GWD-TYPE
Category: TYPE
Common Management Model (CMM) WG

Author-1, Institution
Author-2, Institution

March 4, 2004
[if applicable: Revised DATE]

Resource Management on Grids

Status of This Memo

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

Copyright Notice

Abstract

Document Abstract.

Contents

1. Introduction

Grid computing involves many forms of management: many of its technologies are, or are related to, some form of systems management. Consequently, Grid computing has many overlaps with the management of IT systems in general. Some examples of these overlaps are job management, storage management, security management, etc. As Grid technologies are increasingly adopted on IT systems in general, the overlap will become bigger and the differences will probably blur.

A second aspect of management in Grid computing is the management of the Grid itself. While it is possible to build a Grid with entities that are not manageable, that is simply not practical. For instance, any problem on the Grid would require the whole Grid to be reset, since no form of problem determination and fault recovery would be possible. Also, adaptive functionality would be impossible, since it needs both monitoring and control.

This document first organizes the interfaces, services, activities, etc. that are involved in management in OGSA, including both management *within* OGSA and the management *of* the OGSA infrastructure. Then, based on this organization, this document does a gap analysis of manageability in OGSA, looking mainly for missing functionalities.

The work in this document is intended to build upon the work being carried out in the OASIS Web Services Distributed Management (WSDM) Technical Committee (TC) [1, 2]. 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 addresses management *of* Web services (MOWS) and Management *using* Web services (MUWS).

[To the OGSA-WG: it is hoped that its contents can help the discussions in the OGSA-WG. For instance, at times people have asked about “management”, which means many things in OGSA, or talked about “resources” and “resource managers”, which have never been clearly defined.]

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 without human intervention.

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 of the entity to allow 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 this entity, such as starting it, 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.).

[To the OGSA-WG: the definitions below are the important ones for OGSA, and need reviewing]

Manageable Resources (or simply *Resources*) means the same as manageable entities (including entities—such as software licenses—that do not expose generally-useful manageability interfaces, but may still be managed by some other means).[1]

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
- monitoring (performance, availability, etc.)
- control (start, stop, etc.)
- problem determination, fault management

[The items in these categories will be refined as the OGSA specification matures.]

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

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

[Message from Fred to the OGSA-WG: very often we have seen drawings with a “resource manager” box somewhere in the picture. This is very ambiguous – calling a box a “resource manager” does not say anything beyond “it does whatever needs to be done”. In my opinion, we should not use this term unless it is unambiguous.]

3. Management in OGSA

3.1 Requirements

[To the OGSA-WG: this section could be appended to section 3.5 of the OGSA document if deemed suitable].

[The MUWS requirements document of OASIS WSDM has a lot of requirements that can be used here. The text below gets the most important ones (enough for the OGSA-WG spec?). We could expand the list for the gap analysis document].

The main requirements for management in OGSA (for resource management or otherwise) are:

[1] It must be noticed that the term resource is often applied only to manageable entities that are pooled (e.g., hosts, software licenses, IP addresses, etc.) or entities that provide a given capacity (e.g., disks, networks, memory, etc.). In this case, some part of the pool and the capacity may be allocated and used. In this definition of the word resource a process, a print job, a registry service and a VO are not resources. Notice that this is a subset of the definition of resources as manageable entities.

- Scalability: management 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, for instance, can be done 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), i.e.:
    - Acting as a proxy that allows a manager to perform the same action on multiple resources with a single request
    - Computing metrics for the resources (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)
- Interoperability: management must be able to span software, hardware and service boundaries (e.g., across the boundaries between different products), so interoperability is essential to avoid "stovepipes." Two kinds of interoperability are needed:
    - between levels: e.g., between a resource and its manager;
    - on the same level: to allow the use of other resource management services 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 to both Grid-specific standards and also 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.
- *[Any other high-level requirements? GMA has also the following requirements: low latency, high data rate, low overhead. WSDM also mentions consistency and co-existence with other management standards.]*

3.2 Levels

[Temporary blanket statement: this section needs to be modified based on the expected changes to the OGSA classifications of services and/or interfaces. It should be a re-factoring of the contents below, so it should be OK to proceed with the gap analysis].

[To the OGSA-WG: this section can be added to the OGSA spec as an introduction to the resource management section (currently section 4.4), followed by description of the resource managers].

Management in OGSA consists of many interfaces, which can be categorized in three levels, as shown in **Figure 1**:

- Resource level
- Platform functions level
- OGSA functions level

A detailed description of each level is given below. It must be noticed that the description will focus on the manageability interfaces (which correspond to the OGSA taxonomy), *not* on the locus or hierarchy of implementation (which correspond to the OGSA hierarchy).

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

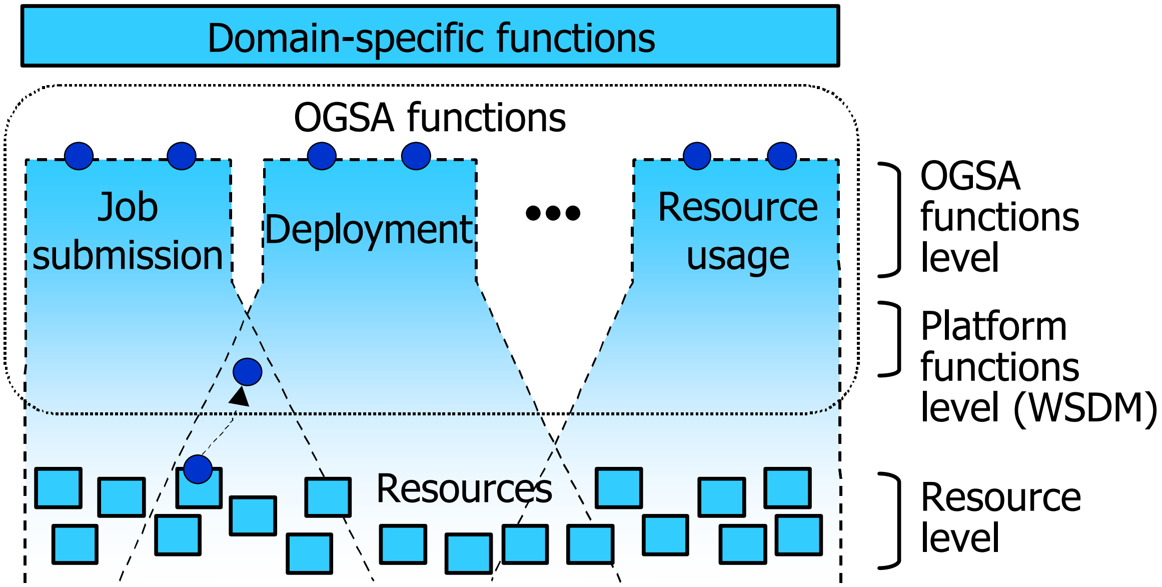


Figure 1: Levels of management in OGSA

At the resource level, the resources are managed through their manageability interfaces (for discrete resources, these are usually SNMP, CIM/WBEM, JMX, or proprietary interfaces). This 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 functions 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 parties. The platform functions level provides:

- The base manageability model, which represents resources as services. This allows resources in OGSA to be manipulated through the standard means for discovery, access, etc. The base manageability model includes resource identity (e.g., through GSHs), the base manageability interface, service data to represent its attributes, etc. This model allows the resources to become manageable to a minimum degree (allowing discovery, termination, introspection, monitoring, etc.).

  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 functions, e.g.:
  - o portTypes for functions that are common to many resources (e.g., start, stop, pause, resume)
  - o Lifecycle representation and operations

    - Relationships among resources
    - Aggregation (WSDM collection interfaces)
    - Metrics (meta-data on resource properties, such as timestamps of these properties)
    - Events
- A *generic manageability interface* that is common to all services implementing OGSA functions. This manageability interface has functionality such as introspection, monitoring, and creation and destruction of service instances. [This is probably WSDM MOWS.]

[The functions above are what the CMM-WG was created to define. They are currently being developed in the OASIS WSDM TC].

At the OGSA services level there are two forms of management, denoted by the two circles on the top of each of the functions shown in **Figure 1**:

- Some of the functions, such as the resource managers, are themselves a form of management. These functions (as all other functions) are accessed through a *functional interface [term kept as "functional" to align with WSDM]*.
- Each function has a *specific manageability interface* through which the function 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 functionality. In the case of the resource managers, this interface might not be clearly separated from the functional interface.

[TBD: draw a UML version of Figure 1*]*

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

The division in levels helps interoperability between levels by defining clear interfaces between them (interoperability on the same level is realized by defining the functional interfaces of the services). While it is possible to build services (implementing OGSA functions) 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—for example, it limits the kinds of resources with which the service and the adapter will be compatible.

4. Resource Models

[To the OGSA-WG: this section tries to make clear what role resource models have in OGSA, and how the OGSA-WG should handle them. It is not clear if this section fits in the OGSA spec (and if it does, where). It's still in a very preliminary state.]

Resource models are used for:

- IT system management
- Resource descriptions targeting mainly resource management

Examples of resource models are:

- CIM, which includes models (*schemas*) for the following areas:
    - 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
  - 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.)
  - [Is storage one of the above or a separate schema? How does SRIM fit in that?]
  - [JSIM (Job Submission Information Model, defined by GGF's CGS-WG) was added to which of the above?]
- 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
  - GLUE schema
  - 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)
- [Anything else?]

It must be noticed that some of the resource descriptions are not intended to be models by themselves, but they contain an implicit model (which defines, for instance, which entities exist, and what their attributes are).

[The contents of the paragraphs below are definitely a subject for discussions. First, I'm giving conclusions before offering a proof, but I think that some of you have war stories and scars that justify them. Second, even if we all agree with its contents, we need to fine-tune them to make them stronger and/or sharper. I'm accepting both opinions and text.]

Ideally, the use of a single resource model is desirable, since it makes interoperability easier to achieve when compared to mediation between models. [Insert examples]. However, usually multiple resource models will be in simultaneous use in a given Grid. Thus, it is highly desirable that the semantics of these resource models are coordinated to make them compatible. For instance, these multiple resource models should be a subset of a single resource model. Or, multiple resource descriptions should be "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).

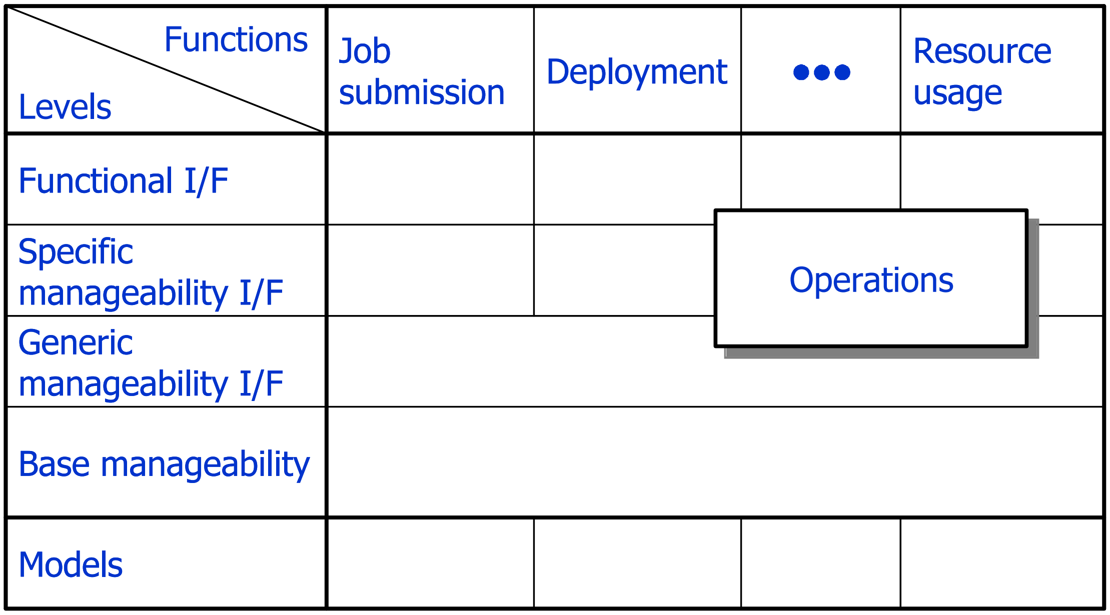
There are two areas in which there is need for coordination between resource models:

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

[What does GRIP do to bridge Globus and Unicore models? Anything to say about GLUE? How far has the ontology research gone?]

5. Analysis of the OGSA Functions

The gap analysis can be viewed conceptually as a table in which the rows are the management levels and the columns are the OGSA functions, as shown in Figure 2.



Functions Levels	Job submission	Deployment	●●●	Resource usage
Functional I/F				
Specific manageability I/F				
Generic manageability I/F				
Base manageability				
Models				

Figure 2: The gap analysis (conceptual view)

The following list calls out 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 Section 6 of the current draft of the GGF OGSA document, dated November 3rd, 2003.

In an OGSA Grid, all manageable resources either are, or are represented by, Grid 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 query them, to change them, or to request notification if any of them changes.

5.1 Function-independent analysis

This comprises the base manageability and the generic manageability interfaces, which are common to all OGSA functions. [Still very preliminary]

- **General Grid Services.**
    - Generic manageability interface: Any Grid service 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 that should be applicable to a Grid service. However we will need to determine if there are additional general interfaces that are specific to the Grid space.

5.2 Function-specific analysis

This comprises the specific manageability interface and the functional interface, plus the models that are specific to a given functionality.

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

- **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?
- **Handle Resolution Services.** Most Grids will need to provide access to one or more Handle Resolver services, and their performance will be critical to overall performance of the Grid. It should be possible for a manager to locate all HandleResolver services associated with the Grid, to monitor performance – e.g. request-volume and response times – and to create and destroy instances or copies as needed.
    - Functional interface: [not present in WSRF – ignore].
    - Specific manageability interface: [not present in WSRF – ignore]
    - Models: [not present in WSRF – ignore]
- **Factory Services.** 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.
    - Functional interface: (none that concerns management – it is not a resource management functionality)

  - Specific manageability interface: needed for monitoring
  - Models: need a simple model to support monitoring?
- **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]
- **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?]
- **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.
  - Specific manageability interface: [Need to be defined, to provide values such as job failure rates, etc. Quite close to the functional interface]
  - Models: See section 4.
- **Choreography, Orchestration & Workflow.** Some management functions will be needed for controlling and monitoring Grid flows, but we'll need better definition in the OGSA document. This is probably low priority.
  - Functional interface: [TBD]
  - Specific manageability interface: [exists? Needed?]
  - Models: [exists? Needed?]
- **Transactions.** As for Choreography, Orchestration & Workflow.
  - Functional interface: [TBD]
  - Specific manageability interface: [TBD]
  - Models: [TBD]
- **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?]

- **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]
- **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?)
  - 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?

- **Event Services.** [OGSA 6.15] If specialized event services are defined they will need to be managed as critical infrastructure services.
  - Functional interface: (probably provider and consumer interfaces, or similar)
  - Specific manageability interface: [any?]
  - Models: We will need common event semantics for interoperability
- **Policy & Agreements.** 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?]
- **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: CIM
- **Agreement 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.
  - Functional interface: [TBD]
  - Specific manageability interface: [TBD]
  - Models: [TBD]
- **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.
  - Functional interface: [TBD]
  - Specific manageability interface: [TBD]
  - Models: [TBD]

- **Security Services.** Security services are not yet fully addressed in detail by OGSA, but services such as authentication and authorization services will need to be managed, and may need specialized interfaces.
  - Functional interface: [Not clear, but probably needed]
  - Specific manageability interface: [Needed for management of security]
  - Models: [TBD]
- **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 Analysis of selected services

TBD

6. Gaps

TBD

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.
- Virtualization? We will certainly have multiple forms of it around. It might be worth writing about it to make the term more concrete.
- Make the differences between interfaces, services and functions very clear in the text (add text to explain the differences).
- 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.

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

With contributions from Jem Treadwell, Andrea Westerinen and Ellen Stokes, and help from Latha Srinivasan, Bryan Murray, Ravi Subramaniam, and Hiro Kishimoto.

Glossary

Recommended but not required.

Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

Full Copyright Notice

References

1. Organization for the Advancement of Structured Information Standards (OASIS): http://www.oasis-open.org
2. Web Services Distributed Management Technical Committee: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm