DRAFT – Job Submission Information Model
Category: Proposed Standard
CIM Grid Schema Workgroup

Ellen Stokes, IBM
Lawrence Flon, USC/ISI
6 June 2003

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 6 June 2003 11:14:12 AM.

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 CIMv2.8 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 preliminary schema classes) are: BatchService, BatchSAP, BatchJob, BatchJobGroup, GroupOfSystems, and JobRecoverySettingData. In addition, attributes have been added to existing CIM classes[1]: 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 JobQueues, via the association QueueForBatchService.

The jobs on a JobQueue are represented by a Batch Job or 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 ExecutionCandidate and ExecutionTarget between BatchJob and CIM System to show the relationship between a batch job and one or more nodes capable of running the job and actually running the job on one or more nodes, respectively.

It is thus possible to locate all of the BatchService's in the CIMOM, all of their JobQueues, and the BatchJobs 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.

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 in Section 4 following the presentation of the UML in Section 3. The ManagedObject Format (MOF) description is given in Section 5. Section 6 offers a corresponding XML description.

---

[1] We intend to submit these additions during the CIM 2.8 prelim member comment period, currently scheduled for June 12 - July 9. It is unclear at this point which of the new classes can be submitted during the comment period; this dialog needs to be had with DMTF.

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).
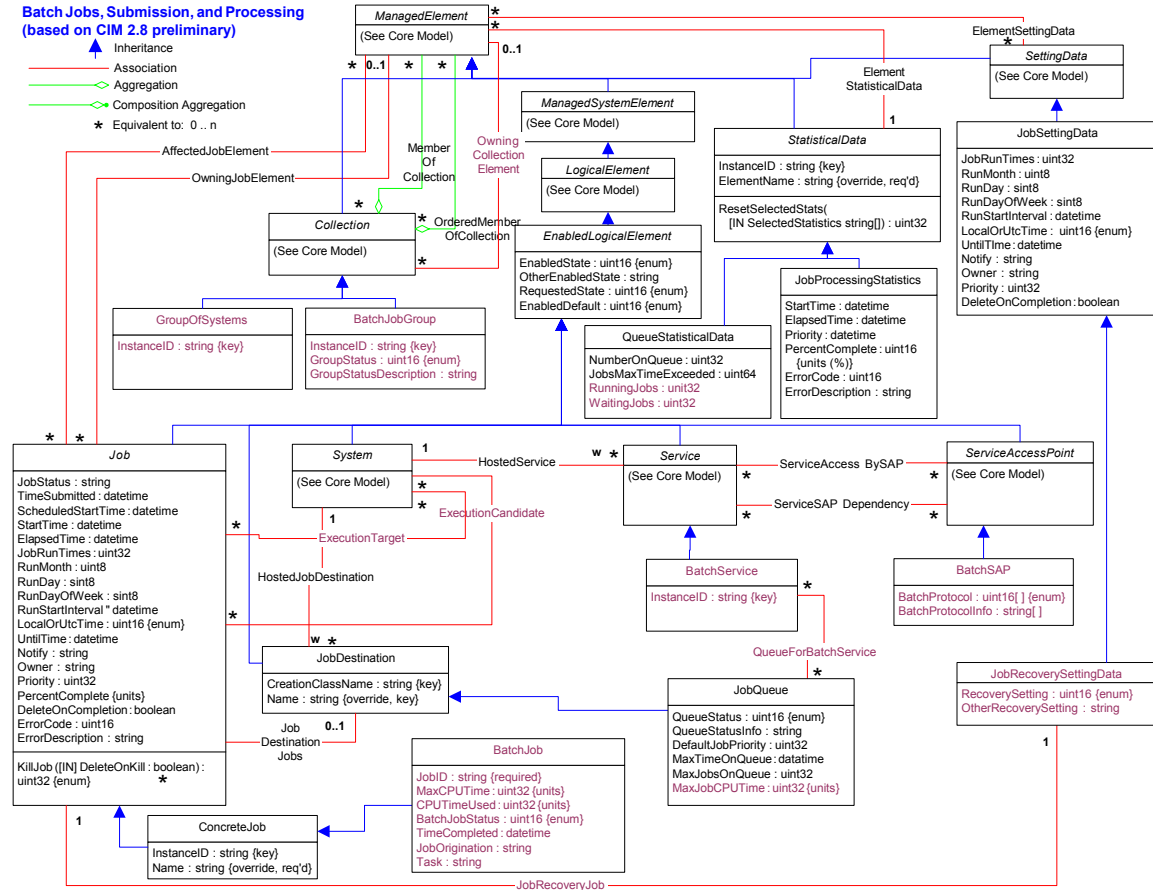
4. Discussion of the Model Elements

This section lists the classes, associations, and properties beyond that provided in CIM 2.8 preliminary.

4.1. Classes Added

4.1.1. BatchJob

The BatchJob class inherits from the CIM_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.

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
  Describes the current state of this BatchJob with respect to the BatchQueue and the system executing this job.

- TimeCompleted
  When this job finished.

- JobOrigination
  Provides additional information, beyond Job Owner inherited from CIM_Job, to identify the origins of the BatchJob. This property could include information such as the System, application or Process that created the BatchJob.

- Task
  The execution command for this job.

4.1.2.   BatchJobGroup

A BatchJobGroup inherits from CIM_Collection. BatchJobs are related through the associations OwningCollectionElement and OrderedMemberOfCollection.

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

- GroupStatus:
  The current status of this BatchJobGroup, based on the statuses of the jobs contained in the Group. The values of GroupStatus are more general than those of the individual jobs

- GroupStatusDescription
  GroupStatusDescription provides additional information regarding the GroupStatus property.

4.1.3.   BatchService

A BatchService inherits from CIM_Service, and especially refers to a job scheduler system.

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

4.1.4. BatchSAP

This class inherits from CIM_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_JobSettingData, and adds the following.

Contents:
- RecoverySetting
  A number connoting the specified recovery mechanism in case of job failure. The choices are:
  - (0) Unknown, meaning the recovery action to take for an unsuccessful job is not defined. No action is taken.
  - (1) Other, meaning a recovery action other than one of the well-defined actions should be taken and is specified in the string property OtherRecoverySetting.
  - (2) 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.
  - (3) ContinueWithNextJob, meaning continue with next single job in queue or job in job group in queue recording job's status.
  - (4) RerunJob, meaning re-run the job or job within the job group.
  - (5) RunRecoveryJob, meaning run the job defined to run if original job completes unsuccessfully.

- OtherRecoverySetting
  A string that specifies the recovery mechanism in case of job failure when the value of RecoverySetting is set to 'Other' (1).

4.1.6. GroupOfSystems

This class inherits from CIM_Collection, and represents a group of System elements. Certain job schedulers, such as 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 is running the Job.

   4.3.2. ExecutionCandidate
     This identifies the System that is capable of running the Job.

   4.3.3. QueueForBatchService
     This ties one or more JobQueues to a BatchService.

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

   4.3.5. OwningCollectionElement
     This identifies the ManagedElement owning the Collection.

5. 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. There may be errors in it, but it is included to provide the details and descriptions necessary to understand the UML.

```
// =====================================================================
// Title:     batch job, based on CIM 2.8 preliminary/feedback
// Filename:   batch job.mof
// Version:
// Release:
// Date:      6 June 2003
// =====================================================================
// Description: This file defines the classes to manage batch job
//          submission.  It builds on the CIM 2.7 preliminary
//          schema (and feedback CRs for CIM 2.7 preliminary)
//          for jobs and processing.
//          *** targeted for CIM 2.8 preliminary ***
// =====================================================================
// Change Log
// Feb 25 2003 - Initial release
// Feb 28 2003 - Added Key property to BatchJobGroup
//      Clarified Descriptions for BatchJob and BatchJob.BatchJobStatus
// Jun 6 2003  - Lots of changes
//
// =====================================================================


#pragma locale ("en_US")


// =====================================================================
// JobQueue
// =====================================================================
```

[Experimental, Version ("2.7.1000"), Description (
    "Status and configuration of a JobQueue. Jobs are held on "
    "a queue, and can move from queue to queue until being "
    "processed.")]
class CIM_JobQueue : CIM_JobDestination {

  [Description (
      "Provides queue-specific status information, beyond the "
      "ManagedSystemElement.OperationalStatus property. This "
      "property further qualifies the status of the queue, "
      "or indicates a secondary condition (e.g. queue full).").
      ValueMap {"0", "1", "2", "3"},
      Values {"Unknown", "Other", "No Additional Status",
          "Queue Full"},
      ModelCorrespondence {"CIM_JobQueue.QueueStatusInfo"}]
  uint16 QueueStatus;

  [Description (
      "Provides a textual explanation for the status of the queue. "
      "Information about both QueueStatus and the OperationalStatus "
      "property (inherited from ManagedSystemElement) may be described "
      "in QueueStatusInfo. Note, however, that OperationalStatus' "
      "Descriptions (if available) MUST be provided in the "
      "StatusDescriptions array, inherited from ManagedSystemElement. "
      "This is where management clients will expect to locate this "
      "information."),
      ModelCorrespondence {"CIM_JobQueue.QueueStatus"}]
  string QueueStatusInfo;

  [Description (
      "Specifies the priority that will be assigned to a new Job "
      "being submitted to this queue, if the Job does not "
      "explicitly specify a priority. Note that lower numbers indicate "
      "higher priorities. The default value of 0xFFFF is the lowest "
      "priority.")]
  uint32 DefaultJobPriority = 65535;

  [Description (
      "Specifies the maximum time (using a datetime interval format) "
      "that a Job can remain on this Queue before completing. If the "
      "MaxTimeOnQueue property is exceeded, then the Job SHOULD be "
      "removed from the Queue. A value of 0 (the default) indicates "
      "that there is no time limit.")]
  datetime MaxTimeOnQueue = "00000000000000.000000:000";

  [Description (
      "Specifies the maximum number of Jobs that can reside on "
      "this Queue. A value of 0 (the default) indicates that "
      "there is no time limit.")]
  uint32 MaxJobsOnQueue = 0;

  [Description (
      "Specifies the maximum number of milliseconds of CPU time "
      "that a job queued to this destination can consume to complete "
      "its execution."),
      Units("MilliSeconds")]

```
    uint32 MaxJobCPUTime;

};




// =====================================================================
// QueueStatisticalData
// =====================================================================
[Experimental, Version ("2.7.1000"), Description (
    "The statistical data of a JobQueue. An instance of this "
    "class is associated with a specific JobQueue using the "
    "inherited ElementStatisticalData relationship.") ]
class CIM_QueueStatisticalData : CIM_StatisticalData {

    [Gauge, Description (
        "The current number of Jobs on the Queue.") ]
    uint32 NumberOnQueue;

    [Counter, Description (
        "The number of Jobs whose MaxTimeOnQueue was exceeded. "
        "This would cause the Job to be removed from the Queue, and "
        "not processed to completion. This count begins at the "
        "time that the Queue was started/initialized.") ]
    uint64 JobsMaxTimeExceeded;

    [Description (
        "Specifies the current number of BatchJobs that "
        "can be in the running state, for the related Queue."),
        Gauge ]
    uint32 RunningJobs;

    [Description (
        "Specifies the current number of BatchJobs that are waiting "
        "to be put in the running state, for the related Queue."),
        Gauge ]
    uint32 WaitingJobs;
};




// =====================================================================
// BatchJob
// =====================================================================
[Experimental, Version ("2.8.0"), Description (
    "Description of a batch request that is either waiting on a "
    "BatchQueue to run, in the process of running, or that has "
    "previously run. Jobs that are completed and not recurring "
    "will NOT be associated with a BatchQueue.  Jobs that are "
    "completed but recurring WILL be associated with a BatchQueue, "
    "since they are waiting to run given the scheduling "
    "information in the associated JobScheduleSettingData instance.") ]
class CIM_BatchJob : CIM_ConcreteJob {

    [Required, Description (
        "Uniquely identifies this Job within a scoping BatchQueue, "
        "and BatchService. This property can be used in the "
```

```
        "construction of the InstanceID key.") ]
   string JobID;

   [Description (
        "Specifies the maximum number of milliseconds of CPU "
        "time that this job can use."),
        Units("MilliSeconds") ]
   uint32 MaxCPUTime;

   [Description (
        "Specifies the number of milliseconds of CPU that this "
        "job has used.  This number will continue to change until "
        "the job has finished its execution, either successfully "
        "or unsuccessfully."),
        Units("MilliSeconds") ]
   uint32 CPUTimeUsed;

   [Description (
        "Describes the current state of this BatchJob with respect "
        "to the BatchQueue and the system executing this job. "
        "Additional information may be specified in JobStatus."),
        ValueMap {"0", "1", "2", "3", "4", "5", "6"},
        Values {"Unknown", "Other", "Pending", "Blocked",
           "Complete", "Completed With Error", "Running"},
        ModelCorrespondence {"CIM_BatchJob.TimeCompleted",
           "CIM_BatchJob.JobStatus"} ]
   uint16 BatchJobStatus;

   [Description (
        "Time when this BatchJob was completed. This value is only "
        "valid if the BatchJobStatus has been set to \"Complete\" "
        "(value=5) or \"Completed With Error\" (value=6)."),
        ModelCorrespondence {"CIM_BatchJob.BatchJobStatus"} ]
   datetime TimeCompleted;

   [Description (
        "Provides additional information, beyond Job Owner "
        "inherited from CIM_Job, to identify the origins of the "
        "BatchJob. This property could include information such as "
        "the System, application or Process that created the "
        "BatchJob.") ]
   string JobOrigination;

   [Description (
           "Provides the command and parameters, in string form, for "
           "the execution of this job.") ]
   string Task;
};


// ======================================================================
// BatchJobGroup
// ======================================================================
[Experimental, Version ("2.8.0"), Description (
   "BatchJobGroup describes a collection of BatchJobs that are "
   "logically grouped.  Two grouping/scheduling concepts are "
```

"modeled: (1) User grouping: Jobs are logically grouped by "
"the user of the batch system to capture common "
"characteristics. There is no sequencing relationship "
"between the jobs belonging to the Group. The jobs are grouped "
"using the MemberOfCollection association. (2) Job "
"sequencing: An ordered collection of jobs is defined "
"where the jobs are executed in sequence. This captures a simple "
"sequencing relationship for a set of jobs and is defined "
"using the OrderedMemberOfCollection association. An individual "
"BacthJob can belong to multiple BatchJobGroups.")]
class CIM_BatchJobGroup : CIM_Collection {

    [Key, Description (
        "Within the scope of the instantiating Namespace, InstanceID "
        "opaquely and uniquely identifies an instance of this class. "
        "In order to ensure uniqueness within the NameSpace, the "
        "value of InstanceID SHOULD be constructed using the "
        "following 'preferred' algorithm: \n"
        "   <OrgID>:<LocalID> \n"
        "Where <OrgID> and <LocalID> are separated by a colon ':', "
        "and where <OrgID> MUST include a copyrighted, trademarked "
        "or otherwise unique name that is owned by the business entity "
        "creating/defining the InstanceID, or is a registered ID that "
        "is assigned to the business entity by a recognized global "
        "authority. (This is similar to the <Schema Name>_<Class Name> "
        "structure of Schema class names.) In addition, to ensure "
        "uniqueness <OrgID> MUST NOT contain a colon (':'). When "
        "using this algorithm, the first colon to appear in "
        "InstanceID MUST appear between <OrgID> and <LocalID>. \n"
        "\n"
        "<LocalID> is chosen by the business entity and SHOULD not be "
        "re-used to identify different underlying (real-world) elements. "
        "If the above 'preferred' algorithm is not used, the defining "
        "entity MUST assure that the resultant InstanceID is not "
        "re-used across any InstanceIDs produced by this or other "
        "providers for this instance's NameSpace.")]
  string InstanceID;

    [Description (
        "Describes the current status of this BatchJobGroup, based "
        "on the statuses of the jobs contained in the Group. The "
        "values of GroupStatus are more general than those of the "
        "individual jobs. The possible values are: \n"
        " 0 = Unknown \n"
        " 1 = Other \n"
        " 2 = All jobs pending \n"
        " 3 = Jobs in mixed states \n"
        " 4 = All jobs completed \n"
        "Additional information may be specified in the Group"
        "StatusDescription property."),
        ValueMap {"0", "1", "2", "3", "4"},
        Values {"Unknown", "Other", "All Jobs Pending",
            "Jobs in Mixed States", "All Jobs Completed"},
        ModelCorrespondence {"CIM_BatchJob.BatchJobStatus",
            "CIM_BatchJobGroup.GroupStatusDescription"}]
  uint16 GroupStatus;

```
    [Description (
        "GroupStatusDescription provides additional information "
        "regarding the GroupStatus property."),
        ModelCorrespondence {"CIM_BatchJobGroup.GroupStatus"} ]
    string GroupStatusDescription;
};



// ====================================================================
// GroupOfSystems
// ====================================================================
[Experimental, Version ("2.8.0"), Description (
    "GroupOfSystems is a collection of resources, for example, a "
    "pool of hosts available for executing jobs.") ]
class CIM_GroupOfSystems : CIM_Collection {

    [Key, Description (
        "Within the scope of the instantiating Namespace, InstanceID "
        "opaquely and uniquely identifies an instance of this class. "
        "In order to ensure uniqueness within the NameSpace, the "
        "value of InstanceID SHOULD be constructed using the "
        "following 'preferred' algorithm: \n"
        "   <OrgID>:<LocalID> \n"
        "Where <OrgID> and <LocalID> are separated by a colon ':', "
        "and where <OrgID> MUST include a copyrighted, trademarked "
        "or otherwise unique name that is owned by the business entity "
        "creating/defining the InstanceID, or is a registered ID that "
        "is assigned to the business entity by a recognized global "
        "authority. (This is similar to the <Schema Name>_<Class Name> "
        "structure of Schema class names.) In addition, to ensure "
        "uniqueness <OrgID> MUST NOT contain a colon (':'). When "
        "using this algorithm, the first colon to appear in "
        "InstanceID MUST appear between <OrgID> and <LocalID>. \n"
        "\n"
        "<LocalID> is chosen by the business entity and SHOULD not be "
        "re-used to identify different underlying (real-world) elements. "
        "If the above 'preferred' algorithm is not used, the defining "
        "entity MUST assure that the resultant InstanceID is not "
        "re-used across any InstanceIDs produced by this or other "
        "providers for this instance's NameSpace.") ]
    string InstanceID;
};



// ====================================================================
// BatchSAP
// ====================================================================
[Experimental, Version ("2.8.0"), Description (
    "The ServiceAccessPoint for accessing a BatchService. The "
    "relationship between the AccessPoint and the Service is "
    "described by instantiating the ServiceAccessBySAP association.") ]
class CIM_BatchSAP : CIM_ServiceAccessPoint {

    [Description (
        "Specifies the batch submission procotols that this "
```

```
      "AccessPoint uses. Note that each entry of this array is "
      "related to the corresponding entry in the BatchProtocolInfo "
      "array that is located at the same index."),
      ArrayType ("Indexed"),
      ValueMap {"0", "1", "2"},
      Values {"Unknown", "Other", "Local"},
      ModelCorrespondence {"CIM_BatchSAP.BatchProtocolInfo"} ]
   uint16 BatchProtocol[];

   [Description (
      "Provides clarifying or additional information about the "
      "protocols supported by this AccessPoint. Note, each entry "
      "of this array is related to the corresponding entry in the "
      "BatchProtocol array that is located at the same index."),
      ArrayType ("Indexed"),
      ModelCorrespondence {"CIM_BatchSAP.BatchProtocol"} ]
   string BatchProtocolInfo[];
};


// ===================================================================
// BatchService
// ===================================================================
[Experimental, Version ("2.8.0"), Description (
   "The Service that provides support for processing BatchJob "
   "requests. The basic assumption of this model is that a "
   "BatchService accepts a BatchJob for processing, via its "
   "BatchSAP. The Job is then placed on a BatchQueue (indicated by "
   "the QueueForBatchService association). A System takes "
   "Jobs from Queues (indicated by the ExecutionTarget "
   "association) and processes them.") ]
class CIM_BatchService : CIM_Service {

   [Key, Description (
      "Within the scope of the instantiating Namespace, InstanceID "
      "opaquely and uniquely identifies an instance of this class. "
      "In order to ensure uniqueness within the NameSpace, the "
      "value of InstanceID SHOULD be constructed using the "
      "following 'preferred' algorithm: \n"
      "   <OrgID>:<LocalID> \n"
      "Where <OrgID> and <LocalID> are separated by a colon ':', "
      "and where <OrgID> MUST include a copyrighted, trademarked "
      "or otherwise unique name that is owned by the business entity "
      "creating/defining the InstanceID, or is a registered ID that "
      "is assigned to the business entity by a recognized global "
      "authority. (This is similar to the <Schema Name>_<Class Name> "
      "structure of Schema class names.) In addition, to ensure "
      "uniqueness <OrgID> MUST NOT contain a colon (':'). When "
      "using this algorithm, the first colon to appear in "
      "InstanceID MUST appear between <OrgID> and <LocalID>. \n"
      "\n"
      "<LocalID> is chosen by the business entity and SHOULD not be "
      "re-used to identify different underlying (real-world) elements. "
      "If the above 'preferred' algorithm is not used, the defining "
      "entity MUST assure that the resultant InstanceID is not "
      "re-used across any InstanceIDs produced by this or other "
```

```
      "providers for this instance's NameSpace.") ]
   string InstanceID;

};
```

```
// ====================================================================
// JobRecoverySettingData
// ====================================================================
[Experimental, version ("2.8.0"), Description (
   "The settings defined for recovery of an unsuccessfully "
   "run job.") ]
class CIM_JobRecoverySettingData : JobSettingData {

   [Description (
      "Describes the recovery action to be taken for an "
      "unsuccessfully run job.  The values and setting descriptions "
      "are: "
      "   o = "Unknown", meaning it is unknown as to what recovery "
      "       action to take "
      "   1 = "Other", meaning "
      "   2 = "Do Not Continue", 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 "
      "   3 = "Continue With Next Job", meaning continue with next "
      "       single job in queue or job in job group in queue "
      "       recording job's status "
      "   4 = "Re-run Job", meaning re-run the job or job within the "
      "       job group "
      "   5 = "Run Recovery Job", meaning run the job defined to run "
      "       if original job completes unsuccessfully.   The "
      "       recovery job must have already been placed on the queue "
      "       from which it will run."),
      ValueMap {"0", "1", "2", "3", "4", "5"},
      Values {"Unknown", "Other", "Do Not Continue", "
         "Continue With Next Job", "Re-run Job", "Run Recovery Job"} ]
   uint16 RecoverySetting;

   [Description (
      "A string describing the recovery action to take when the "
      "instance's RecoverySetting property is set to 1 (\"Other\"). " ]
   string OtherRecoverySetting;

};
```

```
// ====================================================================
// QueueForBatchService
// ====================================================================
[Association, Experimental, Version ("2.8.0"), Description (
   "This association indicates that a BatchService utilizes a "
   "particular BatchQueue, to hold jobs submitted to the Service.") ]
class CIM_QueueForBatchService : CIM_Dependency {

   [Override ("Antecedent"), Description (
```

```
      "The BatchQueue that the Service utilizes.") ]
   CIM_BatchQueue REF Antecedent;

   [Override ("Dependent"), Description (
      "The BatchService that puts BatchJobs on the Queue.") ]
   CIM_BatchService REF Dependent;

   [Description (
      "Indicates that the BatchService can place jobs on the Queue.") ]
   boolean QueueAcceptingFromService;
};



// ===================================================================
// ExecutionCandidate
// ===================================================================
[Association, Experimental, Version ("2.8.0"), Description (
   "ExecutionCandidate represents the association between "
   "a System and a Job, describing that the System is "
   "capable of running the job.") ]
class CIM_ExecutionCandidate : CIM_Dependency {

   [Override("Antecedent"), Description (
      "The System that is capable of running the Job.") ]
   CIM_System REF Antecedent;

   [Override("Dependent"), Description (
      "The Job that can be run.") ]
   CIM_Job REF Dependent;
};



// ===================================================================
// ExecutionTarget
// ===================================================================
[Association, Experimental, Version ("2.8.0"), Description (
   "ExecutionTarget represents the association between "
   "a System and a BatchJob, describing that the System is running "
   "the job.") ]
class CIM_ExecutionTarget : CIM_Dependency {

   [Override("Antecedent"), Description (
      "The System that is running the BatchJob.") ]
   CIM_System REF Antecedent;

   [Override("Dependent"), Description (
      "The BatchJob that is run.") ]
   CIM_BatchJob REF Dependent;
};



// ===================================================================
// JobRecoveryJob
// ===================================================================
[Association, Experimental, Version ("2.8.0"), Description (
   "An association describing what recovery action a Job is taken for "
```

```
    "unsuccefully processing, ie to which JobRecoverySettingData.") ]
class CIM_JobDestinationJobs : CIM_Dependency {

    [Override ("Antecedent"), Max (1), Description (
        "The Job recovery setting.") ]
    CIM_JobRecoverySettingData REF Antecedent;

    [Override ("Dependent"),
        Description ("The Job that is in the Job queue/destination.") ]
    CIM_Job REF Dependent;
};



// ====================================================================
// OwningCollectionElement
// ====================================================================
[Association, Experimental, Version ("2.8.0"), Description (
    "OwningCOllectionElement represents an association between a
    "Collection and the ManagedElement responsible for the creation of
    "the Collection.  This association may not be possible, given that "
    "the execution of collections of jobs can move between systems and "
    "that the lifecycle of the creating entity may not persist for the "
    "total duration of the colleciton.  However, this can be very "
    "useful information when available. "} ]
class CIM_OwningCollectionElement {

    [Key, Max(1), Description (
        "The ManagedElement responsible for the creation of the "
        "Collection.") ]
    CIM_ManagedElement REF OwningElement;

    [Key, Description (
        "The Job created by the ManagedElement.") ]
    CIM_Collection REF OwnedElement;
};

// ====================================================================
// end of file
// ====================================================================
```

6. XML

This section will hold the XML representation corresponding to the JSIM.

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: +1 512 838 0552   Marina Del Rey, CA 90292
Email: stokese@us.ibm.com   Email: flon@isi.edu

Glossary

<insert glossary items>

Intellectual Property Statement

Full Copyright Notice

Acknowledgements

References

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

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

[CIM2.8prelim]
    DMTF CIM Schema, Version 2.8 Preliminary,
    http://www.dmtf.org/standards/cim_schema_v28.php [2]

[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

---

[2] To be available to DMTF members for comment on June 12