GWD-I
Category: Informational
Open Grid Service
Common Management Model (CMM) WG

Frederico Buchholz Maciel, Hitachi, Ltd.
Editor

May 12, 2004

Resource Management in OGSA

Status of This Memo

This memo provides information to the Grid community on resource management in OGSA. It does not define any standards or technical recommendations. Distribution is unlimited.

This document is being constantly updated. The latest version can be found at:

http://forge.gridforum.org/projects/cmm-wg/document/CMM_Gap_Analysis/en

Copyright Notice

Abstract

TBD: Document Abstract.

Contents

1. Introduction ... 3
1.1 Related Work ... 3
2. Definitions ... 4
3. Management in OGSA ... 5
3.1 Requirements ... 5
3.2 Levels ... 7
4. Resource Models ... 10
5. Analysis of the OGSA Capabilities ... 12
5.1 Base Manageability ... 13
5.2 Generic Manageability Interface ... 14
5.3 Specific Manageability Interfaces ... 14
5.4 Analysis of selected services ... 19
6. Conclusion ... 19
6.1 Summary of Gaps ... 19
6.2 Future Work ... 19
7. TBD ... 20

8. Security Considerations 20
Author Information 20
Glossary 20
Intellectual Property Statement 20
Full Copyright Notice 21
References 21

1. Introduction

Any computing environment requires some degree of system management: monitoring and maintaining the health of the systems, keeping software up-to-date, maintaining user accounts, managing storage and networks, scheduling jobs, managing security, and so on. The complexity of the management task increases as the number and types of resources requiring management increases, and is further complicated when those resources are distributed.

The Grid computing model, with its use of resources that tend to be both heterogeneous and distributed across multiple management domains, faces all the traditional IT management issues, and also brings new challenges – not only in the management of its component resources, but also of the Grid itself. For example, in a Grid environment shared resources must remain accessible, key infrastructure services must be available, and virtual organizations must be maintained. It must also be possible to detect, report and deal with faults that may occur in any of the member domains. As Grid technology is increasingly adopted across institutions and enterprises, the distinctions between Grid environments and traditional IT environments will blur, and these challenges will become more widespread.

Effective system management is only possible if resources are manageable, and if tools are available to manage them. Today, system administrators can choose from a wide variety of management tools from system vendors, third party suppliers and the open source community. However, these tools tend to operate independently and to use proprietary interfaces and protocols to manage a limited set of resources, making it difficult for an organization to build an efficient, well-integrated management system. This issue is being addressed through the development of *manageability standards* that will enable conforming management tools to manage conforming resources in a uniform manner, and to interoperate with each other. In turn this will enable system administrators to choose their management tools and suppliers in the knowledge that, regardless of their origin, the tools can work cooperatively in an integrated management environment.

This document offers a detailed discussion of the issues of management in a Grid based on the Open Grid Services Architecture (OGSA) [1]. It first defines the terms and describes the requirements of management as they relate to a Grid, and then organizes the interfaces, services, activities, etc. that are involved in Grid management, including both management *within* the Grid and the management *of* the Grid infrastructure. It concludes with a comprehensive gap analysis of the state of manageability in OGSA, primarily identifying Grid-specific management functionality that is not provided for by emerging Web services-based distributed management standards. The gap analysis is intended to serve as a foundation for future work..

1.1 Related Work

The foundation for this work is the Open Grid Services Architecture document, which is being developed by the Global Grid Forum's (GGF's) OGSA Working Group (OGSA-WG).

The document is also intended to build upon the work being carried out in the OASIS Web Services Distributed Management (WSDM) Technical Committee (TC) [2,3]. The following text appears in the WSDM Statement of Purpose:

> To define web services management. This includes using web services architecture and technology to manage distributed resources. This TC will also develop the model of a web service as a manageable resource.

WSDM is developing separate documents to address management *of* Web services (MOWS) [5] and Management *using* Web services (MUWS) [6]. The interfaces defined in those documents are expected to become key standards for manageability across the IT landscape, and will form the basis for management of Grids.

As the documents being developed by these and other groups mature, the information in this document may need to be revised.

Other related work to document:

- Other gap analyses exist, such as the e-Science Gap Analysis [7, 8] and the GGF Data Area gap analysis currently in progress [9]. These analyses mention management on Grids, however they do not specifically analyze the manageability aspects of Grids, to the authors' knowledge.
- The Grid Monitoring Architecture (GMA) [5, 11] describes the major components of a Grid monitoring architecture and their essential interactions. The scope of this work overlaps with the one of the GMA since monitoring is a subset of management. However, both works don't conflict; this work contains many of the GMA elements, sometimes in a re-factored form or described with different terminology.
- [Add IBM's proposal if and when disclosed]

2. Definitions

Management (in Grids or otherwise) is the process of monitoring an entity, controlling it, maintaining it in its environment, and responding appropriately to any changes of internal or external conditions.

A **manager** initiates management actions; it might be either a management console operated by a human, or a software entity that is able to monitor and control its targets automatically.

Manageability defines information that is useful for managing a resource or service. Manageability encompasses those aspects of an entity that support management specifically through instrumentation that allows managers to interact with the entity. The manageability may be provided by the resource itself or by a separate means.

Manageability interfaces are sets of standardized interfaces that allow a manager to interact with an entity in order to perform common management actions on it. Typical management actions include starting the entity, stopping it, and gathering performance data.

Manageable entities are entities that provide manageability interfaces and thus, as the name implies, can be managed. Manageable entities can be:

- *physical* (e.g., a node, a network switch or a disk) or *logical* (e.g., a process, a file system, a print job, or a service)
- *discrete* (e.g., a single host) or *composite* (e.g., a cluster)
- *transient* (e.g., a print job) or *persistent* (e.g., a host)

A **resource model** is an abstract representation of manageable entities, which defines their schema (conceptual hierarchy and inter-relationships) and characteristics (attributes, management operations, etc.).

The term **manageable resources** (or simply **resources**) means the same as *manageable entities*. The term includes entities such as software licenses, bandwidth and routing tables that do not expose generally-useful manageability interfaces, but may still be managed by some other means.[1]

[TBD: write about the definition of resources being wide, and about resources having multiple facets. Clarify with Jay: is it things like virtual/physical, or multiple management disciplines

[1] In a Grid environment the term *resource* is often applied only to manageable entities that are *pooled* (e.g. hosts, software licenses, IP addresses, etc.) or that provide a given *capacity* (e.g. disks, networks, memory, etc.). For these classes of resource some part of the pool and/or the capacity may be allocated and used. By this definition processes, print jobs, registry services and VOs are *not* resources. Notice that this is a subset of the definition of resources as manageable entities.

(security aspect, deployment aspect, reservation aspect, monitoring aspect), or an application being a resource and a manager at the same time, something else, or all of these?]

Resource management is a generic term for several forms of management as they are applied to resources. These forms of management include (but are not limited to) typical distributed resource management (DRM) activities and IT systems management activities, such as:

- reservation, brokering and scheduling
- installation, deployment and provisioning
- accounting and metering [To the OGSA-WG: as pointed in the OGSA-WG teleconference, accounting is not an OGSA service, but this should not disqualify it. Opinions?]
- aggregation (service groups, WSDM collections, etc.)
- VO management
- security management
- monitoring (performance, availability, etc.)
- control (start, stop, etc.)
- problem determination and fault management

[The items in these categories will be refined as the OGSA specification matures. Sync with the OGSA glossary. BTW, this affects all the text in this document, plus the Figures.]

Resource management includes the various management tasks, but not the mechanisms they use, such as discovery.

Since resource management comprises many activities in many management disciplines, using the term to refer to a single activity may be ambiguous, and should be avoided.

A **resource manager** is a manager that implements one or more resource management functions.

3. Management in OGSA

3.1 Requirements

The basis for manageability in an OGSA Grid is the WSDM MUWS specification [6]. This means that for a resource to be manageable, it must provide the minimum set of manageability capabilities specified by MUWS. The current 0.5 version of MUWS specifies requirements for *identity*, *state* and *metrics*. In the forthcoming MUWS 1.0 release it is anticipated that *notification*, *discovery*, *configuration* and *collections* will be included. All of these topics are critical to management, and must be supported as appropriate within OGSA services.

The following list enumerates the main requirements for management in OGSA. These requirements are especially important in a large-scale, distributed environment with no centralized notion of control, such as a Grid:

- **Scalability**: Management architecture needs to scale to potentially thousands of resources. Management needs to be done in a hierarchical and/or peer-to-peer (federated/collaborative) fashion to achieve this scalability, so OGSA should allow these forms of management. Hierarchical management can be implemented through manageability interfaces that allow resources to be grouped and managed collectively (e.g. Grid Monitoring Architecture (GMA) aggregators and intermediaries that implement WSDM collection interfaces). Hierarchical management techniques include:
    - o Providing a proxy that allows a manager to perform the same action on multiple resources with a single request.

  - Computing metrics that aggregate resource data (e.g., average load, average reservation rate).
  - Filtering and aggregating events.
  - Polling resources for state (reserved, running, failed, idle, saturated, etc.) and providing the results on request, as well as sending events when the state changes (a.k.a. *pull* or *push* notification).

  Requirements related to peer-to-peer management are stated in a later item.

- **Interoperability**: Management architecture must be able to span software, hardware and service boundaries, e.g., across the boundaries between different products, so standardized and broad interoperability is essential to avoid "stovepipes." Two kinds of interoperability are needed:
  - between levels: e.g., between a resource and its manager;
  - at the same level: e.g., a scheduler accessing a broker.

  Interoperability in both cases requires that the interfaces are defined in a standard way. This applies both to Grid-specific standards and to general IT management standards.

- **Security**: There are two security aspects in management:
  - Management *of* security: the management of the security infrastructure, including the management of authentication, authorization, access control, VOs and access policies.
  - Secure management: using the security mechanisms on management tasks. Management should be able to ensure its own integrity and to follow access control policies of the owners of resources and VOs.

- **Reliability**: A management architecture should not force a single point of failure. Managers must be allowed to manage multiple manageable resources, and a manageable resource must be allowed to be managed by multiple managers.

  For purposes of reliability, a resource may be virtualized by multiple services exposing a single URL as the management endpoint. In such situations, the system that provides manageability capabilities must be aware that, for certain queries such as metrics, the manageability provider must aggregate the results from the multiple services that virtualize that single resource.

- **Policy**: Management must be able to enforce policy assertions that are put in place to support requirements and capabilities such as authentication scheme, transport protocol selection, QoS metrics, privacy policy, etc.

- **Performance Monitoring**: Performance monitoring facilities should satisfy the following requirements outlined in the Grid Monitoring Architecture [10, 11]:
  - Low latency to keep performance data relevant
  - Handle high data rates
  - Minimal measurement overhead

- **Peer-to-Peer Management Requirements**: Grid systems that comprise large peer-to-peer systems have the following requirements:
  - **Discovery**: While discovery mechanisms are used in traditional distributed systems, membership of peer-to-peer systems is typically highly dynamic, and hence they rely even more heavily on discovery mechanisms being both efficient and effective.
  - **Security**: Some specific requirements are around community-based trust mechanisms, replication, and verification of user identities. User privacy and anonymity are also characteristics of such systems.

  - **Location awareness**: This is the capability of an application to take advantage of proximity – relative, absolute or contextual. This is important in providing location-based services or system-level optimizations.
  - **Group support**: Peer-to-peer systems allow for the creation and management of dynamic groups with large transient populations. Management must be able to create and manage dynamic user groups.

3.2 Levels

In an OGSA Grid there are three types of management:

- Management of the resources themselves (e.g., rebooting a host, or setting VLANs on a network switch)
- Resource management on the Grid (i.e., the functions provided by the resource managers)
- Management of the OGSA infrastructure (e.g., monitoring a registry service)

Different types of interfaces realize these forms of management. These interfaces can be categorized into three levels, shown in the middle column of Table 1, and also on the right in **Figure 1**.

Table 1: Relationships between types of management and interfaces

Type of management	**Level of interface**	**Interface**
Management of the resources themselves	Resource level	CIM, SNMP, etc.
	Platform capabilities level	WSRF, WSDM, etc.
Resource management on the Grid	OGSA capabilities level	Functional interfaces
Management of OGSA infrastructure		Specific manageability interfaces

A detailed description of each level and its interfaces is given below. Note that the descriptions focus on the manageability interfaces, *not* on the locus of implementation (e.g., on the services that implement them). Also note that a service may implement multiple interfaces (which are possibly unrelated in terms of functionality), and that a service may be separated from the functionality that it represents (e.g., a manageability provider for a resource that is separate from this resource). Therefore a description based on services would be imprecise, and a description based on interfaces is chosen instead.

In Figure 1, the OGSA capabilities cover all levels, extending to capabilities in the resources that are needed to implement these OGSA capabilities. The interfaces are shown as small circles.

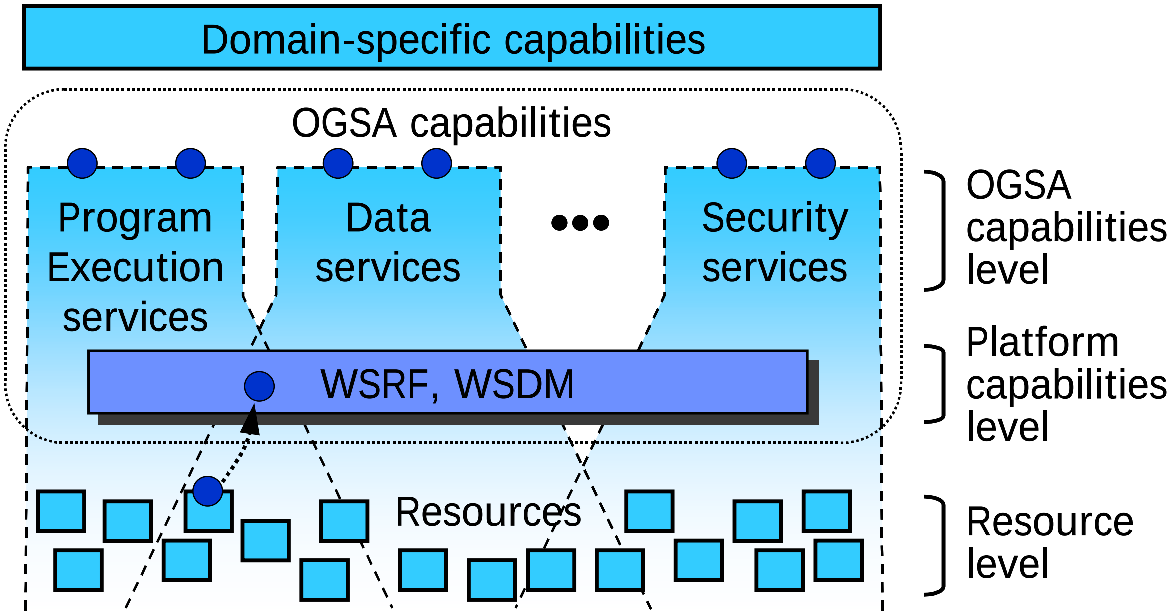

Figure 1: Levels of management in OGSA

At the resource level, the resources are managed directly through their native manageability interfaces (for discrete resources, these are usually SNMP, CIM/WBEM, JMX, or proprietary interfaces). Management at this level involves *monitoring* (i.e. obtaining the state of the resource, which includes events), *setup and control* (i.e. setting the state of the resource), and *discovery*.

The platform capabilities level provides the base management behavior of resources, forming the basis for both manageability and management in an OGSA environment. Standardization of this base management behavior is required in order to integrate the vast number and types of resources—and the more limited set of resource managers—that are introduced by multiple suppliers. The platform capabilities level provides:

- The base manageability model, which represents resources as services. This allows resources in OGSA to be manipulated through the standard Web services means for discovery, access, etc. The base manageability model includes resource identity (e.g., through endpoint references (EPRs)), the base manageability interface, resource properties to represent its attributes, etc. This model allows the resources to become manageable at least to a minimum degree, by enabling discovery, termination, introspection, monitoring, etc.

  Adopting a single framework in the base management also improves interoperability. For instance, if a Grid node is reserved, an application is deployed on it and the usage of this application is metered, the identities used by reservation, deployment and resource usage services must be common and refer to the same entities if interoperability is to be possible.

  It is important to note that the base manageability model is *not* itself a resource model – the resource model of the resources themselves is accessed *through* the base manageability model. This is shown in Figure 1 by the arrow linking the interface at the resource level to the interface corresponding to this resource at the platform level.

- Basic functionality that is common to the OGSA capabilities, e.g.:
  - portTypes for capabilities that are common to many resources (e.g., start, stop, pause, resume)
  - Lifecycle representation and operations
  - Relationships among resources
  - Aggregation (WSDM collection interfaces)

- o Metrics (meta-data on resource properties, such as timestamps of these properties)
- o Events

- A *generic manageability interface* that is common to all services implementing OGSA capabilities. This manageability interface has functionality such as introspection, monitoring, and creation and destruction of service instances.

At the OGSA capabilities level there are two forms of management (and interfaces), denoted by the two circles on the top of each of the capabilities shown in Figure 1:

- Some of the capabilities, such as resource management, are themselves a form of management. These capabilities (as all other capabilities) are accessed through a *functional interface*.

- Each capability has a *specific manageability interface* through which the capability is managed (e.g., monitoring of registries, monitoring of handle resolution, etc.). This interface should extend the generic manageability interface, adding manageability functionality that is specific to the management of this capability.

A simple example of these interfaces for a job manager service is given in Figure 2.

The functional and manageability interfaces are often not clearly separated (especially in the case of resource managers). However, a clear separation is desirable, since these interfaces are invoked by different users with different roles and access permissions. For instance, in Figure 2, the functional interface is used by the manager (or user) of the application being run (the “Grid administrator” in the Commercial Data Center use case [3]), and the manageability interface is used by the system manager (the “IT business activity manager” in [3]). [Add text on the increasing lack of distinction between manageability and functional interfaces.]

Manageability is often an afterthought, so often the functional interface is present but not the manageability interface. [Expand]

[TBD: draw a UML version of Figure 1]

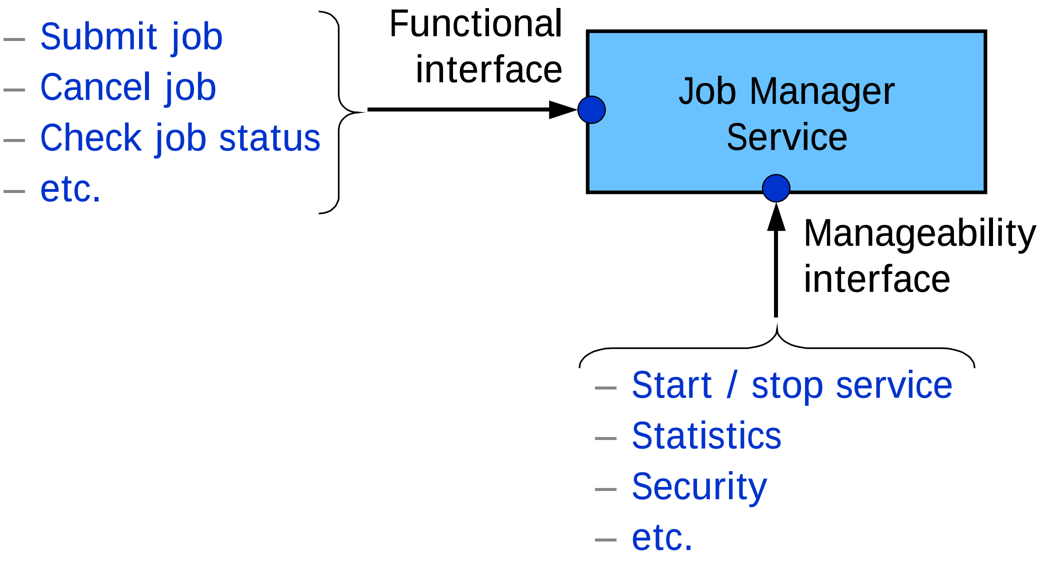

Figure 2: an example of the functional and manageability interfaces

The interfaces in both the platform and OGSA capabilities levels are both services, but they have a different nature. In the platform capabilities level the services are a wrapper around the manageability of a resource (mainly the resource model semantics), which provide a means to access this manageability. In the OGSA capabilities level the services provide functionality at a level higher than the features of the resources, or provide interfaces that don't access a resource model.

Discovery provides a concrete example of the differences between the resource, platform capabilities and OGSA capabilities levels. Discovery at the resource level might involve scanning a network to discover the devices attached to it. Discovery at the platform capabilities level can involve introspecting the service data of a service to find its capabilities. Discovery at the OGSA capabilities level might involve accessing one or more UDDI repositories that contain references to the available resources.

The division in levels helps interoperability between levels by defining clear interfaces between them. While it is possible to build services (implementing OGSA capabilities) that bypass these levels (e.g., using a proprietary adapter in a resource that feeds data directly to the service), that is not desirable from the point of view of interoperability, because, for example, it limits the kinds of resources with which the service and the adapter will be compatible.

4. Resource Models

Resource models describe resources by defining their properties, operations, and events, and their relationships with each other. Resources are managed (monitored, allocated, etc.) by following the description given by the model, and therefore resource models are essential to all facets of resource management. Resource models are used for both the functional and manageability interfaces.

Resource models are used for:

- IT system management
- Resource descriptions used mostly for resource management

Examples of resource models are:

- CIM, which includes models (*schemas*) for the following areas[2]:
    - Core: high-level abstractions (logical and physical elements, collections)
    - Physical: things that can be seen and touched (e.g., physical package, rack and location)
    - System: computer systems, operating systems, file systems, processes, jobs, diagnostic services, etc.
    - Device: logical functions of hardware (e.g., battery, printer, fan, network port and storage extent)
    - Network: services, endpoints/interfaces, topology, etc.
    - Policy: if/then rules and their groupings and applicability
    - User and Security: identity and privilege mgmt, white/yellow page data, RBAC, etc.
    - Applications and Metrics: deployment and runtime management of software and software services
    - Database: properties and services performed by a database (both inventory and behavioral)
    - Event: notifications and subscriptions
    - Interoperability: management of the WBEM infrastructure

[2] The work on JSIM (Job Submission Information Model, defined by the CGS-WG) was added to the schemas of multiple areas.

  - Support: help desk knowledge exchange and incident handling
  - Security Protection and Management: notifications for and management of intrusion detection, firewall, anti-virus and other security mechanisms
  - New work in the areas of Behavior and State (modeling state and transitions) and utility computing (management of utility computing services and related data for provisioning, accounting and metering, reservation handling, etc.)
  - [Ask Andrea: Is storage one of the above or a separate schema? How does SRIM fit in that (same as JSIM, "all over the place")?]
- SNMP MIBs [Add a list of existing functionalities]
- JMX's JSR77 [Add details]
- WSDM MOWS Web service model
- Resource descriptions for reservation/brokering/scheduling:
  - Unicore Resource Schema
  - Globus RSL and the GLUE schema [15]
  - JSDL (being defined by GGF's JSDL-WG)
- Resource descriptions for accounting/metering:
  - Usage Record (defined by GGF's UR-WG)
- Resource descriptions for installation/deployment/provisioning:
  - Configuration Description Language (CDL, being defined by the CDDML-WG)
  - DCML (Data Center Markup Language)

Note that although some resource descriptions are not intended to be models by themselves, they contain an implicit model which defines, for instance, which entities exist, and what their attributes are.

In resource models, it is important to make a distinction between *semantics* and *renderings*. The **semantics** contain the concepts of the model (its entities, their properties and relationships). A **rendering** is a representation of the semantics in a given language, and/or a specification of how to transmit and access the model on the wire. For instance, the CIM model contains the semantics of resources, and its XML representation and HTTP mapping are a rendering of CIM. A rendering of a model allows its semantics to be conveyed, and the semantics may have multiple renderings.

The semantics of a resource model contain its meaning, and thus they are more important in achieving interoperability than are its renderings: translating between two renderings of a single model is not a difficult problem, but translating between the semantics of two different models is likely to be complex. For instance, in different models a fan may be a physical or a logical entity; it may be classified under chassis, cooling devices, enclosure services or physical packaging; or it may have similar properties, such as a status, which have different value sets. Automatic translation between semantics can't be done unless these semantics are matched. An example of this matching is the mapping between Globus and Unicore resources being done as part of the GRIP project [15]. Also, CIM has mechanisms to map its semantics to those of other resource models [17].

Ideally, the use of a single resource model is desirable, since it makes interoperability easier to achieve when compared to mediation between models. However, developments in general IT and in Grid have so far not led to a total unification of the resource models, so it must be expected that multiple resource models will be in simultaneous use in a given Grid. Thus, coordination between models to make them compatible (as done with the GLUE schema), and mechanisms to match the semantics of different models, will have to be used. This is especially important for

OGSA, in which the functionality of a Grid is formed by the composition of multiple capabilities—each of them possibly using multiple semantics and/or renderings—which have to interoperate.

It is desirable that new resource models are created by re-using existing models, which not only allows higher interoperability but also requires less work. For instance, this new resource model could be created as a subset or superset of another resource model. Or, multiple resource descriptions could be created as renderings of a single resource model (with each resource description language representing this model, or a subset of it, using its own syntax, e.g., its own XML schema).

There are two areas in which there is need for coordination between resource models:

- Between the *resource descriptions* (to ease interoperability between OGSA services—reservation, metering, provisioning, etc.).
- Between the *standard management* models and the resource descriptions (to ease interoperability between resources and their resource managers).

Another desirable direction for work on resource models is model neutrality on the mechanisms for resource management. WSRF and WSDM are examples of these mechanisms.

5. Analysis of the OGSA Capabilities

The gap analysis has the objective of finding missing functionality on each level of manageability interface for each of the OGSA capabilities. Thus the gap analysis can be viewed conceptually as filling a table in which the rows are the management levels and the columns are the OGSA functions, as shown in Figure 3. Lack (or insufficient) contents in a cell indicates a gap. However, the analysis of the functional interfaces is one of the tasks of the OGSA-WG, and therefore this gap analysis will only cover manageability (the base manageability, generic manageability, and specific manageability interfaces). When applicable, models are analyzed for each capability.

Capabilities Levels	Program Execution services	Data services	●●●	Security services
Specific manageability I/F	(Section 5.3.1)	(Section 5.3.2)	●●●	(Section 5.3.7)
Generic manageability I/F	(Section 5.2)			
Base manageability	(Section 5.1)			
Models	(Section 5.3.1)	(Section 5.3.2)	●●●	(Section 5.3.7)

Figure 3: The Gap Analysis (conceptual view)

The gap analysis lists elements of a Grid that are candidates for management, and hence need to provide manageability interfaces. The list is intended to be used to identify the types of management actions that need to be possible, and the set of common manageability interfaces that are required. Some interfaces are expected to be defined already, while others will need to be specified. The list is derived in part from the current draft of the OGSA document.

The main objective of the gap analysis is to point out missing functionality. However, OGSA itself is still being defined, as are many of its underlying specifications, and in some areas work has not progressed sufficiently to allow the analysis to be completed. There are also cases where specifications have been completed, but it is not clear that they will be adopted. In such cases the gap analysis will point out items for which future analysis is required.

5.1 Base Manageability

In an OGSA Grid, all manageable resources either are, or are represented by, Web services. By definition, any Grid service exposes some interfaces that are useful in management - e.g. its termination time and the ability to change it (possibly causing immediate termination); the handle of the factory service that created it; a means of retrieving a list of its service data elements and the ability to query them, to change them, or to request notification if any of them changes. [Convert this to WSRF]

The WSRF specifications and related specifications such as WS-Addressing, plus WSDM MUWS, will provide the core functionality for the base manageability interfaces, as follows:

- WSRF
  - [TBD]
- WSDM MUWS: the following functionalities are among those currently being investigated: [double-check with Heather, and add better descriptions]
  - Identity
  - State
  - Metrics
  - Notifications and events
  - Relationships between resources
  - Collections
  - Discovery of manageability
  - Resource types
  - Configuration
  - Correlatable names
  - Meta-data representation
  - Capability extension
  - Composability of WS-Security
- WSDM MOWS: the following functionalities are among those being investigated (in addition to the ones in MUWS)
  - Identification
  - Request processing state
  - Managing operations
  - Sessions

The following gaps have been identified:

- Manageability functionality (and possibly resource models) needs to be defined for the services in the platform services level:

  - It may be important to identify general factory services as such, so that they can be managed in the same way as other key infrastructure services.
  - If specialized notification and event services are defined they will need to be managed as critical infrastructure services.
  - The OGSA document lists Agreement Services for Jobs, Reservations and Data Access. All are likely to be based on the WS-Agreement specification, but each is likely to have specialized interfaces, and may require specialized management. Their correct operation and performance will be critical to a Grid, and must be monitored.
- Mapping from WSDM to other models: WSDM is creating a Web service model in MUWS and defining its mapping to MOWS; however the mapping from MUWS to other models (e.g., CIM and SNMP, and Grid-related models) are not part of their charter, and need to be defined.
- Grid-specific functionality that could be missing and needs future analysis:
  - Grid-specific events
- Profiles: WSRF and WSDM are model-independent, and therefore there is the need to choose a set of resource models to be used [what else] to allow interoperability. Given that agreement on a single resource model cannot be expected, probably a set of profiles will have to be defined.
- Whether MOWS is enough to manage the services in an OGSA Grid or if there are special requirements needs to be verified.
- It needs to be investigated whether the state model of WSDM is suitable for the capabilities in OGSA (e.g., job control).
- The original CMM plans included “canonical services factored out from across multiple resources or domain specific resource managers, such as an operational port type (start/stop/pause/resume/quiesce).” This specific interface (start/stop) can be realized by the canonical state operations of WSDM. The need for other sets of canonical interfaces should be investigated (they are not among the planned functionality for WSDM).

[Anything special about factory services in WSRF?]

5.2 Generic Manageability Interface

Any service in OGSA will provide interfaces for at least minimal management - e.g. termination, introspection and monitoring. The OASIS WSDM TC will define some other standard manageability interfaces for Web services (MOWS) that should be applicable to services in OGSA. However we will need to determine if there are additional general interfaces that are specific to the Grid space.

Security is pervasive, and some activities on the management of security apply to all services. For instance, the management of access permissions to the service for different roles (end-user, managers, etc.), and of the protocol bindings to be use. Also, any service may suffer a denial-of-service attack and ideally the manageability should indicate such facts. Such manageability interfaces may need to be defined for all services.

[What else falls here?]

5.3 Specific Manageability Interfaces

This section analyzes the specific manageability interfaces, plus the models that are specific to a given capability.

The following items detail some specific services, and why it will be important to manage them.

[This section needs to have (1) a description of why it is important to manage these services (2) an analysis of the manageability interface and (3) models involved. No need to describe the functional interface beyond what is needed for (1) to (3) above.]

[The text below is still mostly unchanged from the previous version and need to be edited to the format above. Also, the analysis does not need to (and often shouldn't) tackle the services one by one: e.g., data services can consist of a single item, same for security services.]

5.3.1 Program Execution Services

The Program Execution (PE) services are composed of services in both Platform and OGSA capabilities levels, as follows.

The job, container and vault are resources. They have a resource model that defines their capabilities and properties, which is shown as services through WSDM. The job document corresponds to manageability information of the job (i.e., attributes/properties).

The actual resource in the case of a container, e.g., a node, has some manageability interfaces provided by the actual container, e.g. the operating system. However, these interfaces are probably not enough to realize all the functionality needed, such as deployment, and therefore manageability providers realize these interfaces. This manageability provider can be, and probably is, semantically close to these resources, i.e., the functionality that they provide is close to the one of the actual container. The same applies to vaults.

A job is a resource, however it has differences when compared to containers and vaults. A job has been defined as being created before resources have been committed and the actual execution is taking place, so at the time of its creation it is not known which container, e.g. node, will execute it. Therefore, it is not possible to realize the manageability of a job only through a manageability provider that is close to the actual resource, e.g., to realize the interface only through a manageability provider at the actual host that is running the process. Possibly the job manager service will contain the manageability provider for jobs. (This problem can be circumvented by allowing the EPR for jobs to be changed as jobs are associated with containers, migrated, etc.)

The job manager, execution planning service, candidate container set generator, information services, deployment and configuration services, reservation services are services at the OGSA capabilities level.

The resources (job, container and vault) have interfaces with functionality that enables the functionalities of services at the OGSA capabilities level, but don't implement these functionalities. For instance, the resources have functionality to enable migration, but they do not implement migration by themselves. Also, the job manager provides interfaces to reschedule a job, which is not a capability of the jobs themselves, or provides interfaces to operate on a set of jobs at the same time.

The following gaps have been found:

- It needs to be investigated whether the job manager shows the same interface of the jobs to its clients (e.g., by extending the interface for the jobs), which would provide a "one-stop shop" to control jobs.
- The manageability provider for the job is not close to the actual job. It needs to be analyzed whether an interface close to the actual job (implemented by a manageability provider in the actual container) should also be defined. This would improve the interoperability between the job managers and the containers.
- The resource models for the job, container and vault need to be defined. Existing resource models (CIM, GLUE, Unicore Resource Schema, etc.) should be analyzed and re-used.

- Resource models and interfaces for license management need to be defined. Again, existing resource models should be analyzed and re-used.
- Monitoring services need to be defined; its relationship to the OGSA information services (especially the producer and consumer interfaces) and GMA need to be taken in consideration.
- There is research on mapping Grid-related models among themselves, and mapping standards (e.g., CIM and SNMP) among themselves, but there is currently no work to the authors' knowledge on mapping the Grid-related models with the standard ones.
- On manageability interfaces:
    - A manageability interface for job management exists (JSIM). It needs to be analyzed if its functionality is sufficient for the needs of the PE services.
    - Other manageability interfaces, e.g. for controlling and monitoring Grid workflows, will be needed.
- [Anything from the items below should be added above?]**Job Management.**
    - Functional interface: Program Execution services will need to provide a way for managers to be notified as jobs are started, and either the jobs themselves or the execution services must provide an interface that allows the jobs to be managed – e.g. terminated, suspended or migrated. The Job Agreement Service [OGSA 6.20] may provide the required interfaces.
- **Queuing Service.** The OGSA document currently defines a queuing service as being a mechanism for scheduling jobs according to local policy, and it may be regarded as a part of the overall job management and execution subsystem. A manager may need to monitor the status of individual resource queues, and to be able to control them - e.g. to move jobs between queues to balance loads, to override priorities and to accommodate planned downtime.

5.3.2 Data Services

[Start by Data area gap analysis, then consult the Data design team of the OGSA-WG]

- **Data Services.** The "Base Data Services" and "Other Data Services" OGSA categories describe services that provide data representation and transformation facilities (Base Data Services), and facilities for accessing, transferring and managing replicas. In many Grids such services may be numerous and diverse; they will be fundamental to most, if not all, Grids. They will be critical infrastructure services, and their availability and performance must be monitored and managed.
    - Functional interface: [DataDescription, DataAccess, DataFactory in the OGSA Data services (August 2003 version), and extended interfaces above them?]
    - Specific manageability interface: [DataManagement interface in the OGSA Data services (August 2003 version), and extended interfaces above it?]
    - Models: Work progressing on SRIM (scope?)

5.3.3 Core Services

[Most services assigned to Jay, without clear definitions. We will probably need to do the analysis ourselves]

- **Virtual Organizations.** VOs can be considered as very-high-level manageable entities, and will provide significant management challenges. A manager will need to be able to discover and manage VO registries, create and destroy VOs, and manage the set of resources and users assigned to an individual VO. *There's much more we can say here.*

  - Functional interface: [From OGSA spec] The functional interface provides functions for creation and destruction of VOs, associating entities such as users, groups, and services with a VO, manipulation of user roles within the VO, attachment of agreements and policies to the VO.
  - Specific manageability interface: [Needed? Or does the functional interface include this functionality already?]
  - Models: [some sort of model might be needed for the interfaces above. Say, CIM user management?]

- **Policy Management.** A Policy subsystem, when fully defined, is likely to be composed of multiple related services, including a repository. The subsystem will be a critical infrastructure component of most Grids, and the ability to monitor it and to control certain elements will be essential.
  - Functional interface: Agreement interface
  - Specific manageability interface: need interface to manage (add, remove, change, etc.) the policies of resources to perform the management of policies themselves [already in WSDM?]
  - Models: [IETF/DMTF model?]

5.3.4 Information Services

[Consult mainly Hiro Kishimoto and Bill Horn]

- **Registry Services.** Registry services are likely to be deployed in every Grid. A service must be able to register itself in one or more registries so that it can be discovered, and so that its interfaces and capabilities can be queried. It is important that Registry services are available, and that they operate correctly, so managers will need to be able to monitor their operation and performance, and to create and destroy instances and copies as needed. A primary Registry service is likely to be the starting point for discovering and mapping, and hence managing, all resources in the Grid.
  - Functional interface: (none that concerns management – it is not a resource management functionality)
  - Specific manageability interface: needed for monitoring (as stated above)
  - Models: need a simple model to support monitoring?
- **Logging Services.** Logging services are essential infrastructure services, and they must be managed accordingly. It will be necessary not only to monitor their performance, but also to deal with storage space thresholds, low-space or insufficient-space conditions, periodic purging, access control, and many other facets. Different management domains within a given Grid may have different policies for retention etc. It's likely that this will be one of the more complex management operations.
  - Functional interface: needed to send items to be logged (extensions to the producer and consumer interfaces?).
  - Specific manageability interface: needed for management tasks such as setting the retention period, erasing logs, etc. (extensions to the producer and consumer interfaces?) In the current proposals, the manageability interface is not clearly separated from the functional interface.
  - Models: [TBD]

- **Messaging and Queuing.** If separate messaging and queuing services are defined, it is likely that they will become critical infrastructure services. Management requirements will include monitoring performance and managing the number of available instances and copies to handle the message volume and, if applicable, storage space.
    - Functional interface: (none that concerns management – it is not a resource management functionality)
    - Specific manageability interface: monitoring
    - Models: simple model needed for monitoring?
- **Information and Monitoring Service.** There are contents there (such as persistency and archives) which are not yet covered by WSDM/CMM or OGSA.
    - Functional interface: [TBD]
    - Specific manageability interface: [TBD]
    - Models: [TBD]

5.3.5 Resource Management Services

- **Service Configuration, Installation, Deployment & Provisioning.**
    - Functional interface: The CDDLM working group will address how to describe configuration of services, deploy them in a Grid, and manage their deployment lifecycle (instantiate, initiate, start, stop, restart, etc.). Managers will need the ability to configure, deploy, redeploy (relocate, perhaps with a different configuration) and terminate applications and other types of services within Grids, using the interfaces defined by CDDLM. Installation and Provisioning may be separate issues.
    - Specific manageability interface: [exists? Needed? Or does the functional interface include this functionality already?]
    - Models: CDL, DCML. [Relationship with the DMTF utility computing WG?]
- **Metering/Rating/Accounting/Billing & Payment.** These services all relate to measuring resource usage, and accounting and charging for it – they will not be applicable to all Grids.
    - The Metering service is effectively an infrastructure service - it must be permanently available if resource usage is to be recorded and charged for, and hence the manager must be able to monitor and control its operation as for any other critical service.
    - The Rating and Accounting services might be considered as application-level services – they are likely to be run periodically, reading and processing persistent (logged) data, and hence can be managed in the same way as any application-level service.
    - The Billing & Payment service will be a critical service for Grids that require it. This service may be internal or external, or may be an internal service that makes use of external services, such as credit card authorization services. Where needed, it will be essential that this service is operational, and a manager must be able to monitor and control it.
    - Functional interface: Resource usage service [enough?]
    - Specific manageability interface: [exists?]
    - Models: Resource usage [anything else?]

[Postpone analysis of the two below until they solidifiy]

- **Fault Management.** A manager will need to be notified of faults, and to be able to handle them to some level. This has not yet been addressed by OGSA, and it’s not clear if this would be implemented as a persistent service, or what its requirements for management might be. [OGSA should probably define the mechanisms to allow fault management (e.g., monitoring and control interfaces), but not the policies (e.g., what to do when a job crashes)]

  - Functional interface: [not clear, address again later]
  - Specific manageability interface: [not clear, address again later]
  - Models: [not clear, address again later]

- **Problem Determination.** A Problem Determination service, if available, is likely to be used by a manager, but may not be persistent, and its requirements for management are not clear. Not yet addressed by OGSA. [Same comment as above]
  - Functional interface: [not clear, address again later]
  - Specific manageability interface: [not clear, address again later]
  - Models: [not clear, address again later]

5.3.6 Self-management services

[Postpone analysis until it solidifies]

5.3.7 Security Services

The security services that compose OGSA (and their interfaces) are currently being defined by the OGSA-WG.

Services such as authentication and authorization will need to be managed, and may need specialized manageability interfaces.

There is currently no discussion on models. However, the schema (and the knowledge on manageability behind it) in existing models such as CIM (e.g., the User and Security schema, and the Security Protection and Management schema) could be useful for (and used in) the manageability for security.

5.4 Analysis of selected services

The following analysis goes into more detail on gaps on services that are critical for OGSA.

TBD

6. Conclusion

TBD

6.1 Summary of Gaps

TBD

6.2 Future Work

Many of the OGSA capabilities, their inter-relationships, and the standards on which OGSA is based are currently in evolution. This work takes a snapshot of their current state and performs the gap analysis. However, this work may need to be revised to reflect refinements and evolution in OGSA and related standards. It is hoped that this work in its current state serves as guidance for this refinement and evolution.

The analysis related to the gaps found, e.g., the definition of interfaces and models and analyses of whether existing functionality is sufficient, is better done by the groups responsible for the respective areas, since they have the knowledge to do this analysis. It is expected that external support related to models will be needed, as currently done in the CGS-WG. [Need to verify text to see if there is something that is up to us to do]

7. TBD

- Change text from OGSI to WSRF.
- How far does the OGSA-WG (or the GGF) need to define manageability of the OGSA infrastructure? E.g.: performance monitoring of a registry. Also: who is supposed to define it, how and when?
- Does management of policies (or management through policies) fit in this document, and if so, where?
- Introduction to the gap analysis. Also: what are the questions being asked in the gap analysis? “What is missing?” “What is critical?” “What needs to be done?”
- Go into more detail on items under “Basic functionality that is common to the OGSA functions”. E.g., relationships: “a way to discover relationships”, “a way to describe relationships”. The same applies to events.
- Check if all acronyms are defined in the text

8. Security Considerations

As mentioned in section 3.1, security is among the main requirements on management. Security is one of the many management functionalities covered in this document.

Author Information

Editor: Frederico Maciel (fred-m@crl.hitachi.co.jp)

Contributors: Frederico Maciel, Jem Treadwell, Latha Srinivasan, Andrea Westerinen and Ellen Stokes.

Thanks to Hiro Kishimoto, Bryan Murray, Ravi Subramaniam, Lawrence Flon, Jeffrey Frey and Bill Horn, and also to the members of the OGSA-WG, for discussions that helped improve the contents in this document.

Glossary

The definitions in Section 2 provide a brief glossary of OGSA management terms. Refer to the OGSA Glossary of Terms [12] for further definitions of related terms.

Intellectual Property Statement

Full Copyright Notice

References

1. OGSA document
2. Organization for the Advancement of Structured Information Standards (OASIS): http://www.oasis-open.org
3. Web Services Distributed Management Technical Committee: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm
4. Open Grid Services Architecture Use Cases, GWD-I (draft-ggf-ogsa-usecase-2.0) https://forge.gridforum.org/docman2/ViewCategory.php?group_id=42&category_id=351
5. OASIS WSDM TC, "Web Services Distributed Management: Management of Web Services (WSDM-MOWS 0.5)," Committee Draft, 2 April 2004
6. OASIS WSDM TC, "Web Services Distributed Management: Management Using Web Services (MUWS 0.5)," Committee Draft, 2 April 2004
7. Geoffrey Fox, David Walker, "e-Science Gap Analysis," http://www.grid2002.org/ukescience/gapresources/GapAnalysis30June03.pdf

8. Geoffrey Fox, David Walker, "Appendix : UK Grid Services and Activities, e-Science Gap Analysis," http://www.grid2002.org/ukescience/gapresources/Appendix30June03.pdf

9. Susan Malaika, Andre Merzky, Reagan Moore, "GGF Data Area Structure and Function Analysis," http://www.gridforum.org/Meetings/ggf10/GGF10%20Documents/ggfdataarea.structureandfunctionanalysis.Feb2004.pdf

10. B. Tierney et al., "A Grid Monitoring Architecture," GGF GFD-I.7

11. R. Aydt et al., "A Simple Case Study of a Grid Performance System," GGF GFD-I.8

12. OGSA Glossary

13. MUWS-Req

14. P2P Requirments doc

15. GLUE: http://www.hicb.org/glue/glue.htm

16. J. Brooke, D. Fellows, K. Garwood, C. Goble, “Semantic Matching of Grid Resource Descriptions,” http://www.grid-interoperability.org/

17. Distributed Management Task Force, “Common Information Model (CIM) Specification, Version 2.2,” http://www.dmtf.org/standards/cim

18. Other models?