Frederico Buchholz Maciel,
Hitachi America, Ltd.

July 22, 2007

Guidelines for Information Modeling for OGSA™ Entities

Status of This Document

This memo provides information to the Grid community on information modeling for OGSA (Open Grid Services Architecture) entities. It has recommendations on the process of developing information models for OGSA entities and how to express these models in OGSA specifications. It does not define any standards or technical recommendations. Distribution is unlimited.

Abstract

An information model is an important and fundamental piece of the OGSA architecture since it provides consistent semantic meaning for entities on the architecture. This allows the integration and interoperability of the multiple services and multiple kinds of resources participating in an OGSA system. This document contains a process to create information models for OGSA entities based on methodologies used with the Common Information Model (CIM) of the Distributed Management Task Force (DMTF). The process in this document is based on years of experience in the Global Grid Forum and tested through a proof-of-concept study. This document explains the steps on the modeling process, division of tasks within the OGF, and coordination of work between the OGF and the DMTF. Finally, the appendixes contain an introduction to CIM and to the DMTF.

Contents

1. Introduction

OGSA services span multiple areas (execution management, data services, security services, etc.) and multiple activities in these areas (reservation, brokering, scheduling, provisioning, metering, control, etc.) over multiple kinds of resources (hosts, network devices, file systems, activities, etc.) and services [1, 2, 3]. Concepts such as “what is a host” or “what is processing load” need to have consistent semantic meaning in order to unify the architecture. Information models define such concepts by defining entities, their properties, operations, events, and their relationships with each other. An information model for OGSA entities allows the integration and interoperability of the services and resources participating in an OGSA system, and is consequently an important and fundamental piece of the architecture.

This document explains the process used to create information models for OGSA entities. This process is based on methodologies used with the Common Information Model (CIM) of the Distributed Management Task Force (DMTF), and consists of selecting, re-using and extending a small subset of CIM to develop information models (please see the appendixes for an introduction to the DMTF and CIM). It also gives guidelines on how to express information models in OGSA specifications.

The process in this document is based on years of experience in developing information models based on CIM in the Global Grid Forum (GGF). Especially, this process has been tested for OGSA through a proof-of-concept study described in [4] that created part of the information model for the Execution Management Services.

This document contains a general explanation that is applicable to all OGSA capabilities (execution management, data, security, etc.). Information modeling for specific capabilities requires, in addition to the contents of this document, capability-specific knowledge which is not covered by this document.

1.1 What Is an Information Model

An information model is abstraction and representation of entities in a data processing environment. It defines the entities, and also their properties, operations and relationships. This definition can use an informal natural language such as English, and/or a formal language such as UML. An information model is independent of any specific implementations, platforms, protocols, or repositories. For instance, CIM itself is an information model – it is simply a UML model, with textual descriptions of its contents defined in MOF (Manageable Object Format) files.

An example of a part of an information model represented in UML is shown in Figure 1 (this example is purely illustrative and does not correspond to actual CIM classes or to an information model used for OGSA entities). It contains two classes for two kinds of entities, Directory and FileSystem. Each of them has a series of attributes, such as Name, and each attribute has a type. The classes are just a generic representation, and there might be multiple entities (instances) for a class. For example, there might be multiple file systems in a computer; each of these is an instance of FileSystem, but all are represented by the same FileSystem class.

There are two kinds of relationships between Directory and FileSystem. First, there is an association called Mount that links a Directory instance with a FileSystem instance mounted under it. There is also a FileStorage aggregation that contains all Directory instances in a FileSystem instance. The information model contains also the cardinalities for this association and aggregation (e.g., a Directory may have zero or one FileSystems mounted under it; a FileSystem may be mounted under zero or more Directories). While not obvious from this example, these relationships are useful for discovery of entities and system structure.

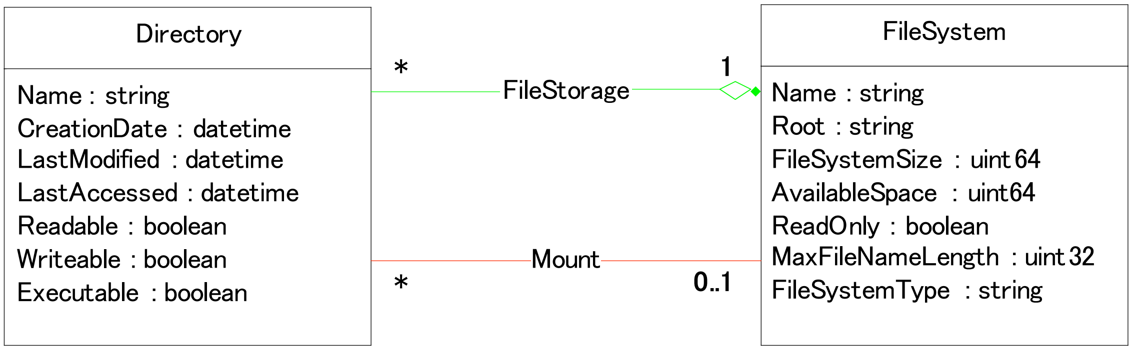

Figure 1: An example of part of an information model

In contrast with an information model, a *data model* is a representation of the information model in a given language, and/or a specification of how to transmit and access the information model on the wire. Thus, a data model allows the information model to be conveyed. To this end, a data model renders an information model according to a specific set of mechanisms for representing, organizing, storing data. The data model may also define operations that can be applied to the representation, such as data retrieval and update, enumeration of entities, etc. Finally, a data model may also define the legal states (set of values) or changes of state (operations on values). For instance, CIM has a data model composed by an XML representation and an HTTP mapping. Multiple data models may exist for a given information model. CIM has also a data model based on the WS-CIM standard that can be used with WSDM and WS-Management (see http://www.dmtf.org/standards/wbem/ for details).

An example of a part of a data model is shown in Figure 2, which is a XML Schema (see http://www.w3.org/XML/Schema) representation for FileSystem and Mount in Figure 1. Part of the XML representation for their instances is shown in Figure 3. Again, these examples are purely illustrative. This example assumes that instances can be addressed by Endpoint References (EPRs, see http://www.w3.org/TR/ws-addr-core), so the Mount association uses EPRs to point to the instances of Directory and FileSystem.

```
<xs:schema ...>
...
  <xs:element name="FileSystem">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Name" type="xs:string"/>
        <xs:element name="Root" type="xs:string"/>
        <xs:element name="FileSystemSize" type=" xs:unsignedLong"/>
        <xs:element name="AvailableSpace" type=" xs:unsignedLong"/>
        <xs:element name="ReadOnly" type="xs:boolean"/>
        ...
      </xs:sequence>
    </xs:complexType>
  </xs:element>
...
  <xs:element name="Mount">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Antecedent" type="wsa:EndpointReferenceType"/>
        <xs:element name="Dependent" type="wsa:EndpointReferenceType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
...
</xs:schema>
```

Figure 2: An example of part of a data model

```
<FileSystem>
      <Name>DataCDROM</Name>
      <Root>/media/cdrecorder</Root>
      <FileSystemSize>314572800</FileSystemSize>
      <AvailableSpace>0</AvailableSpace>
      <ReadOnly>true</ReadOnly>
      ...
</FileSystem>
...
<Mount>
      <Antecedent>
      ...  <!-- EPR of the instance of Directory -->
      </Antecedent>
      <Dependent>
      ...  <!-- EPR of the instance of FileSystem -->
      </Dependent>
</Mount>
```

Figure 3: Instances represented in the data model

The term *resource model* implies both information and data models and thus is often confusing; this term is now deprecated in OGSA nomenclature. The terms *semantics* and *rendering* used so far in OGSA modeling correspond respectively to “information model” and “data model”; these terms are also deprecated to simplify the nomenclature and improve clarity. Finally, it must be noticed that the definitions of information and data model above match RFC 3198 [5] but do not match RFC 3444 [6].

1.2 Why an Information Model Is Needed

As mentioned above, an information model is important to provide consistent meaning to entities, their properties and inter-relationships. However, it is easier to understand such a statement by an example in which this consistency does *not* exist. Assume that a job manager is consulting a resource selection service to find a suitable place to run a job:

- If the job manager asks for the host with lowest “processing load” assuming it means the average number of processes in the last 15 minutes, and the selection service makes the choice based on a different interpretation of processing load (e.g., instant CPU load), the selected host will often not be the one expected by the job manager.
- If the job manager requests all hosts with 1 GB or more of “free memory” assuming it includes memory currently used as cache that can be re-used for program data, and the selection service assumes that free memory is totally unused memory, the chances of a match will be reduced.

The examples above are intentionally simple and they trivialize the problem since they only cover properties. Bigger problems will happen if there is no common understanding of the entities (e.g., if a host can be virtual or only physical, and to which of these “processing load” and “free memory” apply), and their relationships (e.g., whether job queues are related to sites, clusters, sub-clusters or hosts).

The information model is what provides a common unambiguous understanding of the entities, their properties and inter-relationships, and consequently allows interoperability in exchanges of this information between services, between clients and services, and between services and resources. This makes possible the integration and interoperability of the services and resources participating in an OGSA system.

1.3 Relationship Between Information and Data Models

Information and data models will be present in multiple OGSA interfaces. They should appear prominently in the interfaces of the information services (which should organize and provide data according to the information and data models) and manageability interfaces. However, they will sometimes be present in functional interfaces: for example, an interface to retrieve job status will return one of a set of possible states that are defined by an information model, and return this state in a format defined by a data model. The fact that classifications on functional/manageability interfaces or "information services" are often imprecise or overlapping does not change the premises above.

Care must be taken to avoid interoperability problems among the multiple services using interfaces related to information and data models. Information models contain the meaning of the representation of entities, and thus they are more important in achieving interoperability than data models: translating between two data models of a single information model is not a difficult problem, but translating between two different information models is likely to be complex. For instance, in different information 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 information models cannot 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 [7] (see also http://www.grid-interoperability.org). Also, CIM has mechanisms to map its schema to those of other information models [8].

The *target* for OGSA entities should be:

- One information model, in order to unify the concepts in the whole architecture and avoid translation of semantics.
- One "main" data model per basic profile. There should be as much commonality between these data models as possible, e.g., common XML schemas across basic profiles, and common parts to the WSDL to access the information, to simplify translation. Programmatic translation from the information model to the data model is also desirable. However, specific data models may be created for functional interfaces. This is not critical since they are often specific to a given capability and/or can be later mirrored in a manageability interface using the "main" data model.

It is expected that work in progress on the DMTF such as WS-CIM should become a good basis for a "main" data model to use in OGSA interfaces, however such a data model is currently an open problem. This does not delay the development of functional interfaces, which can continue using specific data models.

This document is concerned only with information models. It mentions how the information model is surfaced as data models in OGSA services and interfaces; however it does not prescribe where and how to create data models.

2. Modeling Process

The use of the CIM methodology as a starting point for the creation of information models for a specific area of the architecture implies the following work:

- Creating an initial proposal for the information model, possibly using existing information models as a reference;
- Selecting which parts of the schema to use for this area of the architecture—i.e., creating a profile for this area;
- Creating extensions if and where needed.

These activities are detailed in the following sections.

2.1 Creation of Initial Proposal

It is very useful to start the modeling process by the creation of an initial proposal. This proposal, still informal, identifies what is in the domain of this specific model, gives an idea of the work in the following phases, and identifies portions of CIM to use, change and extend. This proposal also helps to start the discussions on modeling and start collaboration with the DMTF, and also on model requirements with related Working Groups (WGs).

This initial profile will often involve comparing CIM to existing Grid-related information models and creating mappings between them. This work:

- Aids in finding features in these models that are not yet in CIM, which then become candidates for extensions.
- Provides interoperability with work that has been completed, such as GLUE (see http://glueschema.forge.cnaf.infn.it/ and current work in OGF's GLUE-WG).
- Provides synchronization of the specifications being compared and unification for work in progress in the OGF.

It must be noticed that this comparison and mapping work is not restricted to models such as GLUE and the UNICORE Resource Schema [7]. This work also applies to specifications that contain data models with an implicit information model. JSDL (Job Submission Description Language) v1.0 [9] is an example of such a specification. Mapping between CIM and Grid-related standards can make use of the mechanisms in CIM to map it to other models [10].

2.2 Creation of Profiles

CIM was never intended to be used as a whole, and in practice never *is* used as a whole; instead, CIM profiles are used to define which parts of the model are used for specific areas. These profiles specify which CIM classes, properties and methods to use, map these to entities, and provide guidance on their usage. OGSA entities are no exception: coverage is needed for the ones that are in scope for OGSA specifications. OGSA entities are often an abstraction of real entities, so the higher level of abstraction is in scope but the details are not. For example, a file system and its attributes are within scope, but knowing that it is accessed through a specific adapter in the third slot of the second expansion bus is not. The strategic use of profiles reduces the view of CIM to a small subset that is meaningful for OGSA entities.

Profiling brings several advantages compared to the development of a new model:

- It allows faster definition of the model. Although an information model seems simple and obvious after it is complete, modeling is time-consuming work that can often takes years even for the definition of just a handful of classes. The re-use of existing CIM classes through profiling saves time by leveraging all the model development (discussions, spec-writing, testing, etc.) already done in the DMTF, and the use experience.
- When information models are defined for new OGSA areas, there is no need for integration with, or retrofitting of, previously created profiles since the classes are already integrated in CIM.

In OGSA specifications one should expect one or more profiles to be created for each major area of the architecture (execution management, data, etc.). OGSA profiles related to information models are called "Information Model Profiles," and follow the same rules of other OGSA profiles. These profiles are referenced from the specifications contained in these OGSA profiles as shown in Figure 4. Given that the different specifications will not be developed simultaneously even within a single OGSA major area, the creation of information model profiles will be done in a piecewise fashion, selecting the model as the work of each OGSA capability progresses, and in a bottom-up fashion, starting from more basic entities.

OGSA profiles may also refer to OGSA information model profiles in case the information model applies to the whole profile. A possible example is the definition of a manageability interface for some of the entities in the profile.

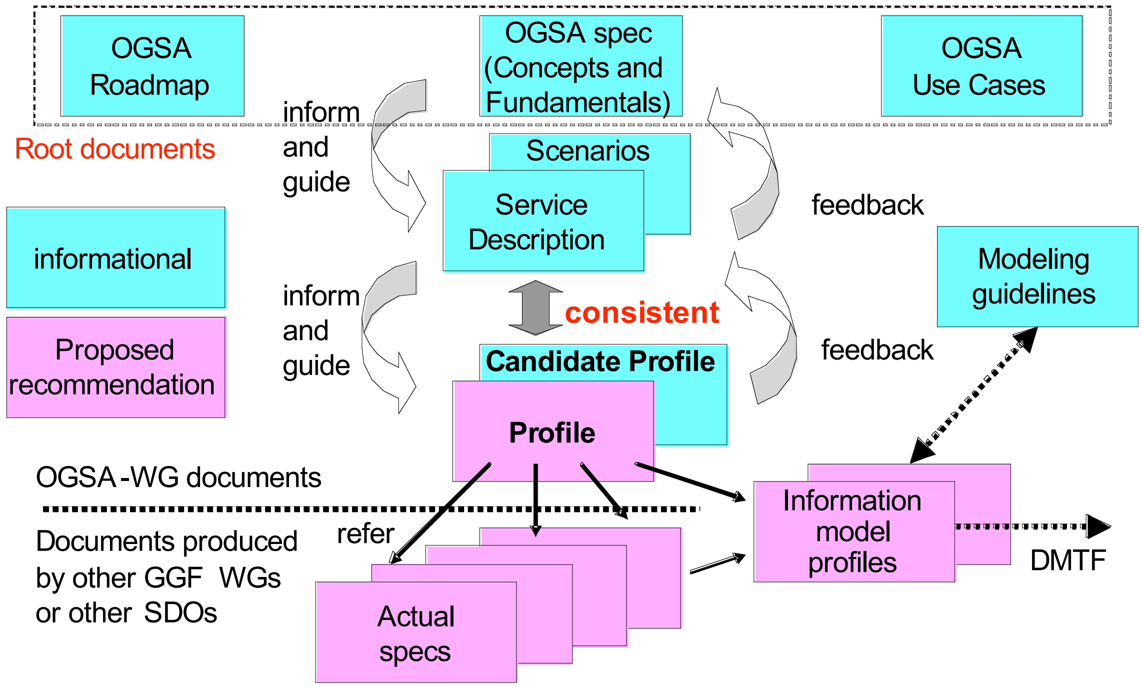

Figure 4: OGSA specification structure with information model profiles. This document corresponds to the "Modeling guidelines."

The information model profiles may be created by the OGSA-WG or by other WGs creating OGSA specifications. However, it is expected that an information model profile will contain entities in multiple OGSA specifications, so the general case should be their creation by the OGSA-WG, collecting entities from these specifications and adding other entities that might be needed. Also, the OGSA-WG should act as the coordinator of information modeling activities in the architecture, to avoid inconsistencies in the overall OGSA information model for different specifications and/or different areas of the architecture.

2.3 Creation of Extensions

While CIM already models a wide range of entities, it does not cover all the needs of OGSA entities, so extensions will be created where needed. Also, OGSA specifications will continue to be extended and refined for years, and these changes will probably require additions to the model.

Similarly to the work on profiling, these extensions will sometimes be created by the OGSA-WG, but in some cases may also be created by WGs developing OGSA-related specs.

Extensions should be created in collaboration between the OGF and the DMTF, with the OGF providing the area expertise and the DMTF providing the modeling (and CIM model) expertise. This addresses the issue of the complexity of CIM and the lack of knowledge and experience with CIM by OGF WGs. The DMTF will be mainly responsible for the development of CIM and all parts thereof. The OGF WGs will be mainly responsible for delineating the needs of their specifications.

While any extensions to CIM created for OGSA entities can be left as OGF-only standards (becoming thus OGF specifications, and OGF "proprietary" extensions to CIM), it is strongly recommended that these extensions are submitted to the DMTF, and referenced from OGF specifications. This allows the integration of the OGSA extensions in the wider CIM schema, and prevents incompatibilities that could result from further CIM extensions. These submissions should be done within the collaboration between the DMTF and the OGF, and moved through the standardization process in the DMTF by the DMTF participants, as explained in Section 4.

Extensions should be defined in OGF documents, either informational or recommendations, to have them reviewed by the OGF membership. Adding the extensions to information model profiles makes reviewing easier and allows the profiles to provide context on the extensions. These documents are used as change request documents in the DMTF, through the process explained in Section 4. In case the extensions are defined as OGF informational documents, the OGSA profiles and information model profiles should refer to the extensions on DMTF specifications after they are adopted by the DMTF.

3. Roles and Responsibilities

There are multiple groups involved in the work above in profiling, extending and comparing information models. This section explains the roles and responsibilities of these groups.

The development of the models requires both area expertise (i.e., expertise on the entities, attributes and relationships) and modeling expertise (i.e., expertise on CIM, on how to create a model and correlate it to existing work). Therefore, the modeling work requires participation from the OGF workgroups and design teams, which provide the area expertise (e.g., requirements), and from the DMTF, which provides modeling expertise. Conversely, this organization also eliminates the need for the workgroups to have modeling expertise (especially, knowledge of CIM), which could be a big barrier to development.

The modeling work is an iterative process among all parties, starting from the initial proposal and gradually progressing to the final specification. New versions of a given specification may be created as work progresses on a given OGSA capability. These updates should modify the information model in a way that is backwards-compatible (see section 2.3 of [10] for a list of such modifications).

3.1 Resource Management Design Team and OGSA Workgroup

The Resource Management (RM) design team of the OGSA-WG acts as coordinator of information modeling for OGSA entities, and also in some cases develops the models. To this end it works in collaboration with both the OGF workgroups and also with the DMTF.

The RM design team determines which profiles are needed. These profiles should align with the work in a given OGSA capability, e.g., with the specifications in development or expected boundaries of implementation and use.

Given that the RM design team coordinates model development, it can provide the linkage between the information model profiles, which is needed for integration of the information model not only within but also between OGSA capabilities. It must be noticed that CIM quite probably already provides this linkage, and as new parts of CIM are selected, the linkage with already-selected parts of CIM should also be selected. The centralized work is necessary also for a broad evolutionary view of the information model—for example, making possible the addition of more entities in different OGSA capabilities as work on OGSA progresses.

3.2 OGF Workgroups and Design Teams

There are two possible scenarios for WGs. The first one is the preferred case, in which a WG includes the development of information models (or possibly a candidate information model) in its scope. In this case this WG can collaborate directly with the DMTF in the creation of information models. However, participation from the RM design team is still needed as the coordinator of information modeling for OGSA entities. In case a WG creates the information model together with a specification, the information model profile should be written separately to ease the collaboration with the DMTF.

In the second case information models for a given area are not in scope of the specifications of any related WG. In this case, either the RM design team develops the information model or a spin-off WG is created to do the modeling. However, the related working groups have the knowledge of what entities, attributes, relationships, etc. are needed—i.e., knowledge of the requirements on the information model. So while these WGs will not develop information models themselves, they should provide these requirements.

The specifications created by working groups may at times describe and/or manipulate entities and attributes that are defined by the information model. As mentioned in Section 1.3, these working groups may define a data model to represent the information model in these specifications.

3.3 DMTF

The DMTF is the ultimate information model librarian—i.e., it maintains the information models created not only in the DMTF and OGF, but also in other standards bodies. The result of the information model developed by the RM design team is given to the DMTF for inclusion in CIM. The interplay between the OGF and DMTF is discussed in detail in Section 4.

4. Standardization Steps

This section analyzes the links between the standardization processes of OGF and DMTF during the development and review of information model profiles.

4.1 Review Process

As stated above, the DMTF is the ultimate librarian of the models, and so extensions have to pass through its standardization process. However, it has been deemed desirable to pass these extensions through the OGF process also, which creates links between the two.

The DMTF standardization process is described in [12], and can be summarized as follows. A WG and the Technical Committee (TC) may approve the release of Work in Progress as such or as Preliminary Standards. The recipients may be another WG, the DMTF membership, an Alliance Partner (e.g. the OGF) or the general public (Preliminary Standards released to the general public require also DMTF board approval). Feedback may be received on Preliminary Standards. If there is implementation experience from two independent implementations by two different parties, the TC takes the schema to the Final Standard phase (however, demonstration of interoperability between them is not needed).

The CIM schema is released in a slightly different process from the one above. Two versions of the schema are released simultaneously: the “Final Schema,” composed of CIM classes released as Final standards, and the “Experimental Schema,” which contains also classes that are still at the Preliminary standard stage. The classes in the Experimental Schema that are not Final are tagged as Experimental.

The extensions created in collaboration between the OGF and the DMTF can be sent for public comment in the OGF and made a Published Work in Progress in the DMTF to receive feedback from both standards bodies. These review periods may be simultaneous or they may overlap, but they do not need to. Collaboration between the OGF and DMTF is essential to avoid major changes being proposed in one of the standards bodies. The details on how to merge the feedback have to be analyzed on a case-by-case basis, but may require a repetition of the public comment process especially in case of major changes or if the reviews don’t overlap. Once feedback is addressed, the specification can be released as a Informational Document or Proposed Recommendation in the OGF and as a Preliminary Standard in the DMTF. Finally, implementation experience can make the extensions a Final Standard in the DMTF.

4.2 Status Type and Adoption Level

The information model profiles contain both submitted extensions and existing CIM schema. This creates a relationship between the Status and Adoption Levels for OGSA profiles [11] and the standardization status in the DMTF, which is discussed in this section. OGSA Informational Profiles are not discussed here since they can be created for any level of Status and Adoption.

DMTF standards and OGSA profiles are related as follows. DMTF Published Work in Progress and Preliminary Standards have a Status of “Evolving Institutional” standard and an Adoption level of (at least) “Unimplemented”. Final Standards have a Status of “Institutional” standard and an Adoption level of (at least) “Implemented”—not “Interoperable”, since the implementation experience does not involve interoperability. So while the OGSA profile and DMTF standards are aligned on the status, they are not on the adoption level, and the latter becomes a key requirement for information model profiles. The requirements for each kind of information model profile then become:

- Recommended information model profile as Proposed Recommendation:
    - Status: DMTF documents may have Work in Progress, Preliminary or Final standard status; CIM classes may have any status (Experimental or Final).
    - Adoption Level: DMTF documents and CIM classes have an adoption level of "Interoperable".
- Recommended information model profile as Grid Recommendation:
    - Status: DMTF documents and all CIM classes must have Final standard status. Once these become Preliminary Standards in the DMTF, interoperability (already reached above) makes them a Final standard since there is enough implementation experience.
    - Adoption Level: DMTF documents and CIM classes must have an adoption level of "Community".

As for any OGSA profile, the versions of specifications referenced directly or indirectly by an information model profile must be consistent. All specifications must refer to the same major version of the CIM schema. For CIM classes, all specifications must refer to the same version of a given class as specified by the Version qualifier of this class. This means that different specifications may refer to different revisions of the CIM schema (e.g., 2.10 and 2.11) as long as the same version of each class is used. This eases the creation of an information model profile, since the schema is updated often and references to different revisions can easily happen. This sort of consistency should not be difficult to reach since revisions of the CIM schema only bring backwards-compatible changes.

4.3 Change Requests

Additions or changes to the CIM schema and to DMTF standard documents should be sent to the DMTF as Change Requests (CRs), which is the mechanism used in the DMTF for change control [12]. A CR may be sent to the related DMTF WG by a DMTF member (working as OGF liaison) or by an Alliance Partner (the OGF). As previously stated, the information model profile is submitted to the DMTF as the change request document.

CRs are discussed and approved by the WG and then sent to the TC. The TC may approve the CR or send it back to the WG with comments. CRs approved by the TC get reflected in DMTF standards.

5. Security Considerations

There are two security aspects in resource management that apply to information models. The first aspect is secure management, i.e. 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.

Access to the information described by the information model may need to be secured with mechanisms such as authorization and encryption. Access to the information may also be restricted to certain users or sites. However, these considerations are part of the data model and thus out of scope of this document, which focuses on information models.

Access to the information model may be restricted in different granularities: an instance, a class or a property or method. Such restrictions have to be considered during the development of the information model.

The second aspect is the management of security: the security infrastructure must be manageable; this includes the management of authentication, authorization, access control, VOs and access policies. The management of security is an important OGSA functionality, and information models for user management, certificates, etc. may be needed for entities and services related to security.

It must be noticed that all considerations above apply not only to manageability interfaces but also to functional interfaces.

6. Contributors

Editor:
Frederico Buchholz Maciel
Hitachi America, Ltd., Research & Development Division
750 Central Expressway, MS: 3224
Santa Clara, CA 95050
Phone: +1 (408) 970-4833
E-mail: Fred.Maciel@hds.com

Thanks to Ellen Stokes, Jem Treadwell, David Snelling, Hiro Kishimoto and to the members of the OGSA-WG, for contributions and discussions that helped improve the contents in this document.

7. Glossary

Section 1.1 explains some of the nomenclature used in this document. For the meaning of other terms, please refer to the OGSA glossary [2].

8. Intellectual Property Statement

The OGF 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 OGF Secretariat.

The OGF 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 OGF Executive Director.

9. Disclaimer

This document and the information contained herein is provided on an “As Is” basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

10. Full Copyright Notice

OGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.

11. References

1. I. Foster et al., “The Open Grid Services Architecture, Version 1.0,” Global Grid Forum, Lemont, Illinois, U.S.A., GFD-I.30, 2005, http://www.ogf.org/ggf_docs_final.htm
2. J. Treadwell (Ed.). “Open Grid Services Architecture Glossary of Terms,” Global Grid Forum, Lemont, Illinois, U.S.A., GFD-I.44, 2005, http://www.ogf.org/ggf_docs_final.htm
3. I. Foster, D. Gannon, H. Kishimoto, J. J. Von Reich (Ed.). "Open Grid Services Architecture Use Cases,” Global Grid Forum, Lemont, Illinois, U.S.A., GFD-I.029, 2004, http://www.ogf.org/ggf_docs_final.htm
4. E. Stokes, “Execution Environment and Basic Execution Service Model in OGSA™ Grids,” forthcoming in the OGF document series, 2007, http://www.ogf.org/ggf_docs_final.htm
5. A. Westerinen et al, “Terminology for Policy-Based Management,” Internet Engineering Task Force, Sterling, Virginia, U.S.A., RFC 3198, 2001, http://www.ietf.org/rfc.html.
6. A. Pras, J. Schoenwaelder, “On the Difference between Information Models and Data Models,” Internet Engineering Task Force, Sterling, Virginia, U.S.A., RFC 3444, 2003, http://www.ietf.org/rfc.html
7. J. Brooke, D. Fellows, K. Garwood, C. Goble, “Semantic Matching of Grid Resource Descriptions,” UoM. 2nd European Across-Grids Conference (AxGrids 2004), January 2004, Cyprus, http://www.grid-interoperability.org/semres.pdf
8. “Management Profile Specification Usage Guide,” DSP1001, Distributed Management Task Force, November 2005, http://www.dmtf.org/standards/published_documents/
9. A. Anjomshoaa et al., “http://www.gridforum.org/documents/GFD.56.pdfJob Submission Description Language (JSDL) Specification v1.0,” Global Grid Forum, Lemont, Illinois, U.S.A., GFD-I.56, 2006, http://www.ogf.org/ggf_docs_final.htm
10. “Common Information Model (CIM) Infrastructure Specification, Version 2.3,” DSP0004, Distributed Management Task Force, October 2005, http://www.dmtf.org/standards/cim/
11. T. Maguire, D. Snelling, “http://www.gridforum.org/documents/GFD.56.pdfOGSA Profile Definition,” Global Grid Forum, Lemont, Illinois, U.S.A., GFD-I.59, 2006, http://www.ogf.org/ggf_docs_final.htm
12. “DMTF Release Process, Version 1.5.0,” DSP4004, Distributed Management Task Force, June 2006, http://www.dmtf.org/standards/published_documents/
13. Chris Hobbs, "A Practical Approach to WBEM/CIM Management," CRC Press, 2004, ISBN 0-8493-2306-1

Appendix A: Background Information on CIM

CIM provides a common definition of management information for systems, networks, applications and services, and allows for vendor extensions. As mentioned above, CIM itself is only of the model semantics; CIM and its data models and protocols are known as WBEM (Web-Based Enterprise Management). CIM includes models (schemas) for the following areas[1]:

---

[1] The work on JSIM (Job Submission Information Model, defined by GGF’s CGS-WG) was added to the schemas of multiple 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 management, white/yellow page data, RBAC (Role-Based Access Control), etc.
- Applications and Metrics: deployment and runtime management of software and software services
- Database: properties and services performed by a database (addresses database components, backing storage, status and statistics)
- Event: notifications and subscriptions
- Interoperability: management of the Web-Based Enterprise Management (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
- Block and file storage
- Application Server: updates JSR77's CIM mapping, for managing the J2EE environment
- New work in the areas of Behavior and State (modeling state and transitions), and virtualization.

CIM as a whole is defined in several places:

- The definition of the CIM schema (the model itself) is at http://www.dmtf.org/standards/cim. The definition is composed of the UML model (available in PDF and Visio formats) and MOF (Managed Object Format) files. The latter contains a textual description of model, with:
  - A full definition of the structure of the model (structure, classes, properties, metadata, etc.) which can be input to CIM software as the definition of the model
  - Human-readable explanations of the classes, properties and methods
- The conceptual definition of CIM, including the meta-model, mapping to other information models, etc. is in [10].
- Profiles constrain the CIM schema and give further details on its usage for specific areas such as record logs, power supplies or boot control. This is needed because the CIM schema contains a generic explanation of the model but not enough detail on how to use it for each area. For instance, a profile specifies which classes and properties are used for the given area, and which classes are linked by which associations. It can also give a subsets of the states specified in the schema that apply to this area, and links this subset to the behavior or the managed entities.
- White papers also give additional information on the model and its usage for specific areas.

There are multiple mechanisms in CIM to map other information models to CIM. Currently there are mappings from CIM to SMBIOS, IETF MIBs, DMI MIFs, TMF (TeleManagement Forum) models, JSR77, and others.

CIM is updated 3 to 4 times a year. Starting in CIM v2.10, the schema is divided in “Final” and “Experimental” parts (the latter contain the Final and Preliminary parts of the schema). These frequent updates do not mean that the model is unstable – changes are backward compatible, usually consisting of additions on areas under development, which recently have been mainly storage management and server management. Even a major version-up of the model is backward compatible by mapping the new version to the previous one using the mechanisms to map to other information models.

CIM is one of the standards being created by the DMTF (Distributed Management Task Force). The DMTF is “the industry organization leading the development of management standards and integration technology for enterprise and Internet environments”. DMTF standards provide common management infrastructure components for instrumentation, control and communication in a platform-independent and technology-neutral way. The DMTF has more than 3,000 active participants. As of March 2007 there are 110 member companies, including most industry leaders in all areas of IT. There are also 14 alliance partner members (other organizations that collaborate with the DMTF), the OGF being one of them. There is also the Academic Alliance membership, a free membership for accredited institutions of higher learning, with 36 members. Academic Alliance Members have access to the DMTF members-only Web pages and member email lists, and are eligible to participate in DMTF working groups, in the DMTF Marketing and Technical Committees as a non-voting member. Every year the DMTF has invited all of its Academic Alliance Members to submit a paper on their work with DMTF standards, and a winner chosen by the DMTF Board (see http://www.dmtf.org/education/academicalliance/ for a list of papers submitted). Finally, the DMTF can have individual members who have to be sponsored by a member company. These multiple classes of membership allow most, if not all, active members of the OGSA-WG and related WGs can have access to information in the DMTF.

There has been collaboration on CIM between the GGF and the DMTF with many results:

- JSIM (Job Submission Information Model, GFD-I.028) was an extension of CIM for batch jobs created in the CGS-WG (CIM Grid Schema WG). It has been contributed to the DMTF, and is present in CIM 2.10.
- JSDL 1.0 bases its definition of a number of types (such as Operating System types) on CIM.
- DAIS-WG collaborated with the DMTF on the creation of SRIM (Software Resource Information Model) extensions.

Appendix B: A Brief Technical Introduction to CIM

This appendix gives a brief technical explanation of CIM that is only complete enough for the understanding of CIM-related OGSA documents, especially the diagrams. For a more detailed explanation, there is a very complete tutorial on the DMTF Web site (see http://www.dmtf.org/education). There are also books that give a good introduction to CIM, including some of its practical aspects [13].

Entities in CIM are represented in classes which have a name, and zero or more properties and methods. Properties are attributes of the entity that a class represents (e.g., CreationDate of a Directory in Figure 1). Methods define actions that can be performed in an instance of a class (e.g., start, stop, reset). The classes in Figure 1 do not contain any methods.

CIM is represented graphically in UML diagrams with extensions. CIM classes with names in italic font in diagrams are abstract, and are not meant to be instantiated. There are three different links between classes, represented with different colors in the UML diagrams:

- Inheritance: CIM is an object-oriented model with single inheritance, which is denoted by blue lines in the diagrams.
- Associations: these are links that show a relationship between classes in the schema, denoted by red lines. An instance of an association contains “pointers” to the instances of the classes it links. Associations usually link two instances, but can be n-ary (e.g., for devices connected to a SCSI bus). Interestingly, an association formally defined as a CIM

class. Consequently, it is identified by a name, thus a command in the data model to enumerate all the instances of a class can also be applied to associations, which can be very useful for instance to traverse the model for discovery. Also, being a class, an association may have properties (other than the "pointers") and methods, but in practice rarely do.

- Aggregations: this is a form of association used for containment or part/whole relationships, and denoted by green lines in the diagrams. It contains a "diamond" shape on the side of the containing class. A stronger form of association, defined in UML as *composition*, requires that the contained part exists in at least one of the aggregations. Compositions are shown by a filled diamond or a diamond and a dot. For instance, FileStorage in Figure 1 is a composition, which means that a Directory has to exist in at least one FileSystem.

It must be noticed that associations and aggregations can be used not only between the classes they link in the diagrams, but also their sub-classes. A somewhat extreme example of this is the ConcreteDependency association, which links ManagedElement (the top class of CIM) to itself. ConcreteDependency can thus be used to link any two sub-classes of ManagedElement, i.e., any two classes of the CIM schema.