Job Submission Information Model
http://www.isi.edu/~flon/cgs-wg/jsim.doc

Status of This Memo

This memo provides information to the Grid community on a Job Submission Information Model based on DMTF's Common Information Model schema. Distribution is unlimited.

Comments and suggestions on this document are encouraged and should be sent to the CIM Grid Schema work group discussion list:

cgs-wg@gridforum.org

This is a work in progress. The document is version 0.2 dated 5 June 2003 3:35:29 PM.

Copyright Notice

Abstract

This document describes the Job Submission Interface Model. It is based on the "job" schema in DMTF's Common Information Model (CIM), version 2.7 preliminary. It includes a UML diagram of the classes associated with job submission, the managed object format (MOF) for those classes, and an XML representation of the UML.

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED" and "MAY" used in this document are to be interpreted as described in [RFC2119].

Contents

1. Overview

The Job Submission Information Model (JSIM) describes the managed objects and their relationships for managing the execution and monitoring of batch jobs in a grid environment. The CIMv2.7 preliminary schema for jobs and processing is the foundation for the development of this model.

It is our recommendation that grid (batch) scheduling systems act as information providers to a CIMOM (CIM Object Manager), so that any CIM browser can be used to inspect the status of queues and jobs on a grid resource, regardless of the brand of scheduler being utilized. The model defined herein represents our current thinking on the CIM extensions necessary to represent this data.

We also suggest, as a strawman, an XML schema that could be used by OGSI-compliant Grid Services to exchange the information in the model presented.

2. Introduction to the Model

The new elements of the model (on top of base CIM 2.8 classes) are: BatchService, BatchSAP, BatchJob, BatchJobGroup, GroupOfSystems, and JobRecoverySettingData. In addition, attributes have been added to existing CIM classes: JobQueue and QueueStatisticalData.

The basics of JSIM is that a BatchService (there can be one more more hosted in a single CIMOM) represents the root of information for a job scheduling system. A BatchService can host any number of JobQueue's, via the association QueueForBatchService.

The jobs on a JobQueue are represented by a BatchJobGroup, which is a CIM Collection. Each item of the BatchJobGroup is a BatchJob. The BatchJob class inherits from the CIM ConcreteJob and CIM Job, and we have added associations ExecutionTarget and ExecutionCandidate between BatchJob and CIM System.

It is thus possible to locate all of the BatchService's in the CIMOM, all of their JobQueue's, and the BatchJob's on those queues. It is likewise possible, from the point of view of a BatchJob, to find the JobQueue and BatchService it is associated with.

Related aspects of the model include GroupOfSystems, which is a CIM Collection of CIM System's. This facilitates the allocation of a job scheduler to a group of computer systems in a cluster, a group of processors on a single computer, etc.

A BatchJob can reference JobRecoverySettingData through the association JobRecoveryJob. This allows the representation of job recovery and/or restart information, although we do not now suggest what form this information should take.

BatchSAP specifies the batch submission procotols that BatchService uses. The relationship is through the association ServiceAccessBySAP.

The model elements are discussed in a bit more detail following the presentation of the UML, in Section 2.2. The official model in full detail is given in Section 2.3, Descriptions in the Managed Object Format (MOF.

This JSIM does not address nor specifically depend upon any resource model, e.g. cluster computing model. Nor does it address or depend upon any particular software or method for submitting jobs for execution. It contains the standard information and relationships that allow a wide range of grid execution systems to operate efficiently and effectively.

3. UML

The figure below depicts the classes and properties of the batch job schema. The new classes and properties are highlighted in this color (plum).
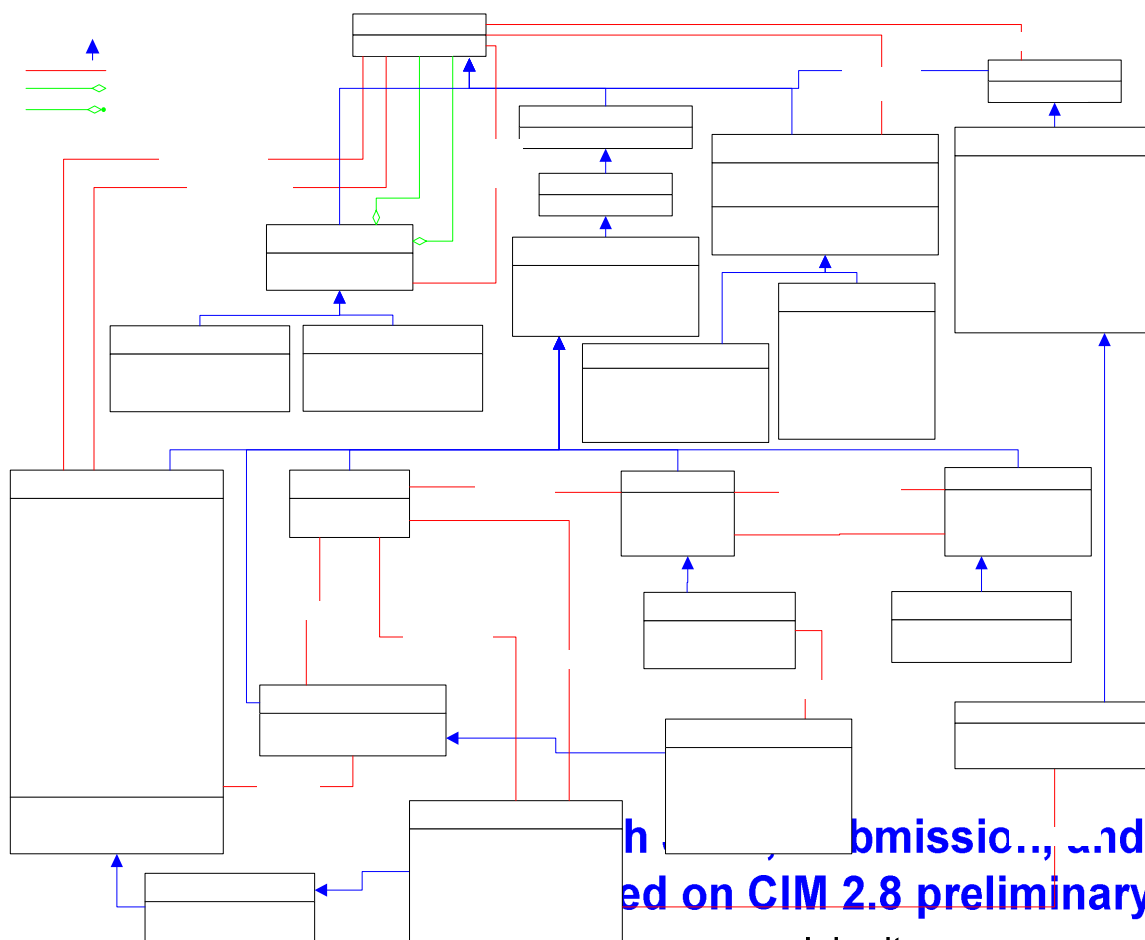
h ..., ...bmissio.., ...nd Processing
ed on CIM 2.8 preliminary)

Inheritance

Association

Aggregation

Composition Aggregation

✱ Equivalent to: 0 .. n

AffectedJobElement

OwningJobElement

4. Discussion of the Model Elements

This section lists the CIM 2.8 classes added and modified by JSIM, as well as newly introduced associations.

4.1. Classes Added

4.1.1.   BatchJob

The BatchJob class inherits from the CIM 2.8 ConcreteJob (and thence Job), with the specific meaning that, while ConcreteJob refers to a job executing on the System, a BatchJob is on a batch queue and traceable in that fashion.

(See Co

Contents:
- JobID (required)
  An arbitrary string (hopefully uniquely) identifying this job.

- MaxCPUTime
  The maximum amount of execution time on CPU this job is allowed.

- CPUTimeUsed
  The amount of execution time on CPU currently consumed by this job.

- BatchJobStatus

- TimeCompleted
  When this job finished.

- JobOrigination

- Task
  The execution command for this job.

4.1.2. BatchJobGroup

A BatchJobGroup inherits from CIM 2.8 Collection. BatchJob's are related through the associations OwningCollectionElement and OrderedMemberOfCollection.

Contents:
- InstanceID
  An arbitrary string uniquely identifying this job.

- GroupStatus:

- GroupStatusDescription

4.1.3. BatchService

A BatchService inherits from CIM 2.8 Service, and especially refers to a job scheduler system.

Contents:
- InstanceID
  An arbitrary string uniquely identifying this service.

4.1.4. BatchSAP

This class inherits from CIM 2.8 ServiceAccessPoint, and specifies the protocols used to interact with the job scheduler service. The protocols are generic in this specification.

Contents:
- BatchProtocol
  A number connoting the specified protocol.

- BatchProtocolInfo
  An arbitrary string containing information specific to this protocol.

4.1.5. JobRecoverySettingData

This class inherits from CIM 2.8 JobSettingData, and adds the following.

Contents:
- RecoverySetting
  A number connoting the specified recovery mechanism in case of job failure. Some choices would be:
  - (0) DoNotContinue, meaning stop the execution of the job stream (either a single job in queue or a job group in the queue) and put it in the 'stuck' state;
  - (1) ContinueWithNextJob, meaning continue with next single job in queue or job in job group in queue recording job's status;
  - (2) RerunJob, meaning re-run the job or job within the job group;
  - (3) RunRecoveryJob, meaning run the job defined to run if original job completes unsuccessfully.

4.1.6. GroupOfSystems

This class inherits from CIM 2.8 Collection, and represents a group of System elements. Certain job schedulers, in particular OpenPBS, use this concept.

Contents:
- InstanceID
  An arbitrary string uniquely identifying this group.

4.2. Classes Modified

4.2.1. JobQueue
The attribute MaxJobCPUTime is added, reflecting the maximum CPU time allowed to each individual job on this queue.

4.2.2. QueueStatisticalData
The attributes RunningJobs and WaitingJobs are added. *It is currently an open issue whether both of these are required, but at least one of them is.*

4.3. New Associations

4.3.1. ExecutionTarget
This identifies the System that a BatchJob is slated to execute on.

4.3.2. ExecutionCandidate
This identifies the BatchJob's that are slated to run on a given System.

4.3.3. QueueForBatchService
This ties one or more JobQueue's to a BatchService.

4.3.4. JobRecoveryJob
This represents the recovery information for a given BatchJob.

5. Managed Object Format (MOF)

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

TBD

6. XML

This section will hold the XML representation for the batch job schema.

7. Security Considerations

This specification defines the model and schema for job submission. While the interactions of job submission 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 schema are bound to specific communication protocols (such as [CIMOPS] or [OGSA]) and programming environments.

Author Information

Ellen Stokes   Lawrence Flon
11400 Burnet Rd   USC/ISI
Austin, TX 78758   4676 Admiralty Way
Phone: +1 512 436 9098   Marina Del Rey, CA 90292
Email: stokese@us.ibm.com   Email: flon@isi.edu

Glossary

<insert glossary items>

References

[RFC2119]
    Bradner, Scott, "Key Words for use in RFCs to Indicate Requirement Levels", RFC 2119, http://www.ietf.org/rfc/rfc2119.txt

[CIM2.7prelim]
    DMTF CIM Schema, Version 2.7 Preliminary, http://www.dmtf.org/standards/cim_schema_v27.php

[CIMspec]
    DMTF, CIM Specification V2.2, http://www.dmtf.org/standards/documents/CIM/DSP0004.pdf

[CIMXML]
    DMTF, Representation of CIM in XML, http://www.dmtf.org/standards/documents/CIM/DSP0201.pdf

[CIMLDAP]
    DMTF, Guidelines for CIM-to-LDAP Directory Mapping, http://www.dmtf.org/standards/documents/DEN/DSP0100.pdf

[CIMOPS]

DMTF, CIM Operations over HTTP,
http://www.dmtf.org/standards/documents/CIM/DSP0200.zip