextgrid.org/>

¹⁵ <http://www.akogrimo.org/>

¹⁶ <http://www.kwfgird.net/>

¹⁷ <http://www.wsmo.org/>

¹⁸ <http://www.daml.org/services/owl-s/>

they provide heavyweight complete solutions (e.g., the WSMX environment¹⁹, IRS-III²⁰ [32] or the OWL-S Virtual Machine [33]) that may make the uptake of their approach more difficult. Finally, even if the platforms follow the *enlightenment* principle, this is not usually addressed in the sample applications that are being developed with these approaches.

Another approach that is worth comparing S-OGSA to is WSDL-S²¹. This approach is also proposed for the explicitation of semantics of Web services, so that they can be more easily discovered and composed. One of the most relevant differences with respect to S-OGSA is that in WSDL-S the original WSDL files of Web services are modified so that they contain the semantics of the Web service, what means that semantic annotations are not manageable entities, since they cannot be separated from the WSDL description and cannot exist by themselves. WSDL-S only follows the principles of parsimony and diversity.

In this respect, the main conclusion is that the approaches presented in S-OGSA and in Semantic Web Services are complementary: S-OGSA can be used for the Grid-compliant exposure and delivery of semantics provisioned and used by Semantic Web Service approaches. At the same time, S-OGSA based applications could benefit from the solutions to knowledge-intensive problems that are provided by SWS approaches, namely discovery, composition and negotiation of service level agreements.

6 Conclusions and future work

The objective of our work in S-OGSA is the provision of a unified platform for exposing and delivering explicit metadata in Grid applications, including a formal framework and a set of guidelines to ease the development of Semantic Grid applications. To achieve this objective, we have identified and defined:

- Extensions to current Grid models to deal with flexible forms of explicit metadata. The central component in this extended model is the Semantic Binding, which relates Grid Entities and Knowledge Entities in multiple forms.
- A set of services (Semantic Provisioning Services) that play an important role in the exposure, delivery and generation of metadata. This set includes ontology management and reasoning services, metadata services and annotation services.
- The actual mechanisms to be used for treating the new components as Grid entities and for delivering them as part of existing Grid service frameworks.

In the development of our approach, we have followed carefully the set of design principles that we defined in section 2. We do not claim that the compliance with all of these criteria is compulsory in any Semantic Grid framework. However, we consider them all equally important in promoting the uptake of any approach that attempts to provide guidelines for the development of Semantic Grid applications, while at the same time staying compliant with most of the design principles of Grid applications. These principles can be summarised as follows:

- **The Semantic Grid is the Grid.** S-OGSA provides an extension of the Open Grid Services Architecture (OGSA). All the newly identified entities and resources are Grid resources, and S-OGSA is an extension of Web Services as well.
- **The Semantic Grid has a spectrum of Semantic Capabilities.** S-OGSA has been designed to allow the coexistence of semantic and non-semantic aware services. In S-OGSA Grid services can be classified as ignorant to semantics, aware of semantics but not able to process it and aware of semantics and able to process it.
- **Painless migration to the Semantic Grid.** In order to ease the uptake of S-OGSA, semantic and non-semantic aware services can coexist, as aforementioned, and different degrees of semantic awareness exist and are all covered under the S-OGSA umbrella.
- **Semantic Grid lifecycle.** Given the S-OGSA model, Semantic Bindings can be created, attached and detached to/from entities at any time during their lifetime. The semantic provisioning services envisaged in S-OGSA allow dealing with different aspects of their lifecycle.

¹⁹ <http://www.wsmx.org/>

²⁰ <http://kmi.open.ac.uk/projects/irs/>

²¹ <http://www.w3.org/2005/04/FSWS/Submissions/17/WSDL-S.htm>

- **Multiple semantics.** The S-OGSA model allows the existence of different kinds of knowledge attached to any resource.

There are still many challenges to explore in the quest for developing Semantic Grid applications. Many are technical—architectural or theoretical foundations, the maturity of Semantic and Grid technologies, their appropriateness for the required tasks, their scalability, the separation of Grid level and application specific semantics, making it easier to combine semantic infrastructure with Grid computing infrastructure, and minimizing the additional overhead and improving the performance of creating and retrieving semantically-encoded metadata. Others are operational—gathering and maintaining the semantic content, convincingly showing the added value of semantics when the return on investment may come downstream, be long term and benefit developers other than the originators. Some are sociological and political—the interplay between the Semantic and the Grid communities, and the legal, security and privacy implications of clearly exposed metadata and automated reasoning.

Some of these challenges will be addressed in the future, in the context of the OntoGrid project, where S-OGSA has been developed. We will perform a further and thorough evaluation of the architecture, which will be done in the context of middleware (VO management) and business (insurance settlement and quality analysis in satellite missions) use cases that have been defined for the OntoGrid project. Furthermore, we are exploring the possibility of applying S-OGSA to the case studies of other Grid projects, as pointed out in the previous section.

Finally, we believe that S-OGSA will be useful in providing a perspective to both Semantic Web and Grid communities in drawing commonalities among existing ad-hoc approaches and providing guidelines for a principled approach.

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 David Snelling, David De Roure, James Myers and the anonymous reviewers for their useful feedback, which have improved substantially the clarity of the final version of this paper.

References

- [1] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*: Morgan Kaufmann, 1999.
- [2] D. De Roure, Y. Gil, and J. A. Hendler, "Guest editors' introduction: E-Science," *IEEE Intelligent Systems*, vol. 19, pp. 24 - 25 2004.
- [3] C. Wroe, C. A. Goble, M. Greenwood, P. Lord, S. Miles, J. Papay, T. Payne, and L. Moreau, "Automating Experiments Using Semantic Data on a Bioinformatics Grid," *IEEE Intelligent Systems*, vol. 19, pp. 48-55, 2004.
- [4] S. Cox, L. Chen, S. Campobasso, M. Duta, M. Eres, M. Giles, C. Goble, Z. Jiao, A. Keane, G. Pound, A. Roberts, N. Shadbolt, F. Tao, J. Wason, and F. Xu, "Grid Enabled Optimisation and Design Search (GEODISE)." UK e-Science All Hands Meeting 2002 Sheffield, UK, 2002.
- [5] L. Pouchard, S. Bechhofer, B. Matthews, J. Myers, D. Snelling, and Y. Sure, "A Virtual Organisation Ontology", 2006, <http://www.csm.ornl.gov/~7lp/ontology/DagstuhlVirtualOrganization1.0.owl>, (2006).
- [6] J. Bradshaw, A. Uszok, R. Jeffers, N. Suri, P. Hayes, M. Burstein, A. Acquisti, B. Benyo, M. Breedy, M. Carvalho, D. Diller, M. Johnson, S. Kulkarni, J. Lott, M. Sierhuis, and R. V. Hoof, "Representation and reasoning for DAML-based policy and domain services in KAoS and nomads," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*. Melbourne, Australia: ACM Press, 2003
- [7] D. De Roure, "Future for European Grids: GRIDs and Service Oriented Knowledge Utilities. Vision and Research Directions 2010 and Beyond", 2006, <http://www.semanticgrid.org/documents/ngg3/ngg3.html>, (2006).

- [8] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, and J. V. Reich, "The Open Grid Services Architecture, Version 1.0." Open Grid Services Architecture WG, Global Grid Forum, 2005
- [9] Z. Németh and V. Sunderam, "Characterizing Grids: Attributes, Definitions, and Formalisms," *Journal of Grid Computing*, vol. 1, pp. 9-23, 2003.
- [10] I. Foster, D. Gannon, H. Kishimoto, and J. V. R. Jeffrin, "Open Grid Services Architecture Use Cases." Open Grid Services Architecture Working Group, Global Grid Forum, 2004
- [11] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration." Open Grid Service Infrastructure WG, Global Grid Forum, , 2002
- [12] M. Atkinson, D. DeRoure, A. Dunlop, G. Fox, P. Henderson, T. Hey, N. Paton, S. Newhouse, S. Parastatidis, A. Trefethen, P. Watson, and J. Webber, "Web Service Grids: An Evolutionary Approach," *Journal of Concurrency and Computation: Practice and Experience*, vol. 17, pp. 377-389, 2005.
- [13] "Semantic Grid: The Convergence of Technologies." Dagstuhl Seminar No05271, organised by C. Goble, C. Kesselman, Y. Sure 2005 <http://drops.dagstuhl.de/portals/index.php?semnr=05271>.
- [14] 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, 2005
- [15] W. Vambenepe, "Web Services Distributed Management: Management Using Web Services (MUWS 1.0)," OASIS, 2005
- [16] L. Pouchard, L. Cinquini, and G. Strand, "The Earth System Grid Discovery and Semantic Web Technologies", Workshop for Semantic Web Technologies for Searching and Retrieving Scientific Data 2nd International Semantic Web Conference 2003.
- [17] N. Sharman, N. Alpdemir, J. Ferris, M. Greenwood, P. Li, and C. Wroe, "The myGrid Information Model", UK e-Science programme All Hands Conference 2004.
- [18] "Common Information Model (CIM), CIM Schema Version 2.11." 2005, http://www.dmtf.org/standards/cim/cim_schema_v211/, (2005).
- [19] "Global Grid Forum (GGF) Common Information Model (CIM) Working Group", 2005, <https://forge.gridforum.org/projects/cgs-wg/>, (2005).
- [20] "GGF Data Format Definition Language Working Group", 2005, <http://forge.gridforum.org/projects/dfdl-wg>, (2005).
- [21] "GGF Job Submission Description Language Working Group (JSDL)", 2005, <https://forge.gridforum.org/projects/jsdl-wg/>, (2005).
- [22] J. Treadwell, "Open Grid Services Architecture Glossary of Terms." Open Grid Services Architecture WG, Global Grid Forum, , 2005 An updated version can be found at the documents section at <http://forge.gridforum.org/projects/ogsa-wg>.
- [23] M. E. Gutiérrez, S. Bechhofer, O. Corcho, M. Fernández-López, A. Gómez-Pérez, Z. Kaoudi, I. Kotsiopoulos, M. Koubarakis, M. Suárez-Figueroa, and V. Tamma, "Specification and Design of Ontology Grid Compliant and Grid Aware Services ", 2005
- [24] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, *Ontological Engineering*: Springer-Verlag, 2003.
- [25] "Unified Modeling Language (UML), version 2.0", <http://www.omg.org/technology/documents/formal/uml.htm>, (2006).
- [26] A. Gómez-Pérez, "A Survey on Ontology Tools," OntoWeb Deliverable D1.3, 2002.
- [27] T. Priol, "Objects, Components, Services for Grid middleware: pros and cons", Keynote talk at European Grid Conference, Science Park, Amsterdam, The Netherlands,, 2005.
- [28] C. A. Goble, D. D. Roure, N. R. Shadbolt, and A. A. A. Fernandes, " Enhancing Services and Applications with Knowledge and Semantics," in *The GRID 2. Blueprint for a New Computing Infrastructure*, I. Foster and C. Kesselman, Eds. San Francisco: Morgan Kaufmann Publishers, 2004, pp. 431-458.
- [29] Ž. Turk, V. Stankovski, A. Gehre, P. Katranuschkov, K. Kurowski, E. Balaton, J. Hyvarinen, M. Dolenc, R. Klinc, J. Kostanjšek, and J. Velkavrh, "Semantic Grid Architecture," 2004
- [30] "The Knowledge-based Workflow System for Grid Applications (K-Wf Grid)", 2005, <http://www.kwfguid.net/>, (2005).

- [31] H. Zhuge, "China's e-science knowledge grid environment," *IEEE Intelligent Systems* vol. 19, pp. 13, 2004.
- [32] J. Domingue, L. Cabral, F. Hakimpour, D. Sell, and E. Motta, " IRS-III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services", Workshop on WSMO Implementations (WIW 2004) CEUR Workshop Proceedings, ISSN 1613-0073, Frankfurt, Germany, 2004.
- [33] M. Paolucci, A. Ankolekar, N. Srinivasan, and K. Sycara, *The DAML-S Virtual Machine*. Lecture Notes in Computer Science: Springer-Verlag, 2003.