Execution Environment and Basic Execution Service Model in OGSA™ Grids

Status of This Document

This document provides information to the Grid community on modeling an Execution Environment and a Basic Execution Service. It does not define any standards or technical recommendations. Distribution is unlimited.

Copyright Notice

Trademarks

OGSA is a trademark of the Open Grid Forum.

Abstract

This memo provides information to the Grid community on the information modeling of execution environment and a basic execution service for OGSA (Open Grid Services Architecture). It defines the proposed execution environment element and basic execution service element for inclusion in DMTF's Common Information Model. Specific execution environments, such as the OGSA Basic Execution Services execution environment, should be defined as profiles and/or extension elements of execution environment. Operations specific to a service should not be part of the execution environment. Distribution is unlimited.

Contents

1. Overview

The OGF Basic Execution Service (BES) specification [BES], GFD.108, defines Web Services interfaces for creating, monitoring, and controlling computational entities such as UNIX or Windows processes, Web Services, or parallel programs—what we call *activities*. Clients define activities using the Job Submission Description Language (JSDL). A BES implementation executes each activity that it accepts on an appropriate computational resource, which—depending on the BES implementation and the type(s) of activities supported—may be a single computer, a cluster managed through a resource manager such as Load Leveler, Sun Grid Engine, Portable Batch System, or Condo, a Web Service hosting environment, or even another BES implementation.

This document defines the DMTF/CIM model elements necessary to express the concepts of an execution environment and basic execution service abstracted from the OGF Basic Execution Service [BES].

The Execution Environment model describes the managed objects and their relationships for defining the execution environment for activities in a grid. The Basic Execution Service model defines the service. This service can act on the execution environment. The CIMv2.15 final schema [CIM2.15final] is the foundation for the development of these two models. It is expected that these two models will be folded into CIMv2.16 experimental [CIM2.16exp].

In this document we provide the models in UML and DMTF CIM Managed Object Format (MOF). DMTF provides the XML representation via an automatic conversion tool (MOF to CIM-XML via CIM DTD as well as WS-CIM).

2. The Execution Environment Model

OGSA defines an execution environment as a collection of resources in which a task can execute. An execution environment may be, for example, a queuing service, a Unix host, a J2EE environment, or a collection of specific resources. An execution environment may contain zero or more execution environments. An execution environment has resource properties that describe both static and dynamic information, e.g. OS version, types of executables allowed, policies, security, load, QoS information. An execution environment may implement some subset of the manageability interfaces such as Web Services Distributed Management (WSDM)[1] or WS-Management.[2] The execution environment has various relationships to other resources. The execution environment may use various services such as reservation services, logging services, information services, job management services, and provisioning services.

The existing CIM class ComputerSystem is used to model an execution environment (refer to Figure 1). The CIM definition of Computer System: *"ComputerSystem is a class derived from System that is a special collection of ManagedSystemElements. This collection is related to the providing of compute capabilities and MAY serve as an aggregation point to associate one or more of the following elements: FileSystem, OperatingSystem, Processor and Memory (Volatile and/or NonVolatile Storage)"*. This means that a very large number of attributes may be associated with an execution environment via the SystemComponent aggregation. The ComputerSystem class captures the basic semantics of an execution environment. It may be necessary to subclass ComputerSystem to express all the semantics of a specific execution environment, for example, to add an attribute or operation that is specific to that execution environment that is not already in a class aggregated by ComputerSystem. A service may act on an execution environment.

The state of whether the execution environment is accepting new activities is captured the use of the property EnabledState (enabled and disabled values) of EnabledLogicalElement. The concept

[1] www.oasis-open.org/committees/wsdm

[2] www.dmtf.org/standards/wbem/wsman

of an execution environment being contained within an execution environment is expressed via the HostedDependency association.

The execution environment for the Basic Execution Service (BES) aggregates and exposes attributes needed to make scheduling decisions. For ease of implementation, usability, extensibility, and interoperability, the details of what is contained in an execution environment should be defined by a profile.

Note: It is expected that additional CIM classes, attributes, and methods may be defined and need to be added to CIM as the specific execution environment profile work progresses. These will be documented in a separate informational document.

3. The Basic Execution Service Model

The Basic Execution Service is a service to which clients can send requests to initiate, monitor, and manage computational activities. Operationally, it uses information in a container and operates on activities within that container. This service is a subclass of the class CIM_Service.

The class added to CIMv2.16 experimental in support of this service is marked as 'Experimental' in the CIM MOF and as {E} in the UML diagram in Figure 1 and appears in bold font for easier identification.

The model elements are discussed in a bit more detail in Section 5 following the presentation of the UML in Section 4. The Managed Object Format (MOF) description is given in Section 6. (MOF is a textual rendering of UML, defined in the CIM Specification [CIMspec] published by the DMTF).

4. UML

The figure below depicts the classes and properties of the container model. The new class with its attributes and methods are highlighted in red **bold**.

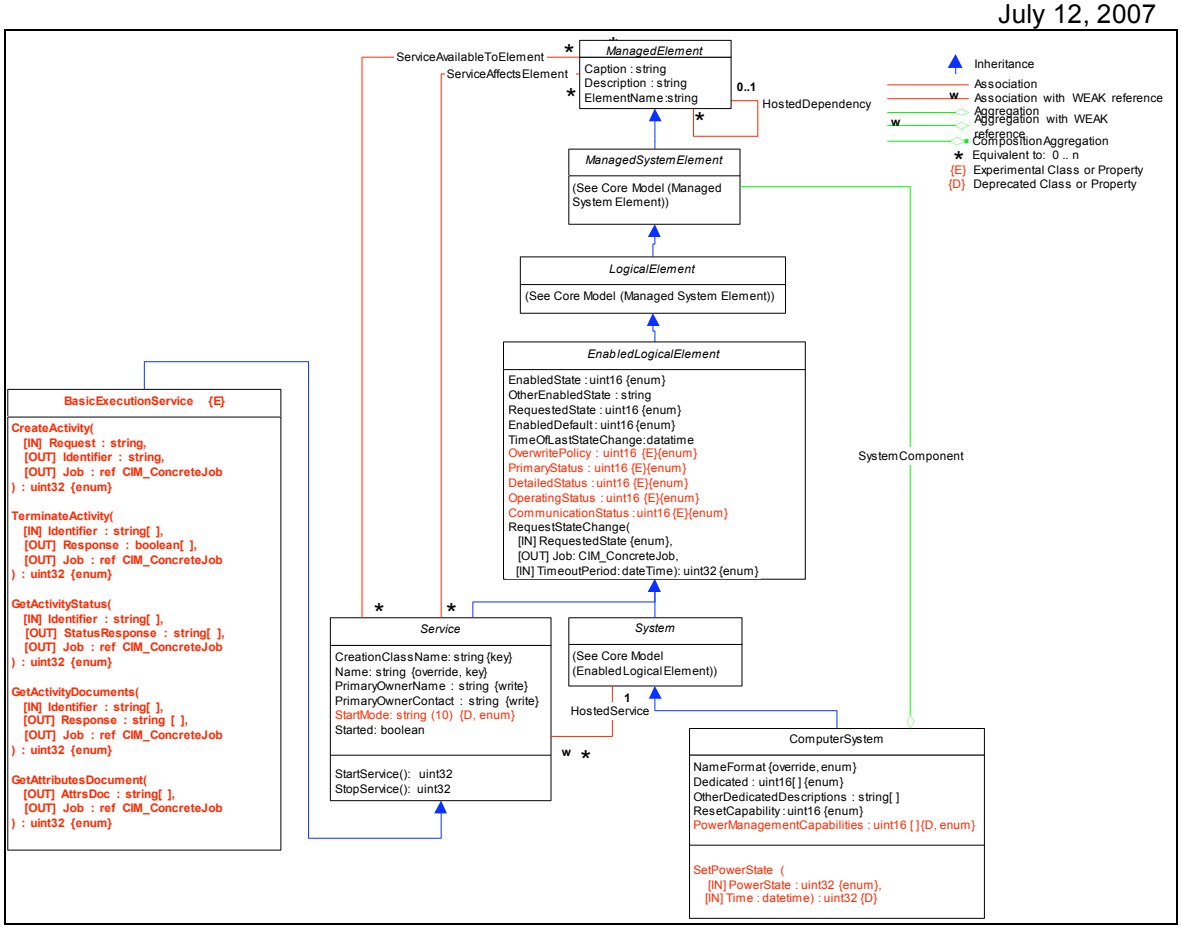

Figure 1 Execution Environment and Basic Execution Service Model

5. Discussion of the Model Elements

This section describes the classes, associations, properties, and methods proposed to be added to CIM v2.16 experimental in support of a execution environment model and a basic execution service model. Background material as well as details (inputs, output, formats) of the basic execution service operations can be found in the Open Grid Forum document 'Basic Execution Service', GFD.108 [BES].

5.1 New Classes

5.1.1 Basic Execution Service

The grid Basic Execution service inherits from the CIM_Service class. The Basic Execution service is related to Container through several existing associations with ManagedElement: (1) ServiceAvailToElement – the Basic Execution service is available within the Container, and (2) ServiceAffectsElement – the Basic Execution service uses resources within the container and hence those resources may affect performance, throughput, availability, etc. The StartService and StopService operations (inherited from CIM_Service) are used to move the container associated with this service into a state where it is open for requests and closed for requests, respectively. For example, within the context of the OGSA Basic Execution Services specification, this means the container can start accepting new activities, and the operation maps to 'StartAcceptingNewActivities()' interface and the container stops accepting new activities, and the operation maps to 'StopAcceptingNewActivities()' interface, respectively. The StartService and StopService operations set the attribute EnabledState (enabled or disabled) in the class EnabledLogicalElement in the execution environment.

- CreateActivity
  This operation adds requests to the container. For example, within the context of the OGSA Basic Execution Services specification, this means that a new activity is added to the container, and the operation maps to the 'CreateActivity()' interface.
- TerminateActivity
  This operation requests that one or more items in the container be terminated. For example, within the context of the OGSA Basic Execution Services specification, this means that a new or existing activity in the container can be requested to be terminated, and the operation maps to the 'TerminateActivities()' interface.
- GetActivityStatus
  This operation requests the status of one or more items in the container. For example, within the context of the OGSA Basic Execution Services specification, this means that the status of one or more activities within the container can be obtained, and the operation maps to the 'GetActivityStatuses()' interface.
- GetActivityDocuments

  This operation requests activity document descriptions for a set of specified activities. These activity documents may be different from those initially passed to this basic execution service in the CreateActivity operation since this service may alter its contents to reflect policy or process within the service. For example, within the context of the OGSA Basic Execution Services specification, this means that the activity documents can be obtained, and the operation maps to the 'GetActivityDocuments()' interface.
- GetAttributesDocument

  This operation requests a document containing the basic execution service management attributes within the associated container. For example, within the context of the OGSA Basic Execution Services specification, this means that the attributes within a container may be obtained, and the operation maps to the 'GetAttributesDocument()' interface.

A new UMLPackagePath – Grid – is defined for this service class defined to keep model extensions related to grid together and aid in the formation of a federated CIM model in the future.

5.2 New Associations

None.

5.3 Re-use of Existing Classes

The class ComputerSystem is used to model the concept of an execution environment. See Section 2 for details.

5.4 Properties Added to Existing Classes

None.

5.5 Methods Added to Existing Classes

None.

6. Managed Object Format (MOF)

The schema is described in Managed Object Format, defined in [CIMspec].

The MOF below reflects the UML diagram in this document. It is included to provide the details and descriptions necessary to understand the UML. It has been approved by DMTF for inclusion in CIMv2.16.

```
// ===================================================================
//  CIM_BasicExecutionService
// ===================================================================
   [Experimental, Version ( "2.16.0" ),
       UMLPackagePath ( "CIM::Core::Grid" ), Description (
       " The basic execution service (BES)is a service to which "
       "clients can send requests to initiate, monitor, and manage "
       "computational activities and access information about the "
       "BES.   A BasicExecutionService can act on one or more "
       "execution environments - modeled, profiled, and instantiated "
       "as a ComputerSystem.    There is no requirement that a "
       "BasicExecutionService reside on the node of a "
       "ComputerSystem on which it acts.   The "
       "associations ServiceAvailToElement and ServiceAffectsElement "
       "relate the BasicExecutionService to ComputerSystem.  The "
       "association HostedDependency expresses the concept that an "
       "execution environment may be contained within another "
       "execution environment. "
       "For example, in a grid or distributed/virtualized environment "
       "the whole point for not explicitly stating which execution "
       "environment to use up front is to allow some client software, "
       "e.g. scheduler, orchestrator, provisioner, application, "
       "to determine where to place "
       "the activity (in which execution environment) based on the "
       "input activity document (that activity's environment/resource "
       "requirements). ")]
class CIM_BasicExecutionService : CIM_Service {

      [Description (
          "This operation adds requests to the execution environment.  "
          "For example, within the context of the OGSA Basic Execution "
          "Services, this means that a new activity is added to an "
```

```
            "execution environment, and the operation maps to the "
            "CreateActivity() interface. "
            "CreateActivity establishes the 'binding' between the "
            "activity and the execution environment that will contain "
            "it.  Selection / implementation of how an execution "
            "environment is outside the scope of basic execution service. "
            "In a grid or distributed environment, this allows other "
            "clients, e.g. schedulers, orchestrators, applications, "
            "to make decisions on which execution environment to "
            "select (place activity) based on the JSDL job description "
            "(the input activity document that describes that activity's "
            "environment/resource requirements. "
            "The return value should be 0 "
            "if the request was successfully executed and some other "
            "value if an error occurred.  The output the CreateActivity "
            "method is an identifier which is used as input to other "
            "methods in this class to identify the activity being "
            "acted upon."),
         ValueMap { "0", "1", "2", "3", "4", "5", "6", "..",
                    "4096", "4097..32767"", "32768..65535" },
         Values { "Operation Completed with No Error", "Not Supported",
            "Unknown", "Not Authorized",
            "Not Accepting New Activities",
            "Unsupported Feature",
            "Invalid Request Message",
            "DMTF Reserved",
            "Method Parameters Checked - Job Started",
            "Method Reserved", "Vendor Specific" }"),
         MappingStrings
                { "MIF.OGF|GFD.108|CreateActivity.Faults" }]

      uint32 CreateActivity(

            [IN, Description (
                "Describes a single request that is to be executed by an "
                "execution environment."),
             MappingStrings
                 { "MIF.OGF|GFD.108|CreateActivity.ActivityDocument",
                   "MIF.OGF|GFD.56|jsdl:JobDefinition" }]
         string Request,

            [IN ( false ), OUT, Description (
                "Identifier associated with the requested execution.  "
                "This Identifier is used as input to other Basic "
                "Execution service methods. "),
             MappingStrings
                 { "MIF.OGF|GFD.108|CreateActivity.Response",
                   "MIF.OASIS|WS-Addressing",
                   "MIF.OGF|GFD.56|jsdl:JobDefinition" }]
         string Identifier,

            [IN ( false ), OUT, Description (
                "Reference to the job (can be null if the task is "
                "completed).")]
         CIM_ConcreteJob REF Job
      );
```

```
        [Description (
            "This operation requests that one or more items in an "
            "execution environment be terminated.  For example, within "
            "the context of the OGSA Basic Execution Services, this "
            "means that a new or existing activity in the container "
            "can be requested to be terminated, and the operation maps "
            "to the TerminateActivities() interface.  The return value "
            "should be 0 if the request was successfully executed and "
            "some other value if an error occurred.  "
            "The return code Invalid Request Message refers to the "
            "input of an invalid identifier. "),
         ValueMap { "0", "1", "2", "3", "..",
                    "4096", "4097..32767", "32768..65535" },
         Values { "Operation Completed with No Error", "Not Supported",
            "Unknown", "Invalid Activity Identifier",
            "DMTF Reserved",
            "Method Parameters Checked - Job Started",
            "Method Reserved", "Vendor Specific" }"),
         MappingStrings
                { "MIF.OGF|GFD.108|TerminateActivities.Faults" }]

    uint32 TerminateActivity(

          [IN, Description (
              "Identifies one or more items in an execution "
              "environment that are to be terminated."),
           MappingStrings
            { "MIF.OGF|GFD.108|TerminateActivities.ActivityIdentifiers",
              "MIF.OASIS|WS-Addressing" }]
        string Identifier[],

          [IN ( false ), OUT, Description (
              "Boolean response value for each requested termination.  "
              "A value of TRUE indicates successful termination."),
           MappingStrings
                { "MIF.OGF|GFD.108|TerminateActivities.Response" }]
        boolean Response[],

          [IN ( false ), OUT, Description (
              "Reference to the job (can be null if the task is "
              "completed).")]
        CIM_ConcreteJob REF Job
    );

        [Description (
            "This operation requests the status of one or more items in "
            "an execution environment.  For example, within the context "
            "of the OGSA Basic Execution Services, this means that the "
            "status of one or more activities within an execution "
            "environment can be obtained, and the operation maps to the "
            "GetActivityStatuses() interface.  The return value should "
            "be 0 if the request was successfully executed and some "
            "other value if an error occurred. "
            "The return code Invalid Request Message refers to the "
            "input of an invalid identifier. "),
         ValueMap { "0", "1", "2", "3", "..",
                    "4096", "4097..32767", "32768..65535" },
```

```
        Values { "Operation Completed with No Error", "Not Supported",
           "Unknown", "Invalid Activity Identifier",
           "DMTF Reserved",
           "Method Parameters Checked - Job Started",
           "Method Reserved", "Vendor Specific" }"),
        MappingStrings
               { "MIF.OGF|GFD.108|GetActivityStatuses.Faults" }]
    uint32 GetActivityStatus(

          [IN, Description (
              "Identifies one or more items in an execution "
              "environment whose status will be obtained."),
           MappingStrings
            { "MIF.OGF|GFD.108|GetActivityStatuses.ActivityIdentifiers",
              "MIF.OASIS|WS-Addressing" }]
       string Identifier[],

          [IN ( false ), OUT, Description (
              "A response for each requested status.  "),
           MappingStrings
               { "MIF.OGF|GFD.108|GetActivityStatuses.Response" }]
       string StatusResponse[],

          [IN ( false ), OUT, Description (
              "Reference to the job (can be null if the task is "
              "completed).")]
       CIM_ConcreteJob REF Job
    );

       [Description (
           "This operation requests activity document descriptions "
           "for a set of specified set of activities.  These activity "
           "documents may be different from those initially input in "
           "the CreateActivity operation since this service may alter "
           "its contents to reflect policy or process within the "
           "service. "
           "The return code Invalid Request Message refers to the "
           "input of an invalid identifier. "),
        ValueMap { "0", "1", "2", "3", "..",
                   "4096", "4097..32767", "32768..65535" },
        Values { "Operation Completed with No Error", "Not Supported",
           "Unknown", "Invalid Activity Identifier",
           "DMTF Reserved",
           "Method Parameters Checked - Job Started",
           "Method Reserved", "Vendor Specific" }"),
        MappingStrings
               { "MIF.OGF|GFD.108|GetActivityDocuments.Faults" }]
    uint32 GetActivityDocuments(

          [IN, Description (
              "Identifies one or more activities for which activity "
              "documents are requested."),
           MappingStrings
            { "MIF.OGF|GFD.108|GetActivityDocuments.ActivityIdentifiers",
              "MIF.OASIS|WS-Addressing" }]
       string Identifier[],
```

```
            [IN ( false ), OUT, Description (
                "An array of activity document response elements."),
             MappingStrings
                 { "MIF.OGF|GFD.108|GetActivityDocuments.Response" }]
        string Response[],

            [IN ( false ), OUT, Description (
                "Reference to the job (can be null if the task is "
                "completed).")]
        CIM_ConcreteJob REF Job
    );

        [Description (
            "This operation requests a document containing the basic "
            "execution service management attributes."
            "The return code Invalid Request Message refers to the "
            "input of an invalid identifier. "),
         ValueMap { "0", "1", "2", "..",
                    "4096", "4097..32767", "32768..65535" },
         Values { "Operation Completed with No Error", "Not Supported",
            "Unknown",
            "DMTF Reserved",
            "Method Parameters Checked - Job Started",
            "Method Reserved", "Vendor Specific" }"),
         MappingStrings
                { "MIF.OGF|GFD.108|GetAttributesDocument.Faults" }]
    uint32 GetAttributesDocument(

         [IN ( false ), OUT, Description (
               "A XML document containing the various attributes within "
               "its associated container.  "),
            MappingStrings

{ "MIF.OGF|GFD.108|GetAttributesDocument.BESResourceAttributesDocument",
                  "MIF.OGF|GFD.56|JSDL Core Element Set and Appendix 1" }]
        string AttrsDoc[]

            [IN ( false ), OUT, Description (
                "Reference to the job (can be null if the task is "
                "completed).")]
        CIM_ConcreteJob REF Job
    );

};
```

7. Security Considerations

This specification defines the model and XML Schema for containers and a basic execution service. While the interactions of containers with its activities must be secured, the security details are outside the scope of this specification. Instead, it is assumed that security is addressed in specifications that define how this model and XML Schema are bound to specific communication protocols (such as [CIMOPS]) and programming environments.

8. Contributions

The Author listed on the title page is committed to taking permanent stewardship for this document – receiving communication in the future and otherwise being responsive to its content. The contact information is provided below:

Ellen Stokes
IBM
11400 Burnet Rd
Austin, TX 78758
stokese@us.ibm.com
+1 512 778 5821

Other contributors include members of the OGF OGSA and BES workgroups and the DMTF CIMCore workgroup.

9. 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.

10. 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.

11. Full Copyright Notice

12. References

[CIM2.15final] CIM Schema, Version 2.15 Final, Distributed Management Task Force, April 17, 2007, http://www.dmtf.org/standards/cim/cim_schema_v215

[CIM2.16exp] CIM Schema, Version 2.16 Experimental, Distributed Management Task Force, August 2007, http://www.dmtf.org/standards/cim/cim_schema_v216

[CIMspec] "CIM Infrastructure Specification, Version 2.3", Distributed Management Task Force, October 4, 2005, http://www.dmtf.org/standards/documents/CIM/DSP0004.pdf

[CIMOPS] "Specification for CIM Operations over HTTP, Version 1.2", DSP0200, Distributed Management Task Force, January 9, 2007, http://www.dmtf.org/standards/published_documents/DSP200.html

[BES] I. Foster, A. Grimshaw, P. Lane, W. Lee, M. Morgan, S. Pickles, D. Pulsipher, C. Smith, M. Theimer, "OGSA Basic Execution Services, Version 1.0", Open Grid Forum, GFD.108, July 2006, http://www.ogf.org/gf/docs/?final