ETSI TS 102 827 V1.1.1 (2008-08)

Technical Specification

GRID;
Grid Component Model (GCM);
GCM Interoperability Deployment

Reference

DTS/GRID-0004-1

Keywords

interoperability, service, architecture, network

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee GRID (GRID).

The present document is related to document 102-828 (GCM Interoperability Application Description).

Introduction

The purpose of the present document is to offer a standard, uniform way of describing grid resources, so as to make these resources available to grid-based applications.

When building a grid, one is confronted to several problems, among which:

- choosing a grid framework, as many are available.
- when running a grid-enabled application, one must rewrite configuration directives when switching to a different grid framework.
- a grid can be built on widely different hardware infrastructures, from an heterogenous set of desktop machines to a dedicated cluster, or even any combination of those.
- grid infrastructures can also be very dynamic, it should be easy to add or remove machines, and such changes should be transparent to users.

To simplify this, the GCM Interoperability Deployment standard offers a uniform way to describe grid resources, how they are connected to one another, how they can be activated, and how they can be accessed.

A grid infrastructure is modelized with several entities: Host, Bridge, and Group (see definitions clause). By combining these entities, one can describe the physical infrastructure of the grid, i.e. how the machines it includes are configured and networked together.

The GCM has been first defined in the NoE CoreGRID (42 institutions). A reference Open Source implementation has been tested in the 4 previous GRID Plugtests organized from 2004 to 2007 by ETSI, and will also be part of the forthcoming 5th one in 2008. The GridCOMP EU project (FP6, started June 2006 to November 2008) is working to further assess and experiment with the specification.

1 Scope

The present document describes an XML schema for describing resources to be used in distributed and parallel infrastructures like enterprise and scientific grids and job schedulers.

The standard will help enterprises and laboratories to manage large-scale computer and telecom infrastructures with the necessary virtualization.

Its primary audience are grid system administrators who will need to write deployment descriptors for their grid.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
    - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
    - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1] ETSI TS 102 828: "GRID; Grid Component Model (GCM); GCM Application Description".

2.2 Informative references

The following referenced documents are not essential to the use of the present document but they assist the user with regard to a particular subject area**.** For non-specific references, the latest version of the referenced document (including any amendments) applies.

[i.1] ARC / Nordugrid: "middleware for grid computing".

NOTE: Available at http://www.nordugrid.org/.

[i.2] CGSP / China Grid: "middleware for ChinaGrid".

NOTE: Available at http://www.cngrid.org

[i.3] EGEE gLite: "middle-ware for grid computing".

NOTE: Available at http://glite.web.cern.ch/glite/ .

[i.4] Fura / Inner Grid: "middleware for grid computing".

NOTE: Available at http://fura.sourceforge.net/.

[i.5] Globus: "open source software toolkit to build Grid systems and applications".

NOTE: Available at http://www.globus.org/.

[i.6] GridBus: "middleware for grid computing".

NOTE: Available at http://www.gridbus.org/middleware/.

[i.7] IBM Load Leveler: "job scheduler".

NOTE: Available at http://www.ibm.com/systems/clusters/software/loadleveler/, http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.loadl.doc/loadl331/am2ug30323.html#jobkey.

[i.8] LSF: "job scheduler".

NOTE: Available at http://www.platform.com/products/ LSF /.

[i.9] OAR: "job scheduler".

NOTE: Available at http://oar.imag.fr/docs/manual.html.

[i.10] PBS / Torque: "job scheduler".

NOTE: Available at http://www.clusterresources.com/pages/products/torque-resource-manager.php.

[i.11] PBS / PRUN: "run server from an old version of PBS".

NOTE: Available at http://www.cs.vu.nl/das/prun/prun.1.html.

[i.12] Sun GridEngine: "grid computing framework".

NOTE: Available at http://gridengine.sunsource.net/.

[i.13] CCS: "Microsoft Compute Cluster Server".

NOTE: Available at http://www.microsoft.com/windowsserver2003/ccs/default.aspx.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

bridge: GCM Deployment concept that represents a machine through which multiple machines are accessed

NOTE: A typical example is a cluster frontend (see frontend).

certificate: electronic document which incorporates a digital signature to bind together a public key with an identity - information such as the name of a person or an organization, their address, and so forth

certificate authority: entity which issues digital certificates for use by other parties

cluster: set of loosely-coupled machines working together so that they are seen as a single computer

Core (CPU Core): processing unit within a processor chip, effectively seen as a CPU from the Operating System

NOTE: For instance, a dual-core CPU will be seen as two CPUs by the OS.

deployment descriptor: XML file following the schema specified in the present document, which describes what grid resources are to be deployed and how

frontend: in a physical network, a single machine acting as a gateway for a set of machines

NOTE: This is often the case for machine clusters, where individual machines are not accessible directly.

Grid Component Model (GCM): component architecture and framework for the portability, and reusability of Grid applications, the integration with SOA infrastructures, and providing interoperability of GRID systems

group: GCM Deployment concept that represents a set of machines (i.e. hosts, see definition) and the grid protocol through which they can be accessed

NOTE: These machines may all be sharing the same configuration (OS type, installation path of executable etc.), but it may also only have the same grid protocol while having very different configurations. A group is generally used to represent a cluster.

host: GCM Deployment concept which represents a single physical machine on a network

NOTE: A GCM Deployment model can be seen as a tree, and hosts are the leaves.

Host Capacity (HC): number of programs or VM per Host

NOTE: An indicator of the number of programs to be executed in parallel on a host.

infrastructure: organized set of processing resources which form a grid

job scheduler: program which purpose is to manage a queue of program execution requests (jobs) in order to run them in turn, or at a specified time

node: GCM Deployment concept that represents a location to execute a computation

NOTE: At execution, a node is a place where a program or an activity can be executed. This can be a single physical machine (a host), or, in the case of a multi-processor / multi-core machine, a single processor or a single core within a machine.

resource: processing unit which is a building part of the grid

NOTE: A resource can either be a **host** (a single physical machine), a **bridge** (a frontend machine providing access to several others), or a **group** (a set of identical machines).

Service Oriented Architecture (SOA): computer system's architectural style for creating and using business processes, packaged as services, throughout their lifecycle

Schema (XML Schema): set of structure constraints for an XML document

NOTE: They provide a means for defining the structure, content and semantics of XML documents. Schemas are a W3C recommendation. See http://www.w3.org/XML/Schema**.**

Virtual Machine (VM): software implementation of a computer that executes programs like a real machine

VM Capacity (VMC): number of activities or node- per VM - an indicator of the number of activities (threads) to be executed in parallel within a program

NOTE: In a standard approach, the *Host Capacity x VM Capacity* should be set to the number of CPU cores of the machine. However, these indicators are by no means absolute. They can be tuned to take into account the specifications of the platform (memory, network, etc.), or of the tasks to be executed in order to obtain the desired load.

Virtual Node (VN): set of nodes -A Virtual Node is an abstraction for deploying parallel and distributed applications. A VN, after a GCM deployment, refers to a set of Nodes, which is seen as a single entity

NOTE: Without JVMs, at execution, a VN maps to a set of Hosts, each Host containing a set of Nodes. With JVMs, at execution, a VN maps to a set of Hosts, each Host containing a set of JVMs, each JVM containing a set of Nodes.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CA	Certification Authority
CPU	Central Processing Unit
GCM	Grid Component Model
HC	Host Capacity
JVM	Java Virtual Machine
OS	Operating System
RSH	Remote Shell
SOA	Service Oriented Architecture
SSH	Secure Shell
VM	Virtual Machine
VMC	Virtual Machine Capacity
VN	Virtual Node
XML	eXtensible Markup Language

4 Overall resource specification

The purpose of the following clauses is to explain the "model toolkit" that the GCM Deployment offers to describe a grid. As explained in the introduction, a grid can be almost any kind of machines networked together. We detail here how to represent any given setup through the GCM Deployment standard.

4.1 Principles

A deployment descriptor is written in XML. The XML document must conform to the GCM Deployment schema (specified in the present document).

A deployment descriptor is meant to be referenced by an application descriptor, which is the description of the application that will be run, and its resource requirements (GCM Application Descriptor, described in the companion document TS 102 828 [1]).

The benefit of the deployment/application split is that the infrastructure part (which is described by the present document) is reuseable for any other deployment on the same grid. It can also be used by schedulers and meta-schedulers.

A grid infrastructure is modelized through three kinds of resources: hosts (single machines), groups (a set of machines) and bridges (a single machine acting as a gateway for groups or hosts).

The model is defined in two parts:

1) **Infrastructure:** this is a flat list of each individual element of the grid: hosts, groups and bridges listed in no particular order.

2) **Resources:** this is a tree describing the hierarchical relationships between these infrastructure elements. These relationships are defined by:

   - a host being within which group;

   - a group being behind a bridge;

   - a host being directly available.

Considering a very simple grid, that is two desktop PCs networked together. Such a setup would be represented as follows (configuration parameters are omitted for the sake of clarity):

```
<resources>
    <hosts>
                <host refid="host1"  />
                  <host refid="host2"  />
        </hosts>
    </resources>

<infrastructure>
      <hosts>
                <host id="host1"  />
                  <host id="host2"  />
        </hosts>
</infrastructure>
```

There is no hierarchical relation between the two hosts, so both resources and infrastructure parts are identical (aside of the extra configuration parameters which are omitted here).

A slightly more complex example would be a cluster of 12 mono-processor machines running LSF. The representation in GCM Deployment would be as follows:

```
<resources>
        <group refid="LSF_GROUP">
            <host refid="LSF_GROUP_MEMBER" />
        </group>
    </resources>

    <infrastructure>
        <hosts>
            <host id="LSF_GROUP_MEMBER" />
        </hosts>

        <groups>
            <lsfGroup id="LSF_GROUP" >
                <resources processorNumber="12" />
            </lsfGroup>
        </groups>
    </infrastructure>
```

Within the <infrastructure>, the <hosts> part describes the configuration common to the machines in the group. The <groups> part describes the LSF group itself. Finally, the <resources> part describes how they fit together, in this case the host model being within the LSF group.

The following clauses give the complete reference on how a grid can be described through the GCM deployment.

4.2 Overall structure of a descriptor

A deployment descriptor has the following XML structure:

```
<environment>
        <descriptorVariable …/>
…
</environment>

<resources>
        <bridge …/>
        <group>
            <host …/>
            <host …/>
            …
        </group>
```

```
    …
</resources>

<acquisition>
        <lookup …/>
        <p2p …/>
    …
</acquisition>

<infrastructure>
        <hosts>
            <host…/>
        </hosts>

        <groups>
            <groupType …/>
        </group>

        <bridges>
            <bridgeType …/>
        </bridges>
    …
</infrastructure>
```

The elements must be specified in this order. The `<environment>` and `<acquisition>` elements can be omitted, while the `<resources>` and `<infrastructure>` ones are mandatory. The next paragraphs go more in depth on the content and usage of each element.

4.2.1 Environment element

To allow for a bit of flexibility, it is possible to define variables in a descriptor. The variables can be used in any XML value element. They cannot be used in an XML element name. The `<environment>` element is where the variables are defined. It is a simple list of `<descriptorVariable>` elements. There is no limitation on the number of elements. The element has only two attributes, "name" and "value", each accepting string values. The element cannot have child elements.

EXAMPLE:

```
<environment>
    <descriptorVariable name="usertype" value="admin" />
  <descriptorVariable name="username" value="jsmith" />
</environment>
```

This allows the following usage later on in the decriptor:

```
<sshGroup user="${username}" />
```

4.2.2 Resources element

The `<resource>` element describes the hierarchical structure of the available grid resources. This can be seen as the topology of the grid: which hosts are part of a group, which group is behind a bridge, etc. All the grid resources which are listed in it are fully defined in the `<infrastructure>` element. However it does not have to hold every element listed in `<infrastructure>`, it is meant to contain only the subset of resources which are actually used by the deployment. This is to give further flexibility in the descriptor, in that the infrastructure part can be kept while the resource part is changed to suit the current deployment needs. See clause 4 for some examples of the <resource> tag.

There are 3 possible elements within the `<resource>` element:

1) <host>

2) <group>

3) <bridge>

These elements all take a single argument named 'refid'. The value of the argument is the id of the corresponding host/bridge/group element defined in the <infrastructure> element. The topology must be described according to the following rules:

- A host can be at the top level, or in a group element.
- A group can be at the top level, or in a bridge element.
- A bridge can only be at the top level.

For example, the following constructions are correct:

Single host:

```
<host refid="A_HOST" />
```

Group:

```
<group refid="CLUSTER">
    <host refid="CLUSTER_NODE" />
</group>
```

Group behind a bridge:

```
<bridge refid="CLUSTER_FRONT_END" />
    <group refid="CLUSTER">
        < host refid="CLUSTER_NODE" />
    </group>
</bridge>
```

4.2.3 Acquisition element and desktop grid

The `<acquisition>` element describes how resources which are already running can be acquired. It can be used or extended towards standards for managing resources from desktop grids. It contains a simple sequence of two types of elements: `<lookup>` and `<p2p>`, in this order. Each element can either have a single occurrence or be omitted.

The `<lookup>` element has the following three attributes:

1) type: one of "RMI", "HTTP", "IBIS";

2) hostlist: a HostList as defined in clause 5.1.1;

3) port: a positive integer.

It cannot have any child elements.

`The <p2p>` element has a single attribute named "nodesAsked", which holds a positive integer value, or the value 'MAX', indicating the number of requested nodes. The value 'MAX' means that the maximum number of available nodes will be allocated to the task. If more nodes are made available during the course of the task's execution, they will be added as well.

The `<p2p>` element can have two children. The first one is `<localClient>`, which defines the parameters for the local P2P client. These parameters are represented by two attributes:

- protocol (enumeration), which can be one of 'RMI', 'HTTP', or 'IBIS';
- port (positive integer), which is the client's IP port number.

The second child is `<peerSet>`, which is a simple sequence of `<peer>` elements. Each peer element is a string defining the url toward an access point to the p2p network.

Here is an example of an `<acquisition>` tag featuring all the tags described:

```
<acquisition>
    <lookup type="RMI" port="6666" hostList="host[0-9].grid.fr"></lookup>
    <p2p nodesAsked="50">
        <localClient protocol="RMI" port="2410" />
        <peerSet>
            <peer>rmi://schubby.inria.fr</peer>
            <peer>http://gaudi.inria.fr</peer>
        </peerSet>
    </p2p>
</acquisition>
```

4.2.4 Infrastructure element

The `<infrastructure>` tag holds the list of grid resources on which the deployment can take place, in no particular order. Its purpose is to describe how these resources are deployed (i.e. through which protocols). It can have a single child element of each of the following types: `<hosts>`, `<bridges>`, `<groups>`. `<bridges>` and `<groups>` may be empty or omitted, but there should be at least one child element in `<hosts>`. The child elements which can be stored in these types are described in clause 5.2.

5 Infrastructure specification

5.1 Common types

The following data types are used by many different elements defining the various grid and job scheduler protocols.

5.1.1 Host list

The format of a host list is a whitespace-separated list of name patterns or hostnames. A name pattern describes a set of hostnames with a common root. The format is as follows:

<root name><*interval*>

with *root name* being an alphanumeric string (only letters and digits, no spaces or punctuation signs), and *interval* defining a set of numerical values in the form of an interval or list of values, possibly followed by an exclusion interval or list of values. The general form of an interval is:

[<value set>]^[<value set>]

or simply:

[<value set>]

if no exclusion interval is needed.

A *value set* is a coma-separated list of integers or integers pairs separated by a dash, meaning an interval of values. The values of an interval must be specified in increasing order, and the generated values will be in increasing order. Also, the first integer of an interval can have leading zeroes to indicate the number of digits (numbers will be padded with zeroes if needed).

Some examples:

- host[0-5]: host0, host1 … host5;
- host[0-5]^ [i.3]: host0, host1, host2, host3, host5;

- host[0-10]^[4-6]: host0, host1, host2, host3, host7, host8, host9, host10;
- host[00-5]: host00, host01, host02… host05;
- host[1, 004-7, 09]: host1, host004, host005, host006, host007, host09.

Here is the BNF grammar of a host list:

```
hostList ::= <host> { <host> }

host ::= <prefix><interval> | <interval><suffix> | <prefix><interval><suffix> | <hostname>

prefix ::= letter { letter | digit }
suffix ::= letter { letter | digit }
hostname ::= letter { letter | digit }

interval ::= "["<valueset>"]" | "["<valueset>"]" "^" "["<valueset>"]"

valueset ::= integer_or_interval { "," integer_or_interval }

integer_or_interval ::= integer | <interval>

interval ::= integer "-" integer
```

5.1.2 ID

An ID is an string of alphanumeric characters (no spaces nor punctuation signs). Each ID must be unique in the descriptor.

5.1.3 Path element and path string

A path element has two attributes: 'base' and 'relpath'. 'base' defines what the path is relative to. The following values are defined:

- **root:** the 'relpath' will be interpreted as an absolute path;
- **home:** the home directory of the user under which the job is run.

A path string is simply a text element which value represents a path in the format of the platform the path will be used on.

5.1.4 Argument list

An argument list is a simple list of `<arg>` elements, each holding a string value.

5.2 Elements of the infrastructure

This clause presents the concepts within the `<infrastructure>` tag, which describes the actual deployment of resources on the grid infrastructure. Deployment is the process through which a machine becomes part of the grid (this generally involves starting the appropriate program on the machine). There are three types of deployment:

1) Host: deployment of a single machine which is reached directly, with no intermediate.

2) Bridge: deployment of one or several machines through a single gateway machine (the bridge).

3) Group: deployment of several identically configured machines.

5.2.1 Host description

The HostInfo data structure describes a single machine and the environment it provides, with the following information:

- **userName:** (string) the name of the user under which this host can be accessed;
- **homeDirectory:** (absolute path) the home directory of the user;

- **os:** (one of "unix" or "windows") the operating system the host is running;
- **hostCapacity:** (positive integer) the number of processes (VM or other executable) that this host can handle (default value is 1);
- **vmCapacity:** (positive integer) the number of nodes a single VM on this host can handle (default value is 1);
- **id:** (ID) an ID identifying the host.

HostCapacity and vmCapacity are both optional, however one cannot be specified without the other. Either they are both specified or not at all.

The purpose of the HostInfo structure is to abstract an application from its environment, and therefore to make its adaptation to other environments easier.

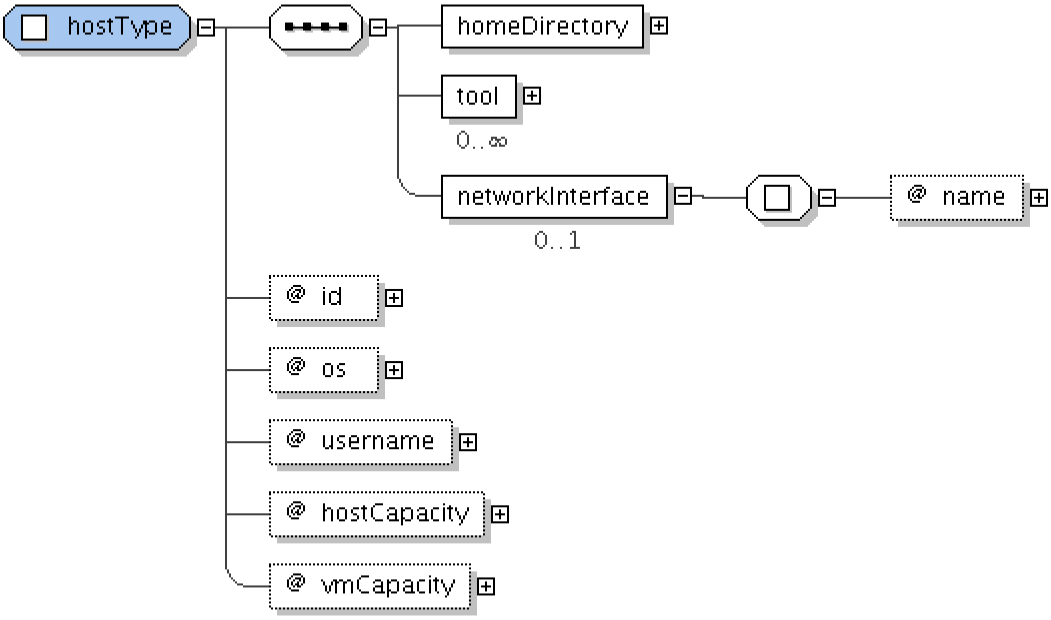


Figure 1

5.2.2 Bridge description

A bridge is meant to represent a frontend to a computing resource. Many grid architectures have such a feature: each physical machine is not accessible directly, the user must instead go through a single machine called a front-end. In a deployment descriptor, a bridge is a gateway toward either:

- a host;
- a set of groups;
- another bridge.

A bridge is defined as a base structure meant to be derived. The base structure only defines an id (string).

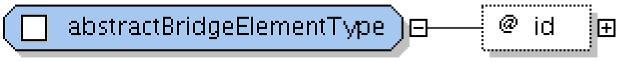


Figure 2

In addition, the following bridge types are defined, as they are the most widely used: rsh (6.1), ssh (6.2) and oarsh (6.3).

5.2.3 Group description

A Group is a data structure defining a set of machines with identical configuration (like a cluster). It is meant as a base structure which can be derived in an Object-Oriented manner to implement any kind of group. There currently are two kinds of groups:

1) "direct" groups;

2) job schedulers.

It is therefore possible to define a standard-compliant deployment descriptor even on a grid which has its own job scheduler.

All group protocols have the following child elements:

- **environment** (environment)**:** the environment for the command.

and the following attributes:

- **id** (ID)**:** the id of the group this element represents;
- **commandPath** (path string)**:** path of the command which is used to submit a job to the group protocol.

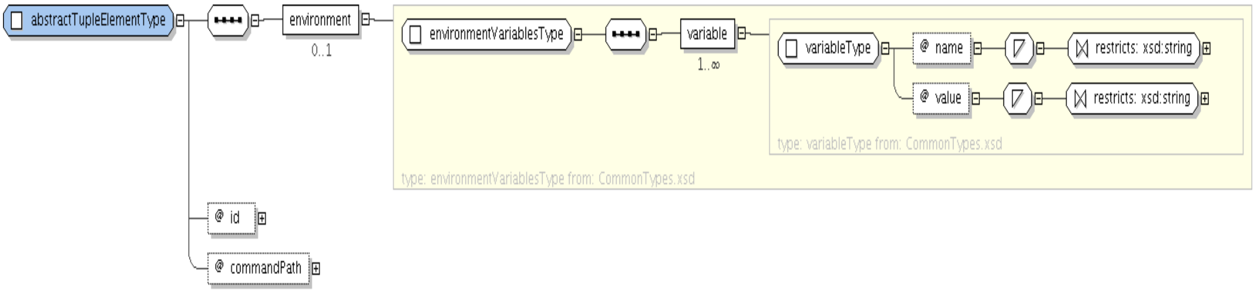


Figure 3

Direct group protocols have a supplemental attribute:

- **username** (string)**:** the username under which the job will be run.

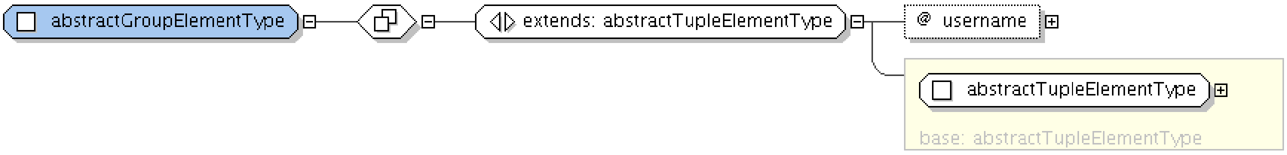

abstractGroupElementType
extends: abstractTupleElementType
@ username
abstractTupleElementType
base: abstractTupleElementType

Figure 4

Scheduler groups have the following child element:

- **scriptPath** (path element)**:** the path to the job script which will be run.

and the following supplemental attribute:

- **bookedNodeAccess** ('sh', 'ssh', 'oarsh')**:** access method of the started nodes (the value must be set according to the group protocol the element describes).

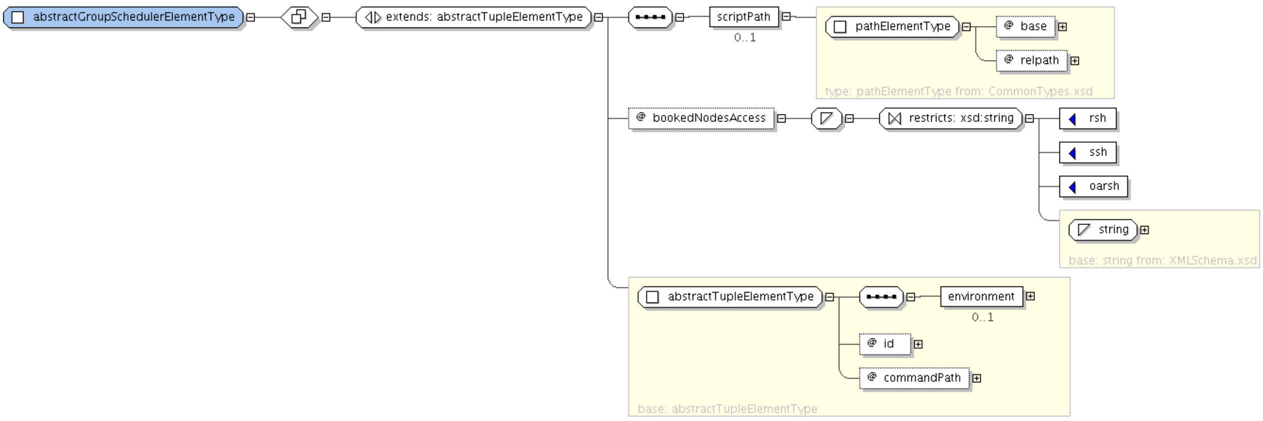

abstractGroupSchedulerElementType
extends: abstractTupleElementType
scriptPath
0..1
pathElementType
base
relpath
type: pathElementType from: CommonTypes.xsd
bookedNodesAccess
restricts: xsd:string
rsh
ssh
oarsh
string
base: string from: XMLSchema.xsd
abstractTupleElementType
environment
0..1
id
commandPath
base: abstractTupleElementType

Figure 5

6 Instances of bridge protocols (one to one)

A bridge starts from one machine to go to a single other machine. It can be used to contact a resource through an SSH gateway for example.

6.1 rsh

An RSH bridge element can have the following attributes:

- **id** (string)**:** the id of the bridge connector corresponding to this definition;
- **hostname** (string)**:** the network hostname of the physical machine which acts as the bridge;
- **username** (string, optional)**:** the user name under which the machine can be accessed;
- **commandPath** (string, optional)**:** the path of rsh client to use.

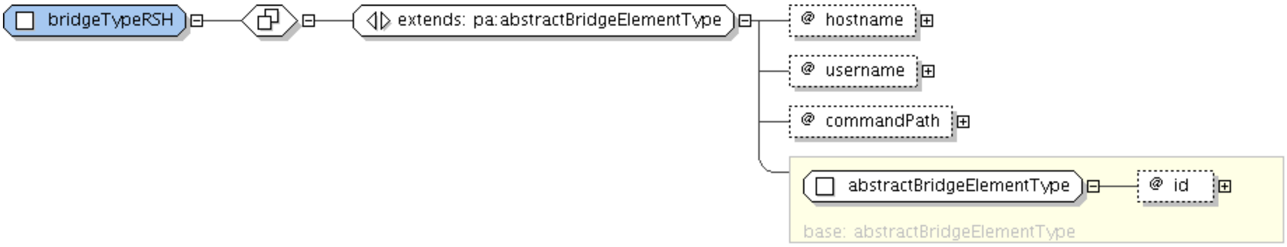

bridgeTypeRSH
extends: pa:abstractBridgeElementType
hostname
username
commandPath
abstractBridgeElementType
id
base: abstractBridgeElementType

Figure 6

6.2 ssh

An SSH bridge element can have the following child element:

- **privateKey** (path string)**:** the file of the private ssh key needed to access the bridge.

An SSH bridge element can have the following attributes:

- **id** (string)**:** the id of the bridge connector corresponding to this definition;
- **hostname** (string)**:** the network hostname of the physical machine which acts as the bridge;
- **username** (string, optional)**:** the user name under which the machine can be accessed;
- **commandPath** (string)**:** the path of the ssh client to use;

- **commandOptions** (string)**:** options to pass to the ssh command.

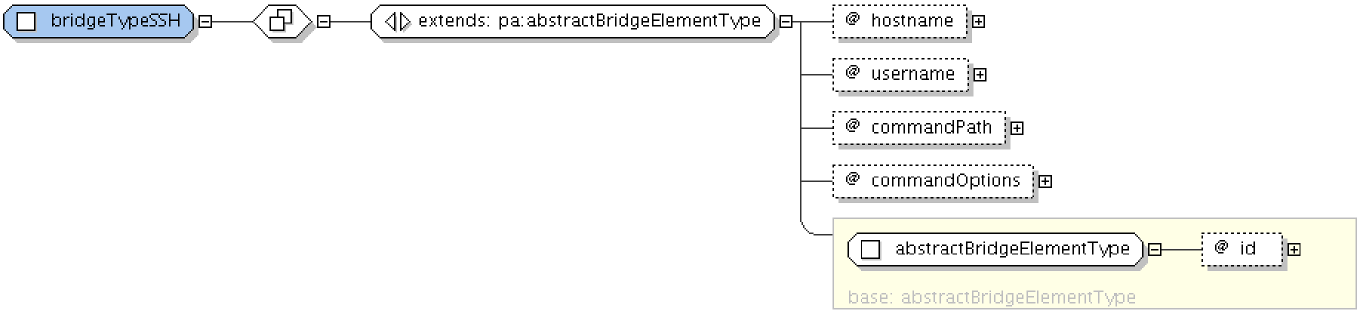

bridgeTypeSSH
extends: pa:abstractBridgeElementType
hostname
username
commandPath
commandOptions
abstractBridgeElementType
id
base: abstractBridgeElementType

Figure 7

6.3 oarsh

An OARSH bridge element has the same parameters as an SSH bridge element, plus a '**jobId**' (positive integer) mandatory attribute, which holds the OAR ID of the job.

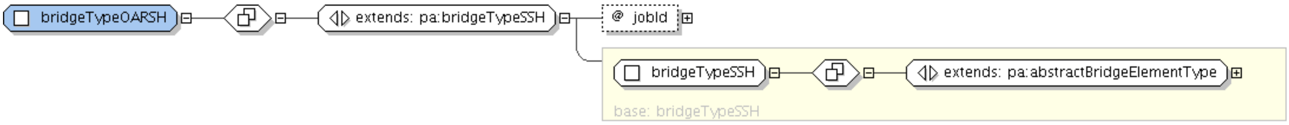

bridgeTypeOARSH
extends: pa:bridgeTypeSSH
jobId
bridgeTypeSSH
extends: pa:abstractBridgeElementType
base: bridgeTypeSSH

Figure 8

6.4 gsissh

A gsissh bridge element has the same parameters as an ssh bridge element, plus two optional attributes: a '**port**' (positive integer) which holds the port used by the gsissh server and '**certificate**' that indicate the path of a valid p12 certificate delivered by a Certification Authority (CA). By default, these values follow standard Globus configuration, the port being retrieved on the $GLOBUS_LOCATION/etc/ssh/ssh_config file and the certificate on $HOME/.globus/usercred.p12.

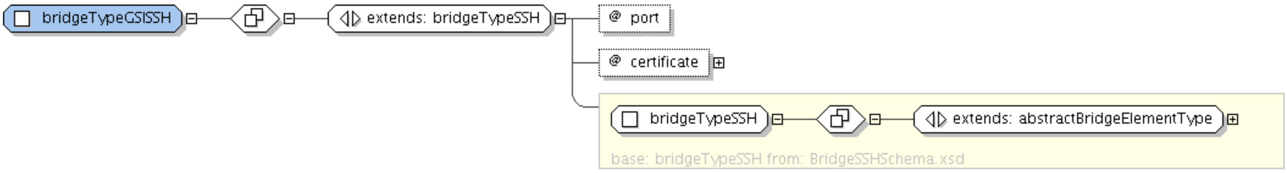

bridgeTypeGSISSH
extends: bridgeTypeSSH
port
certificate
bridgeTypeSSH
extends: abstractBridgeElementType
base: bridgeTypeSSH from: BridgeSSHSchema.xsd

Figure 9

7 Instances of group protocols (One to N)

In the following clauses, elements and attribute names which are underlined mean that the item is mandatory.

7.1 Direct groups

The following groups are simply sets of machines with a common configuration. To make the distinction with job schedulers, they are named "direct groups", since the deployment process directly accesses the machines in the group.

7.1.1 GroupSSH

The SSH Group has the following child elements:

- **hostList** (host list)**:** the list of hosts to connect to.
- **privateKey** (path string)**:** the file of the private ssh key needed to access the host.
- **commandOptions** (string)**:** the list of options which will be passed to the ssh command.

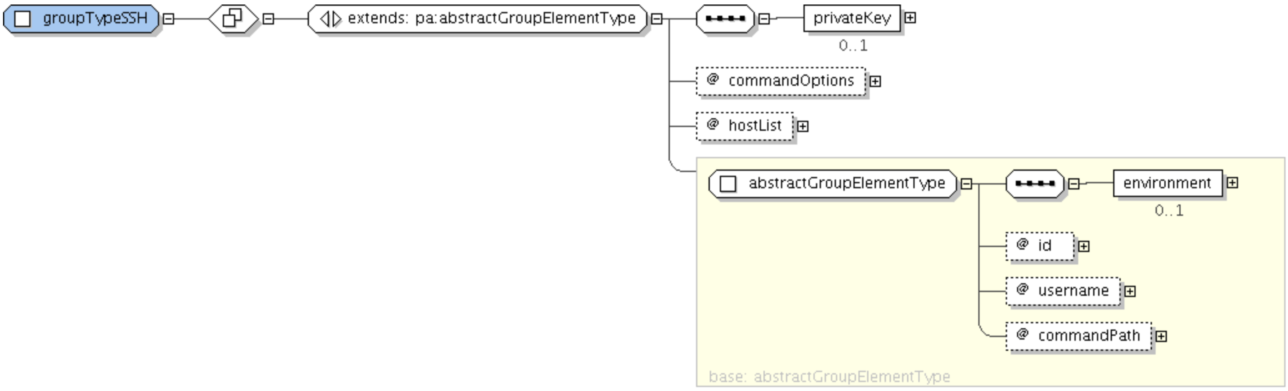


Figure 10

7.1.2 GroupRSH

The rsh Group has the following child elements:

- **hostList** (host list)**:** the list of hosts to connect to.

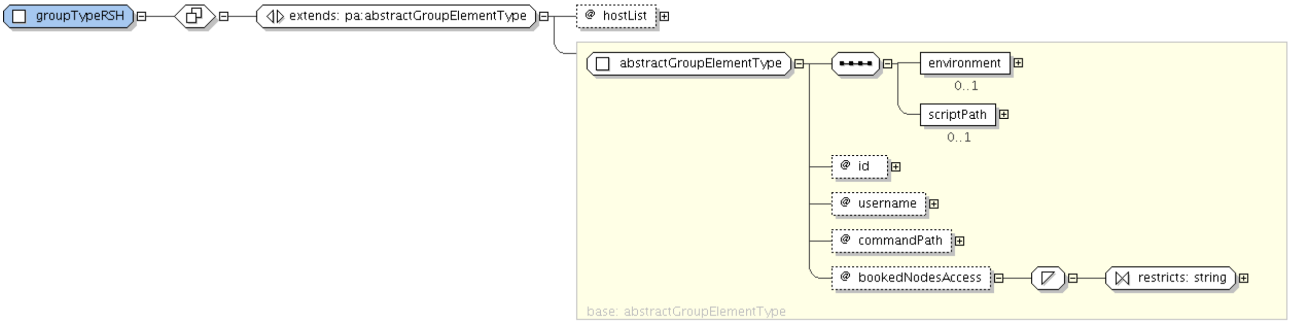


Figure 11

7.1.3 GroupOARSH

An oarsh group element has the same parameters as an ssh group element, plus a '**jobId**' (positive integer) mandatory attribute, which holds the OAR ID of the job.

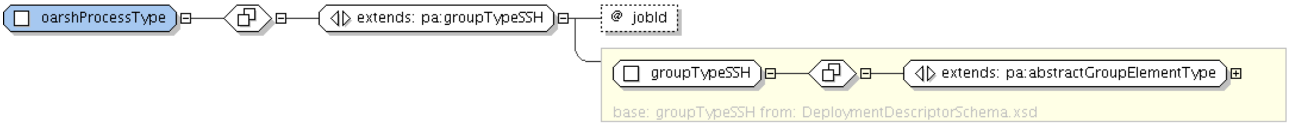

oarshProcessType
extends: pa:groupTypeSSH
jobId
groupTypeSSH
extends: pa:abstractGroupElementType
base: groupTypeSSH from: DeploymentDescriptorSchema.xsd

Figure 12

The following examples are group scheduling protocols. A scheduling protocol defines a way to communicate with a grid framework to submit job requests. The groups below allow the description of a deployment through such a framework.

7.1.4 ARC (NorduGrid)

Group definition for the NorduGrid's Advanced Resource Connector [i.1].

The NorduGrid group definition has the following child elements:

- **count** (positive integer)**:** amount of sub-jobs to be submitted for parallel tasks;
- **arguments** (argument list)**:** list of arguments passed to the job executable;
- **inputFiles** (sequence of 'ARCFileTransfer' elements)**:** list of files to be copied to the grid before the execution. This element also has a '**cache**' attribute (boolean) which specifies whether the files should be placed in the cache or not;
- **outputFiles** (sequence of 'ARCFileTransfer' elements)**:** list of files to be retrieved by the user;
- **stdout** (path string)**:** path of the file where the standard output of the job will be stored;
- **stderr** (path string)**:** path of the file where the standard error of the job will be stored;
- **stdin** (path string)**:** path of the file from which the standard input of the job will be read;
- **maxTime** (time)**:** maximum amount of time the job is allowed to run;
- **notify** (string)**:** email notification on job status change. The string is of the format accepted by the 'notify' parameter as defined in ARC's XRSL task description language.

It also has the following attribute:

- **jobName** (string)**:** the name of the job.

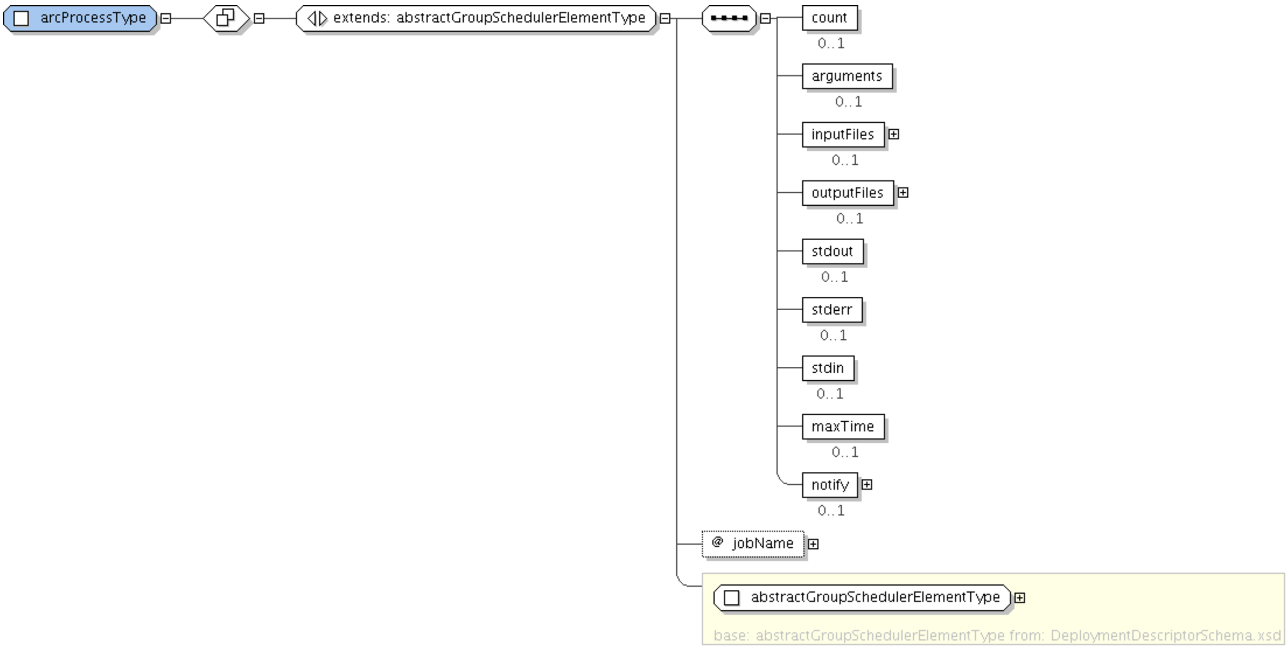

arcProcessType
extends: abstractGroupSchedulerElementType
count
0..1
arguments
0..1
inputFiles
0..1
outputFiles
0..1
stdout
0..1
stderr
0..1
stdin
0..1
maxTime
0..1
notify
0..1
@ jobName
abstractGroupSchedulerElementType
base: abstractGroupSchedulerElementType from: DeploymentDescriptorSchema.xsd

Figure 13

The `ARCFileTransfer` type is a simple list of `<transfer>` elements. A `<transfer>` element has only two attributes:

- **filename** (string)**:** the local name of the file;
- **location** (string)**:** the remote location of the file.

7.1.5 CGSP China Grid

Group definition for the ChinaGrid middleware [i.2].

The CGSP group definition has the following child elements:

- **count** (positive integer)**:** number of executions of the executable;
- **directory** (path string)**:** the working directory of the job script;
- **stdout** (path string)**:** path of the file where the standard output of the job will be stored;
- **stderr** (path string)**:** path of the file where the standard error of the job will be stored.

It also has the following attributes:

- **hostname** (string)**:** the name of the CGSP node to which the job will be submitted;
- **queue** (string)**:** name of the queue to submit the job to.

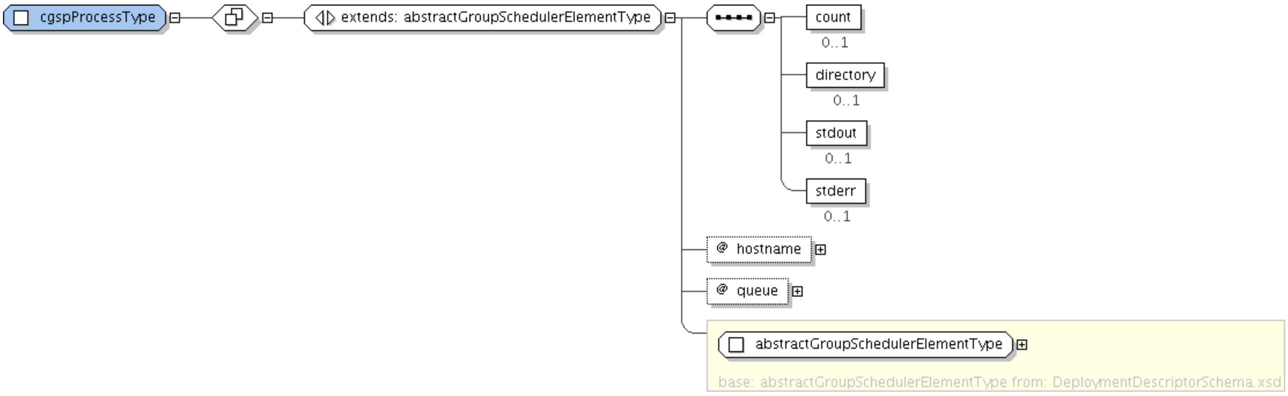

cgspProcessType
extends: abstractGroupSchedulerElementType
count
0..1
directory
0..1
stdout
0..1
stderr
0..1
hostname
queue
abstractGroupSchedulerElementType
base: abstractGroupSchedulerElementType from: DeploymentDescriptorSchema.xsd

Figure 14

7.1.6 EGEE gLITE

Group definition for the EGEE gLite grid framework [i.3].

The gLite group definition has the following child elements:

- **stdout** (path string)**:** path of the file where the standard output of the job will be stored;
- **stderr** (path string)**:** path of the file where the standard error of the job will be stored;
- **stdin** (path string)**:** path of the file from which the standard input of the job will be read;
- **arguments** (argument list)**:** list of arguments passed to the job executable.

It also has the following attributes:

- **virtualOrganisation** (string, required)**:** name of the virtual organization the job submitter is working for;
- **nodes** (postive integer)**:** number of nodes requested.

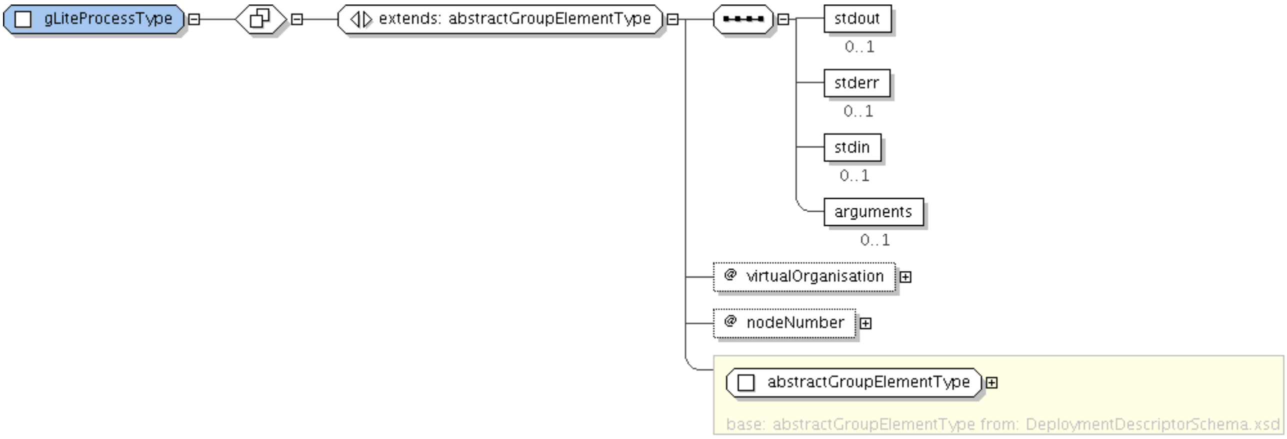


Figure 15

7.1.7 Fura/InnerGrid

Group definition for the Fura grid programming environment [i.4].

The Fura group definition has the following child elements:

- **description** (string)**:** a simple textual description of the job;
- **stdout** (path string)**:** path of the file where the standard output of the job will be stored;
- **stderr** (path string)**:** path of the file where the standard error of the job will be stored;
- **stdin** (path string)**:** path of the file from which the standard input of the job will be read;
- **maxTime** (time)**:** maximum amount of time the job is allowed to run.

It has one attribute:

- **jobName** (string)**:** the name of the job.

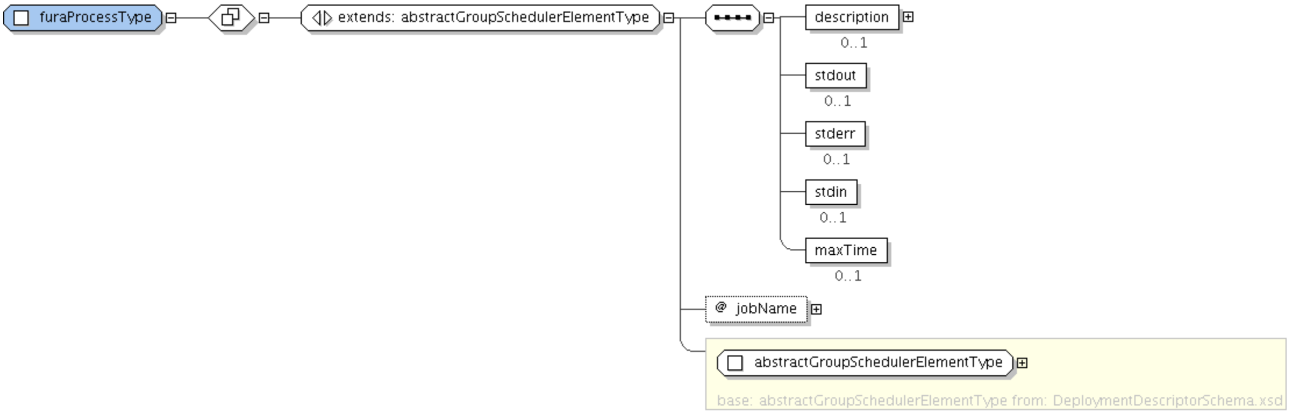

furaProcessType
extends: abstractGroupSchedulerElementType
description
0..1
stdout
0..1
stderr
0..1
stdin
0..1
maxTime
0..1
jobName
abstractGroupSchedulerElementType
base: abstractGroupSchedulerElementType from: DeploymentDescriptorSchema.xsd

Figure 16

7.1.8 GLOBUS

Group definition for the Globus toolkit [i.5].

The GLOBUS group definition has the following child elements:

- **count** (positive integer)**:** number of executions of the executable;
- **maxTime** (time)**:** maximum amount of time the job is allowed to run;

- **directory** (path string)**:** the working directory of the job script;
- **stdout** (path string)**:** path of the file where the standard output of the job will be stored;
- **stderr** (path string)**:** path of the file where the standard error of the job will be stored;
- **stdin** (path string)**:** path of the file where the standard error of the job will be stored.

It also has the following attributes:

- **hostname** (string)**:** the name of the Globus host to which the job will be submitted;
- **queue** (string)**:** name of the queue.

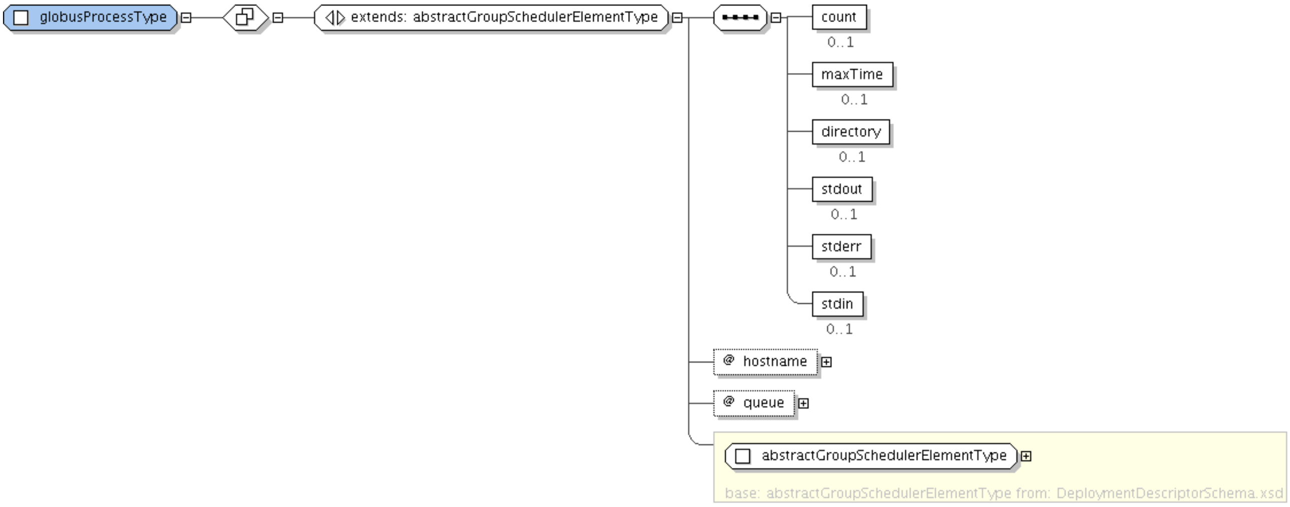

globusProcessType
extends: abstractGroupSchedulerElementType
count
0..1
maxTime
0..1
directory
0..1
stdout
0..1
stderr
0..1
stdin
0..1
hostname
queue
abstractGroupSchedulerElementType
base: abstractGroupSchedulerElementType from: DeploymentDescriptorSchema.xsd

Figure 17

7.1.9 GridBus

Group definition for the GridBus middleware [i.6].

The GridBus element has a single child element:

- **arguments** (argument list)**:** list of arguments passed to the job executable.

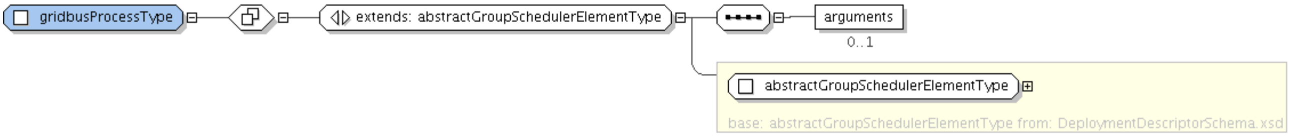

gridbusProcessType
extends: abstractGroupSchedulerElementType
arguments
0..1
abstractGroupSchedulerElementType
base: abstractGroupSchedulerElementType from: DeploymentDescriptorSchema.xsd

Figure 18

7.1.10 IBM Load Leveler

Group definition for IBM's Load Leveler job scheduler [i.7].

The Load Leveler group definition has the following child elements:

- **directory** (path string)**:** the working directory of the job script;
- **stdout** (path string)**:** path of the file where the standard output of the job will be stored;
- **stderr** (path string) path of the file where the standard error of the job will be stored;
- **resources** (string) a resource string as defined by the 'resources' parameter in the IBM Load Leveler: Using and Administering document [i.7];

- **maxTime** (time format string)**:** the maximum amount of real time the job is allowed to run;
- **arguments** (argument list)**:** list of arguments passed to the job executable;
- **taskRepartition** (structure)**:** this element contains the following element tree:
  - modeBlock:
    - **totalTasks** (positive integer)**:** total number of tasks to run on all available nodes
    - **blocking** (positive integer)**:** base factor of which multiples will be used to determine the number of tasks to run per node.
  - modeTotalTasks:
    - **totalTasks** (positive integer)**:** total number of tasks to run on all available nodes;
    - **node** (positive integer)**:** number of nodes requested.
  - modeTasksPerNode:
    - **tasksPerNode** (positive integer)**:** number of tasks to run on a single node;
    - **node** (positive integer)**:** number of nodes requested.
  - modeGeometry:
    - **taskGeometry** (string)**:** group tasks so they will run together on the same node (see IBM Load Leveler: Using and Administering document [i.7]).

It also has the following attribute:

- **jobName** (string)**:** the name of the job.

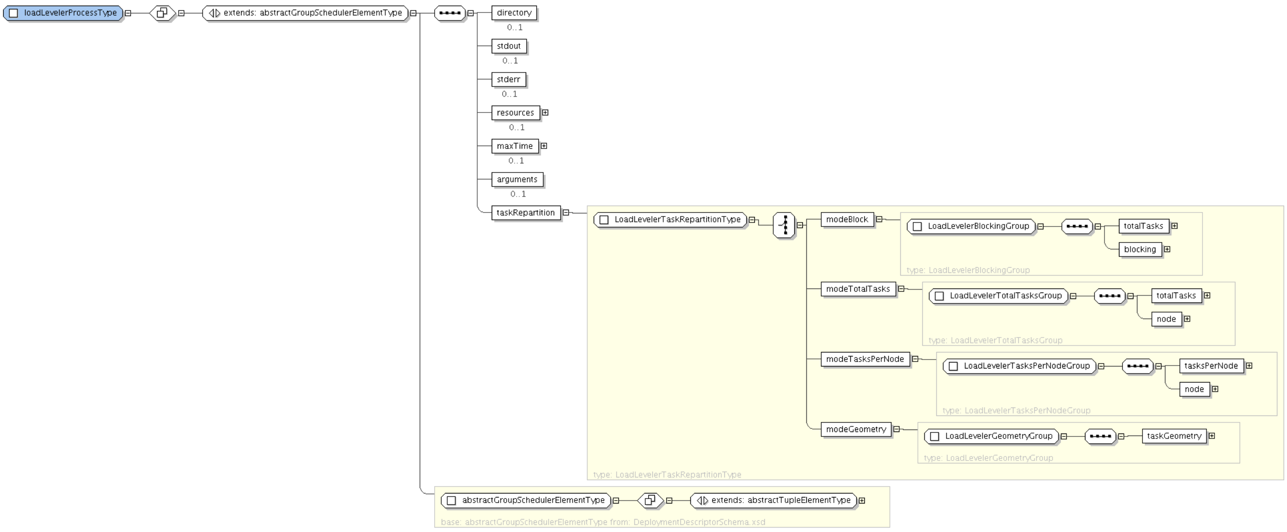

loadLevelerProcessType
extends: abstractGroupSchedulerElementType
directory
0..1
stdout
0..1
stderr
0..1
resources
0..1
maxTime
0..1
arguments
0..1
taskRepartition
LoadLevelerTaskRepartitionType
modeBlock
LoadLevelerBlockingGroup
totalTasks
blocking
type: LoadLevelerBlockingGroup
modeTotalTasks
LoadLevelerTotalTasksGroup
totalTasks
node
type: LoadLevelerTotalTasksGroup
modeTasksPerNode
LoadLevelerTasksPerNodeGroup
tasksPerNode
node
type: LoadLevelerTasksPerNodeGroup
modeGeometry
LoadLevelerGeometryGroup
taskGeometry
type: LoadLevelerGeometryGroup
type: LoadLevelerTaskRepartitionType
abstractGroupSchedulerElementType
extends: abstractTupleElementType
base: abstractGroupSchedulerElementType from: DeploymentDescriptorSchema.xsd

Figure 19

7.1.11 LSF

Group definition for the LSF scheduler [i.8].

The LSF group definition has the following child elements:

- **resource** (string)**:** this element has the following attributes:
  - **processorNumber** (positive integer)**:** number of processors requested;
  - **walltime** (time)**:** maximum time allowed for the job.

- **processorsNumber** (positive integer)**:** minimum number of processors requested to run the job;
- **resourceRequirement** (string)**:** a resource requirement string as defined by the lsf documentation ('lsfintro' manpage).

It also has the following attributes:

- **interactive** (boolean)**:** whether the job is interactive or not;
- **jobName** (string)**:** name of the job;
- **queue** (string)**:** name of the queue the job will be submitted in.

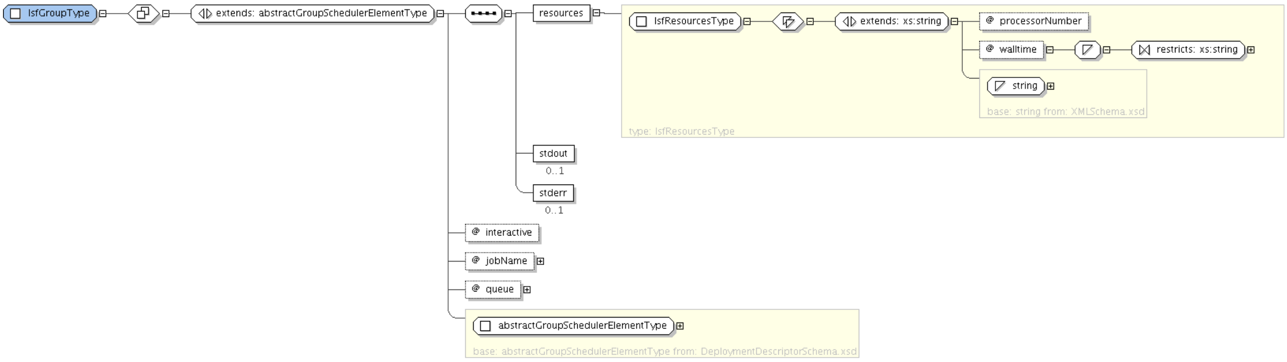

lsfGroupType
extends: abstractGroupSchedulerElementType
resources
lsfResourcesType
extends: xs:string
processorNumber
walltime
restricts: xs:string
string
base: string from: XMLSchema.xsd
type: lsfResourcesType
stdout
0..1
stderr
0..1
interactive
jobName
queue
abstractGroupSchedulerElementType
base: abstractGroupSchedulerElementType from: DeploymentDescriptorSchema.xsd

Figure 20

7.1.12 Microsoft CCS

Group Definition for Microsoft's Compute Cluster Server [i.13].

The CCS group definition has the following child elements:

- **resources:** the resources that will be allowed to the job. This element can have two children:
  - **cpus** (positive integer)**:** the number of CPUs allocated for the job;
  - **runtime** (time)**:** the maximum runtime allowed for the job.
- **stdout** (path string)**:** path of the file where the standard output of the job will be stored;
- **stderr** (path string)**:** path of the file where the standard error of the job will be stored.

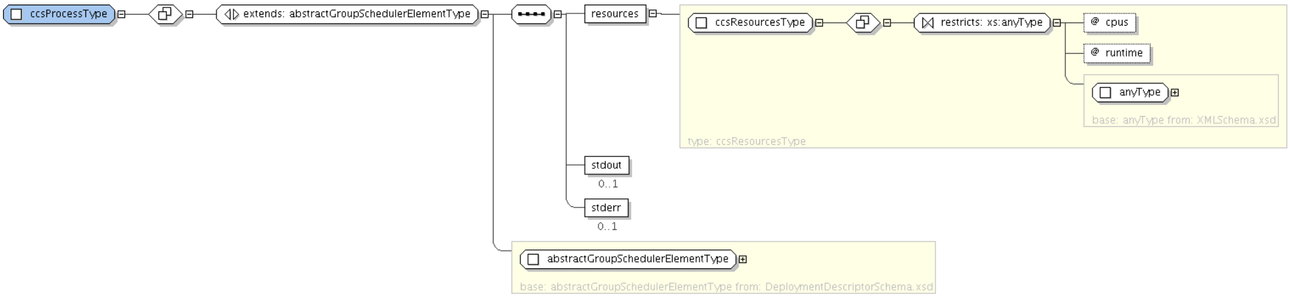

ccsProcessType
extends: abstractGroupSchedulerElementType
resources
ccsResourcesType
restricts: xs:anyType
cpus
runtime
anyType
base: anyType from: XMLSchema.xsd
type: ccsResourcesType
stdout
0..1
stderr
0..1
abstractGroupSchedulerElementType
base: abstractGroupSchedulerElementType from: DeploymentDescriptorSchema.xsd

Figure 21

7.1.13 OAR

Group definition for the OAR job scheduler [i.9].

The OAR group definition has the following child elements:

- **resource** (string)**:** this element has the following attributes:

- **nodes** (positive integer)**:** number of nodes requested;
- **cpu** (positive integer)**:** number of CPUs requested;
- **core** (positive integer)**:** number of cores requested;
- **walltime** (time)**:** maximum time allowed for the job.

It also can have a string content which is passed verbatim to the '-resource' option of the *oarsub* command.

NOTE: The *oarsub* command is a command of the OAR grid framework. It is defined in [i.9].

- **directory** (path string)**:** the working directory of the job script;
- **stdout** (path string)**:** path of the file where the standard output of the job will be stored;
- **stderr** (path string)**:** path of the file where the standard error of the job will be stored.

It has the following attributes:

- **interactive** (boolean)**:** start an interactive job. If true, open a login shell on the first node instead of running a script (default is false);
- **queue** (string)**:** name of the queue to submit the job to;
- **type** ('deploy', 'besteffort', 'cosystem', 'checkpoint', 'timesharing')**:** job type - the default is 'deploy'.

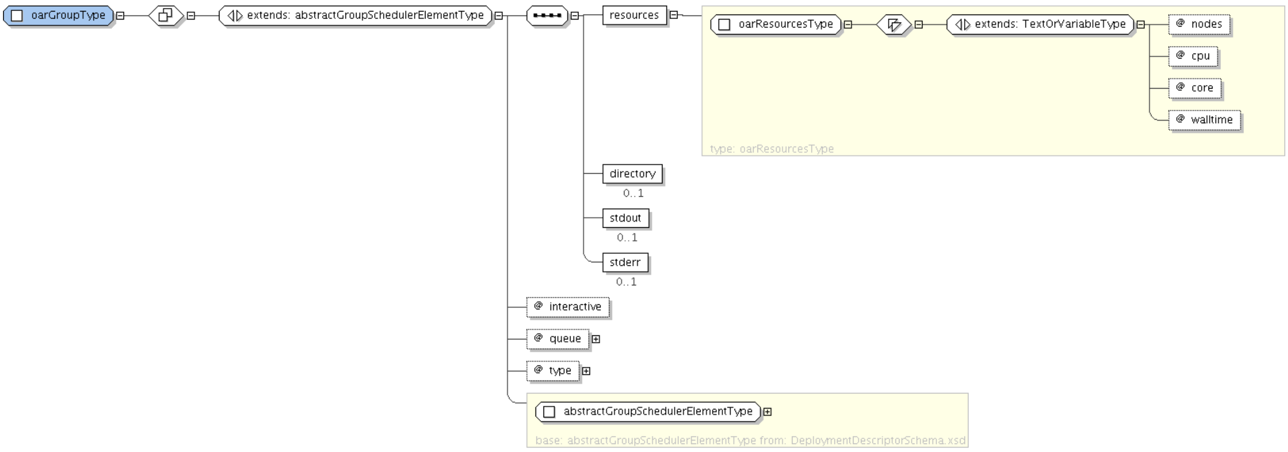

oarGroupType
extends: abstractGroupSchedulerElementType
resources
oarResourcesType
extends: TextOrVariableType
nodes
cpu
core
walltime
type: oarResourcesType
directory
0..1
stdout
0..1
stderr
0..1
interactive
queue
type
abstractGroupSchedulerElementType
base: abstractGroupSchedulerElementType from: DeploymentDescriptorSchema.xsd

Figure 22

7.1.14 PBS / Torque

Group definition for the Torque job scheduler [i.10].

The PBS/Torque group definition has the following child elements:

- **resource** (string)**:** this element has the following attributes:
  - **nodes** (positive integer)**:** number of nodes requested;
  - **ppn** (positive integer)**:** number of CPUs requested;
  - **walltime** (time)**:** maximum time allowed for the job.
- **nodes** (positive integer)**:** number of nodes requested;
- **processorsPerNode** (positive integer)**:** number of processors per node requested;
- **mailWhen** (combination of Abort, Begin, End separated by '|')**:** when to send an email (Abort: if the job is aborted, Begin: when the job is started, End: when the job terminates);

- **mailTo** (comma-separated list of email addresses)**:** where the job status emails should be sent;
- **joinOutput** (boolean)**:** if true, join the output of stderr to stdout;
- **stdout** (path string)**:** path of the file where the standard output of the job will be stored;
- **stderr** (path string)**:** path of the file where the standard error of the job will be stored.

It has the following attributes:

- **queue** (string)**:** destination queue for the job. The argument can be of the following format:
  - queue: a queue on the default server;
  - @server: the default queue on the server;
  - queue@server: the queue on the given server.
- **jobName** (string 15 chars long, no whitespace, first char must be alphabetic)**:** the name of the job;
- **interactive** (boolean)**:** whether the job is interactive or not.

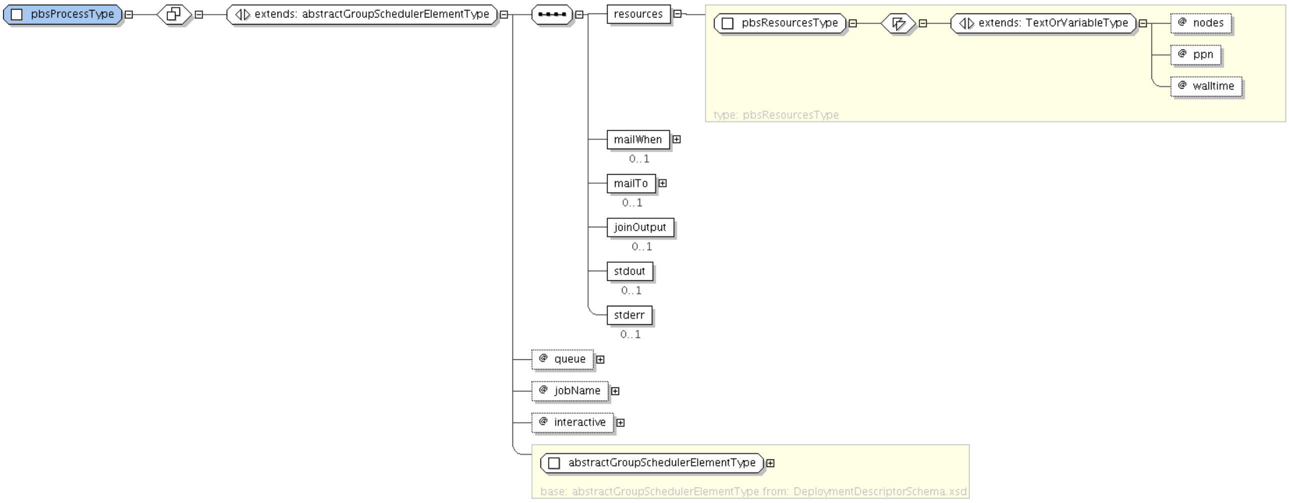

pbsProcessType
extends: abstractGroupSchedulerElementType
resources
pbsResourcesType
extends: TextOrVariableType
nodes
ppn
walltime
type: pbsResourcesType
mailWhen
0..1
mailTo
0..1
joinOutput
0..1
stdout
0..1
stderr
0..1
queue
jobName
interactive
abstractGroupSchedulerElementType
base: abstractGroupSchedulerElementType from: DeploymentDescriptorSchema.xsd

Figure 23

7.1.15 PRUN

Group definition for the PRUN run server [i.11].

The PRUN group definition has the following child elements:

- **resource** (string)**:** this element has the following attributes:
  - **nodes** (positive integer)**:** number of nodes requested;
  - **ppn** (positive integer)**:** number of CPUs requested;
  - **walltime** (time)**:** maximum time allowed for the job.
- **stdout** (path string)**:** name of the file in which the results will be printed.

It has no attribute.

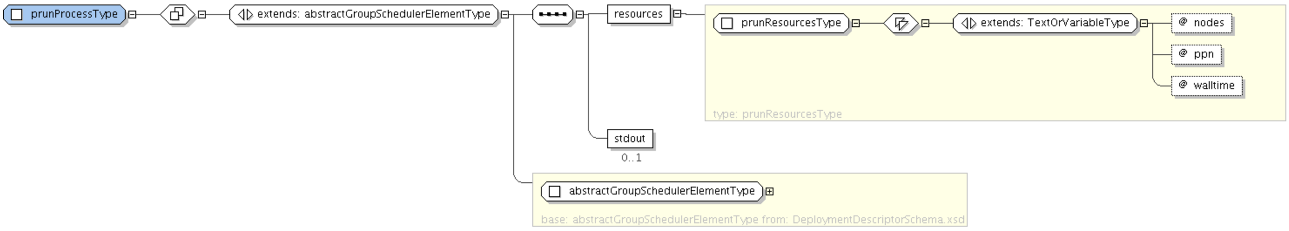

prunProcessType
extends: abstractGroupSchedulerElementType
resources
prunResourcesType
extends: TextOrVariableType
@ nodes
@ ppn
@ walltime
type: prunResourcesType
stdout
0..1
abstractGroupSchedulerElementType
base: abstractGroupSchedulerElementType from: DeploymentDescriptorSchema.xsd

Figure 24

7.1.16 Sun Grid Engine

Group definition for the Grid Engine framework [i.12].

The Sun Grid Engine group definition has the following child elements:

- **resource** (string)**:** this element has the following attributes:
  - **nodes** (positive integer)**:** number of nodes requested;
  - **parallelEnvironment** (string)**:** the name of the Grid Engine parallel environment which will be instantiated;
  - **walltime** (time)**:** maximum time allowed for the job.
- **directory** (path string)**:** the working directory of the job script;
- **stdout** (path string)**:** path of the file where the standard output of the job will be stored;
- **stderr** (path string)**:** path of the file where the standard error of the job will be stored.

It also has the following attributes:

- **queue** (string)**:** queue name;
- **jobName** (string)**:** job name.

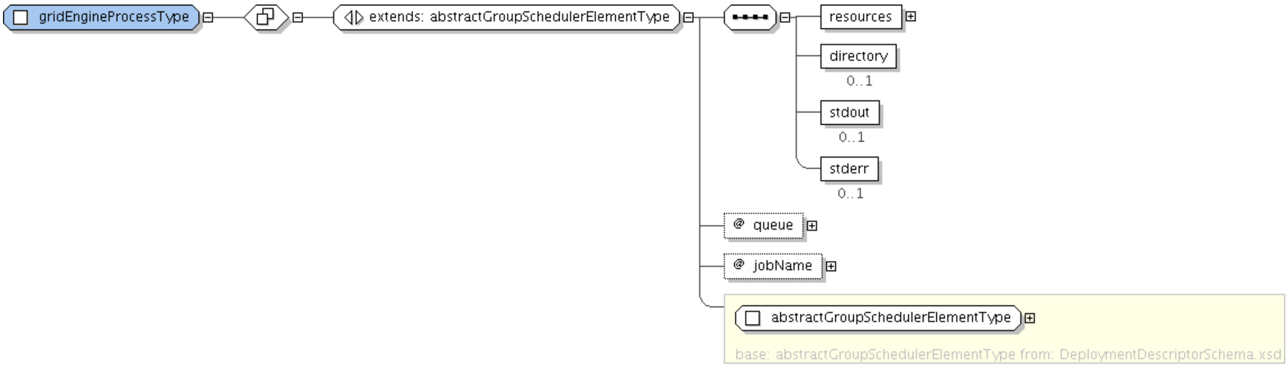

gridEngineProcessType
extends: abstractGroupSchedulerElementType
resources
directory
0..1
stdout
0..1
stderr
0..1
@ queue
@ jobName
abstractGroupSchedulerElementType
base: abstractGroupSchedulerElementType from: DeploymentDescriptorSchema.xsd

Figure 25

8 XML files examples (informative)

8.1 Bridge protocols

8.1.1 rsh

```
<rshBridge id="brSchubby" hostname="schubby.inria.fr" username="${USERNAME}" />
```

8.1.2 ssh

```
<sshBridge id="SEA" hostname="sea.inria.fr" username="${USERNAME}" commandPath="" />
```

8.1.3 oarsh

```
<oarshBridge id="oarbridge" hostname="oargrid.inria.fr" jobId="12" />
```

8.1.4 gsissh

```
<gsisshBridge id="gsibridge" hostname="grid11.lal.in2p3.fr"
port="2222"  certificate="${HOME}/.globus/usercred.p12" />
```

8.2 Group protocols

8.2.1 GroupSSH

```
<sshGroup id="LAN_INRIA" hostList="node2 node4 host[10-20]" username="${USERNAME}"
                commandPath="/usr/local/bin/ssh">
                <scriptPath base="proactive" relpath="myscript.sh" />
</sshGroup>
```

8.2.2 GroupRSH

```
<rshGroup id="LAN_INRIA" hostList="node2 node4 host[10-20]" username="${USERNAME}">
                <scriptPath base="proactive" relpath="myscript.sh" />
</rshGroup>
```

8.2.3 GroupOARSH

```
<oarshGroup id="oarshgroup" hostList="host1 host2" jobId="42">
                <scriptPath base="proactive" relpath="scripts/unix/cluster/oarStartRuntime.sh" />
</oarshGroup>
```

8.2.4 ARC (Nordugrid)

```
<arcGroup id="arcgroup" jobName="arcjob">
                <scriptPath base="root"
relpath="/usr/local/arc/scripts/unix/cluster/startRuntime.sh" />
                <stderr>trace/err.log</stderr>
                <maxTime>10000000</maxTime>
</arcGroup>
```

8.2.5 CGSP China Grid

```
<cgspGroup id="cgspgroup" hostname="cgsp.inria.fr">
                <scriptPath base="root" relpath="./calc" />
                <count>50</count>
                <directory>datadir/jobdata1</directory>
                <stdout>trace/results.log</stdout>
                <stderr>trace/err.log</stderr>
</cgspGroup>
```

8.2.6 EGEE gLite

```
<gLiteGroup id="gliteGroup" virtualOrganisation="proactive" jobType="normal"
                retryCount="3">
                <rank>-other.GlueCEStateEstimatedResponseTime</rank>
                <stdout>stdout.log</stdout>
                <stderr>error.log</stderr>
                <configFile>scripts/unix/proactiveVO/proactiveVO.conf</configFile>
</gLiteGroup>
```

8.2.7 Fura/InnerGrid

```
<furaGroup id="furagroup" jobName="furajob">
                <scriptPath base="root" relpath="/usr/local/fura/scripts/jobscript.sh" />
                <description>Fura job descriptor sample</description>
                <stdout>trace/results.log</stdout>
                <stderr>trace/err.log</stderr>
                <maxTime>15000000</maxTime>
</furaGroup>
```

8.2.8 GLOBUS

```
<globusGroup hostname="globus_frontend" id="globusgroup">
                <count>8</count>
                <maxTime>120</maxTime>
                <stderr>${GLOBUS_USER_HOME}/error.txt</stderr>
</globusGroup>
```

8.2.9 GridBus

```
<gridbusGroup id="gridbusgroup">
                <scriptPath base="root" relpath="./calc" />
                <arguments>
                    <arg>arg1</arg>
                    <arg>arg2</arg>
                </arguments>
</gridbusGroup>
```

8.2.10 IBM load leveler

```
<loadLevelerGroup id="loadlevelergroup">
                <scriptPath base="root" relpath="/usr/local/loadleveler/scripts/jobscript.sh" />
                <directory>datadir/jobdata1</directory>
                <stdout>trace/results.log</stdout>
                <stderr>trace/err.log</stderr>
                <maxTime>15000000</maxTime>
                <arguments>
                    <arg>arg1</arg>
                    <arg>25</arg>
                    <arg>foo bar</arg>
                </arguments>
                <nbTasks>10</nbTasks>
                <tasksPerHost>1</tasksPerHost>
</loadLevelerGroup>
```

8.2.11 LSF

```
<lsfGroup id="lsfgroup">
                <scriptPath base="proactive" relpath="scripts/unix/cluster/startRuntime.sh" />
                <processorsNumber>2</processorsNumber>
                <resourceRequirement>value="span[ptile=2]"</resourceRequirement>
</lsfGroup>
```

8.2.12 Microsoft CCS

```
<ccsGroup id="ccs" bookedNodesAccess="rsh">
    <resources cpus="9" runtime="01:01:01"/> <!-runtime=days:hours:min -->
    <stdout>\\HN\work\out</stdout>
    <stderr>\\HN\work\err</stderr>
</ccsGroup>
```

8.2.13 OAR

```
<oarGroup bookedNodesAccess="ssh" id="oargroup" interactive="true">
                <scriptPath base="proactive" relpath="scripts/unix/cluster/oarStartRuntime.sh" />
                <resource nodes="2" cpu="4" />
                <stderr>trace/err.log</stderr>
</oarGroup>
```

8.2.14 PBS / Torque

```
<pbsGroup id="pbsgroup1" queue="queue01@pbsserver" jobName="demoJob"
                commandPath="/usr/local/bin/pbs">
                <scriptPath base="home" relpath="scripts/pbsjob.sh" />
                <nodes>50</nodes>
</pbsGroup>
```

8.2.15 PRUN

```
<prunGroup id="prungroup1" queue="prunqueue">
                <hostList>prunhost[00-10]</hostList>
                <processorPerNode>2</processorPerNode>
</prunGroup>
```

8.2.16 Sun Grid Engine

```
<gridEngineGroup id="gridEngineGroup">
                <scriptPath base="home" relpath="jobs/jobscript1.sh" />
                <hostsNumber>10</hostsNumber>
                <wallTime>500000</wallTime>
                <directory>"jobs/job_datas</directory>
                <stderr>jobs/jobtrace_err.txt</stderr>
</gridEngineGroup>
```

Annex A (normative): XML Schema

A.1 Common types

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xsd:simpleType name="variableRefType">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="\$\{[A-Za-z0-9._]+\}" />
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="TimeType"><!-hour:mn:sec|hour:mn|hour -->
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="\d\d?(:\d\d)?(:\d\d)?" />
        </xsd:restriction>
    </xsd:simpleType>

    <!-PosintOrVariableType is defined as a xsd:union of a positive integer or a variable ref-->
    <xsd:simpleType name="PosintOrVariableType">
        <xsd:union memberTypes="xsd:positiveInteger" />
    </xsd:simpleType>
    <!-BooleanOrVariableType is defined as a xsd:union of a positive integer or a variable ref-->
    <xsd:simpleType name="BooleanOrVariableType">
        <xsd:union memberTypes="xsd:boolean" />
    </xsd:simpleType>
    <xsd:simpleType name="TimeOrVariableType">
        <xsd:union memberTypes="xsd:time" />
    </xsd:simpleType>
    <xsd:simpleType name="TextOrVariableType">
        <xsd:union memberTypes="xsd:string" />
    </xsd:simpleType>
    <!-ListType is defined as a xsd:union of a positive integer or a variable ref-->
    <xsd:simpleType name="ListType">
        <xsd:list itemType="hostnameOrVariableType" />
    </xsd:simpleType>

    <!-CloseStreamType is defined as a global xsd:element, it is the xsd:union of a CloseStream and
a variable type.-->
    <xsd:simpleType name="CloseStreamOrVariableType">
        <xsd:union memberTypes="CloseStreamType" />
    </xsd:simpleType>
    <xsd:simpleType name="CloseStreamType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="yes" />
            <xsd:enumeration value="no" />
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="pathElementBaseType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="proactive" />
            <xsd:enumeration value="home" />
            <xsd:enumeration value="root" />
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="idType">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="[0-9a-zA-Z_\-]+"></xsd:pattern>
            <xsd:whiteSpace value="collapse"></xsd:whiteSpace>
            <xsd:minLength value="1"></xsd:minLength>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="capacityType">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="([0-9]+)|max"></xsd:pattern>
            <xsd:whiteSpace value="collapse"></xsd:whiteSpace>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="pathElementGeneralType">
        <xsd:restriction base="xsd:string">
            <xsd:whiteSpace value="collapse"></xsd:whiteSpace>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:complexType name="pathElementType">
        <xsd:attribute name="base" type="pathElementBaseType" use="required"></xsd:attribute>
        <xsd:attribute name="relpath" type="pathElementGeneralType" use="required"></xsd:attribute>
    </xsd:complexType>
    <xsd:complexType name="pathType">
        <xsd:sequence>
            <xsd:element name="pathElement" type="pathElementType" maxOccurs="unbounded"
minOccurs="1"></xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="classpathType">
        <xsd:complexContent>
            <xsd:extension base="pathType">
                <xsd:attribute name="type" use="required">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="append"></xsd:enumeration>
                            <xsd:enumeration value="overwrite"></xsd:enumeration>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:simpleType name="osType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="unix"></xsd:enumeration>
            <xsd:enumeration value="windows"></xsd:enumeration>
            <xsd:enumeration value="cygwin"></xsd:enumeration>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="osOrVariableType">
        <xsd:union memberTypes="osType" />
    </xsd:simpleType>
    <xsd:simpleType name="hostnameType">
        <xsd:union memberTypes="xsd:string" />
    </xsd:simpleType>
    <xsd:simpleType name="hostnameOrVariableType">
        <xsd:union memberTypes="hostnameType" />
    </xsd:simpleType>
    <xsd:simpleType name="usernameType">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="[a-z0-9]+" />
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="usernameOrVariableType">
        <xsd:union memberTypes="usernameType" />
    </xsd:simpleType>

    <xsd:complexType name="environmentElementType" />
    <xsd:complexType name="variableType">
        <xsd:complexContent>
            <xsd:extension base="environmentElementType">
                <xsd:attribute name="name">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string"></xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
                <xsd:attribute name="value">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string"></xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="includeFileType">
        <xsd:complexContent>
            <xsd:extension base="environmentElementType">
                <xsd:attribute name="location" type="xsd:string" use="required" />
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:element name="abstractEnvironmentElement" abstract="true"
type="environmentElementType"></xsd:element>
```

```
    <xsd:element name="descriptorVariable" substitutionGroup="abstractEnvironmentElement"
type="variableType"></xsd:element>
    <xsd:element name="descriptorDefaultVariable" substitutionGroup="abstractEnvironmentElement"
type="variableType"></xsd:element>
    <xsd:element name="javaPropertyVariable" substitutionGroup="abstractEnvironmentElement"
type="variableType"></xsd:element>
    <xsd:element name="programVariable" substitutionGroup="abstractEnvironmentElement"
type="variableType"></xsd:element>
    <xsd:element name="programDefaultVariable" substitutionGroup="abstractEnvironmentElement"
type="variableType"></xsd:element>
    <xsd:element name="javaPropertyDescriptorDefault" substitutionGroup="abstractEnvironmentElement"
type="variableType"></xsd:element>
    <xsd:element name="javaPropertyProgramDefault" substitutionGroup="abstractEnvironmentElement"
type="variableType"></xsd:element>
    <xsd:element name="includePropertyFile" substitutionGroup="abstractEnvironmentElement"
type="includeFileType"></xsd:element>

    <xsd:complexType name="environmentType">
        <xsd:sequence>
            <xsd:element ref="abstractEnvironmentElement" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="environmentVariablesType">
        <xsd:sequence>
            <xsd:element name="variable" type="variableType" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="argumentListType">
        <xsd:sequence>
            <xsd:element name="arg" type="xsd:string" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

A.2 Main schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:gcm:deployment:1.0"
xmlns="urn:gcm:deployment:1.0"
    xmlns:ns="urn:gcm:deployment:1.0" elementFormDefault="qualified">

    <xsd:include schemaLocation="./CommonTypes.xsd" />

    <xsd:complexType name="abstractTupleElementType" abstract="true">
        <xsd:sequence>
            <xsd:element name="environment" type="environmentVariablesType" minOccurs="0" />
        </xsd:sequence>
        <xsd:attribute name="id" type="idType" use="required" />
        <xsd:attribute name="commandPath" type="pathElementGeneralType" />
    </xsd:complexType>

    <xsd:complexType name="abstractGroupElementType">
        <xsd:complexContent>
            <xsd:extension base="abstractTupleElementType">
                <xsd:attribute name="username" type="ns:usernameOrVariableType" />
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="abstractGroupSchedulerElementType">
        <xsd:complexContent>
            <xsd:extension base="abstractTupleElementType">
                <xsd:sequence>
                    <xsd:element name="scriptPath" type="pathElementType" minOccurs="0" />
                </xsd:sequence>
                <xsd:attribute name="bookedNodesAccess" use="required">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:enumeration value="rsh" />
                            <xsd:enumeration value="ssh" />
                            <xsd:enumeration value="oarsh" />
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:element name="abstractShElement" type="abstractTupleElementType"></xsd:element>

    <xsd:complexType name="peerSetType">
        <xsd:sequence>
            <xsd:element name="peer" type="xsd:string" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>

    <!-a 'MAX' xsd:string constant -->
    <xsd:simpleType name="MAX">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="MAX"></xsd:enumeration>
        </xsd:restriction>
    </xsd:simpleType>

    <xsd:simpleType name="lookupProtocolType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="RMI"></xsd:enumeration>
            <xsd:enumeration value="HTTP"></xsd:enumeration>
            <xsd:enumeration value="IBIS"></xsd:enumeration>
        </xsd:restriction>
    </xsd:simpleType>

    <xsd:complexType name="localClientType">
        <xsd:attribute name="protocol" type="lookupProtocolType"></xsd:attribute>
        <xsd:attribute name="port" type="xsd:positiveInteger"></xsd:attribute>
    </xsd:complexType>

    <xsd:complexType name="p2pType">
        <xsd:sequence>
            <xsd:element name="localClient" type="localClientType"/>
            <xsd:element name="peerSet" type="peerSetType"/>
        </xsd:sequence>
        <xsd:attribute name="nodesAsked">
            <xsd:simpleType>
                <xsd:union memberTypes="xsd:positiveInteger MAX" />
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:complexType>

    <xsd:complexType name="lookupType">
        <xsd:attribute name="type" type="lookupProtocolType"></xsd:attribute>
        <xsd:attribute name="hostList" type="xsd:string"></xsd:attribute>
        <xsd:attribute name="port" type="xsd:positiveInteger"></xsd:attribute>
    </xsd:complexType>


    <xsd:complexType name="acquisitionType">
        <xsd:sequence>
            <xsd:element name="lookup" type="lookupType" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="p2p" type="p2pType" minOccurs="0" maxOccurs="1" />
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="abstractResourceType"></xsd:complexType>

    <xsd:complexType name="groupRefType">
        <xsd:complexContent>
            <xsd:extension base="abstractResourceType">
                <xsd:sequence>
                    <xsd:element name="host" type="hostRefType"></xsd:element>
                </xsd:sequence>
                <xsd:attribute name="refid" type="idType" use="required" />
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="bridgeRefType">
        <xsd:complexContent>
            <xsd:extension base="abstractResourceType">
                <xsd:sequence>
                    <xsd:element name="group" type="groupRefType" minOccurs="0"
maxOccurs="unbounded"></xsd:element>
```

```
                    <xsd:element name="bridge" type="bridgeRefType" minOccurs="0"
maxOccurs="unbounded"></xsd:element>
                    <xsd:element name="host" type="hostRefType" minOccurs="0" maxOccurs="1" />
                </xsd:sequence>
                <xsd:attribute name="refid" type="idType" use="required" />
                <xsd:attribute name="relay" type="xsd:string"></xsd:attribute>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:element name="abstractResourceElement" abstract="true"
type="abstractResourceType"></xsd:element>

    <xsd:element name="group" substitutionGroup="abstractResourceElement"
type="groupRefType"></xsd:element>

    <xsd:element name="bridge" substitutionGroup="abstractResourceElement"
type="bridgeRefType"></xsd:element>

    <xsd:complexType name="resourceType">
        <xsd:sequence>
            <xsd:element name="host" type="hostRefType" minOccurs="0" maxOccurs="1"></xsd:element>
            <xsd:element ref="abstractResourceElement" minOccurs="0"
maxOccurs="unbounded"></xsd:element>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="toolType">
        <xsd:attribute name="id" type="xsd:string"></xsd:attribute>
        <xsd:attribute name="path" type="pathElementGeneralType"></xsd:attribute>
        <xsd:attribute name="version" type="xsd:string"></xsd:attribute>
    </xsd:complexType>

    <xsd:complexType name="homeDirectoryType">
        <xsd:complexContent>
            <xsd:restriction base="pathElementType">
                <xsd:attribute name="base" type="pathElementBaseType" fixed="root"
use="required"></xsd:attribute>
                <xsd:attribute name="relpath" type="pathElementGeneralType"
use="required"></xsd:attribute>
            </xsd:restriction>
        </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="hostType">
        <xsd:sequence>
            <xsd:element name="homeDirectory" type="homeDirectoryType"></xsd:element>
            <xsd:element name="tool" type="toolType" minOccurs="0"
maxOccurs="unbounded"></xsd:element>
            <xsd:element name="networkInterface" minOccurs="0" maxOccurs="1">
                <xsd:complexType>
                    <xsd:attribute name="name" type="ns:TextOrVariableType" use="required"/>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
        <xsd:attribute name="id" type="idType" use="required"></xsd:attribute>
        <xsd:attribute name="os" type="osOrVariableType" use="required"></xsd:attribute>
        <xsd:attribute name="username" type="TextOrVariableType"></xsd:attribute>
        <xsd:attribute name="hostCapacity" type="PosintOrVariableType" use="optional" />
        <xsd:attribute name="vmCapacity" type="PosintOrVariableType" use="optional" />
    </xsd:complexType>

    <xsd:complexType name="hostRefType">
        <xsd:attribute name="refid" type="idType" use="required" />
    </xsd:complexType>

    <xsd:complexType name="abstractInfrastructureType"></xsd:complexType>

    <xsd:complexType name="hostsType">
        <xsd:sequence>
            <xsd:element name="host" type="hostType" minOccurs="1" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="groupsType">
        <xsd:sequence minOccurs="1">
            <xsd:element ref="abstractShElement" minOccurs="1" maxOccurs="unbounded"></xsd:element>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="abstractBridgeElementType">
        <xsd:attribute name="id" type="idType" use="required" />
    </xsd:complexType>

    <xsd:element name="abstractBridgeElement" type="abstractBridgeElementType"></xsd:element>


    <xsd:complexType name="bridgesType">
        <xsd:sequence>
            <xsd:element ref="abstractBridgeElement" maxOccurs="unbounded"></xsd:element>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="infrastructureType">
        <xsd:all>
            <xsd:element name="hosts" type="hostsType" minOccurs="0" />
            <xsd:element name="groups" type="groupsType" minOccurs="0" />
            <xsd:element name="bridges" type="bridgesType" minOccurs="0" />
        </xsd:all>
    </xsd:complexType>

    <xsd:element name="GCMDeployment">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="environment" type="environmentType" minOccurs="0" />
                <xsd:element name="resources" type="resourceType" />
                <xsd:element name="acquisition" type="acquisitionType" minOccurs="0" />
                <xsd:element name="infrastructure" type="infrastructureType" />
            </xsd:sequence>
        </xsd:complexType>

        <!-Groups refid/id key correctness -->
        <!--  -->
        <xsd:key name="idKeyGroups">
            <xsd:selector xpath="ns:infrastructure/ns:groups/*" />
            <xsd:field xpath="@id" />
        </xsd:key>
        <xsd:keyref name="refIdKeyGroups" refer="idKeyGroups">
            <xsd:selector xpath="ns:resources/ns:group"></xsd:selector>
            <xsd:field xpath="@refid"></xsd:field>
        </xsd:keyref>

        <!-Bridges refid/id key correctness -->
        <!--  -->
        <xsd:key name="idKeyBridges">
            <xsd:selector xpath="ns:infrastructure/ns:bridges/*" />
            <xsd:field xpath="@id" />
        </xsd:key>
        <xsd:keyref name="refIdKeyBridge" refer="idKeyBridges">
            <xsd:selector xpath="ns:resources/ns:bridge"></xsd:selector>
            <xsd:field xpath="@refid"></xsd:field>
        </xsd:keyref>

        <!-Hosts refid/id key correctness -->
        <!--  -->
        <xsd:key name="idKeyHosts">
            <xsd:selector xpath="ns:infrastructure/ns:hosts/*" />
            <xsd:field xpath="@id" />
        </xsd:key>
        <xsd:keyref name="refIdKeyHost" refer="idKeyHosts">
            <xsd:selector xpath="ns:resources/ns:host"></xsd:selector>
            <xsd:field xpath="@refid"></xsd:field>
        </xsd:keyref>
        <xsd:keyref name="refIdKeyGroupHost" refer="idKeyHosts">
            <xsd:selector xpath="ns:resources/ns:group/ns:host"></xsd:selector>
            <xsd:field xpath="@refid"></xsd:field>
        </xsd:keyref>

    </xsd:element>
</xsd:schema>
```

A.3 Bridges

A.3.1 rsh

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>
    <xs:complexType name="bridgeTypeRSH">
        <xs:complexContent>
            <xs:extension base="abstractBridgeElementType">
                <xs:attribute name="hostname" type="hostnameType" use="required" />
                <xs:attribute name="username" type="usernameType"/>
                <xs:attribute name="commandPath" type="xs:string"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="rshBridge" type="bridgeTypeRSH" substitutionGroup="abstractBridgeElement"/>

</xs:schema>
```

A.3.2 ssh

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="bridgeTypeSSH">
        <xs:complexContent>
            <xs:extension base="abstractBridgeElementType">
                <xs:sequence>
                    <xs:element name="privateKey" type="pathElementType" minOccurs="0" maxOccurs="1"
/>
                </xs:sequence>
                <xs:attribute name="hostname" type="hostnameType" use="required" />
                <xs:attribute name="username" type="usernameType"/>
                <xs:attribute name="commandPath" type="xs:string"/>
                <xs:attribute name="commandOptions" type="xs:string"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="sshBridge" type="bridgeTypeSSH" substitutionGroup="abstractBridgeElement"/>
</xs:schema>
```

A.3.3 gsissh

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="urn:gcm:deployment:1.0"
   xmlns="urn:gcm:deployment:1.0"
   elementFormDefault="qualified">

  <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

  <xs:include schemaLocation="./BridgeGSISSHSchema.xsd" />

   <xs:complexType name="bridgeTypeGSISSH">
      <xs:complexContent>
```

```
         <xs:extension base="bridgeTypeSSH">
            <xs:attribute name="port" type="PosintOrVariableType" />
            <xs:attribute name="certificate" type="xs:string"/>
         </xs:extension>
    </xs:complexContent>
  </xs:complexType>

   <xs:element name="gsisshBridge" type="bridgeTypeGSISSH"
substitutionGroup="abstractBridgeElement"/>
</xs:schema>
```

A.3.4 oarsh

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:include schemaLocation="./BridgeSSHSchema.xsd" />

    <xs:complexType name="bridgeTypeOARSH">
        <xs:complexContent>
            <xs:extension base="bridgeTypeSSH">
                <xs:attribute name="jobId" type="PosintOrVariableType" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="oarshBridge" type="bridgeTypeOARSH"
substitutionGroup="abstractBridgeElement"/>
</xs:schema>
```

A.4 Groups

A.4.1 GroupSSH

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="groupTypeSSH">
        <xs:complexContent>
            <xs:extension base="abstractGroupElementType">
                <xs:sequence>
                    <xs:element name="privateKey" type="pathElementType" minOccurs="0" maxOccurs="1"
/>
                </xs:sequence>
                <xs:attribute name="commandOptions" type="xs:string"/>
                <xs:attribute name="hostList" type="xs:string" use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="sshGroup" substitutionGroup="abstractShElement" type="groupTypeSSH"/>
</xs:schema>
```

A.4.2 GroupRSH

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="groupTypeRSH">
        <xs:complexContent>
            <xs:extension base="abstractGroupElementType">
                <xs:attribute name="hostList" type="xs:string" use="required" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="rshGroup" substitutionGroup="abstractShElement" type="groupTypeRSH" />
</xs:schema>
```

A.4.3 GroupOARSH

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>
    <xs:include schemaLocation="./GroupSSHSchema.xsd" />

    <xs:complexType name="oarshGroupType">
        <xs:complexContent>
            <xs:extension base="groupTypeSSH">
                <xs:attribute name="jobId" type="PosintOrVariableType" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="oarshGroup" type="oarshGroupType" substitutionGroup="abstractShElement"/>
</xs:schema>
```

A.4.4 ARC (NorduGrid)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="arcProcessType">
        <xs:complexContent>
            <xs:extension base="abstractGroupSchedulerElementType">
                <xs:sequence>
                    <xs:element name="count" type="PosintOrVariableType"
                        minOccurs="0" />
                    <xs:element name="arguments" type="argumentListType"
                        minOccurs="0" />
                    <xs:element name="inputFiles"
                        type="arcFileTransferType" minOccurs="0" />
                    <xs:element name="outputFiles"
                        type="arcFileTransferType" minOccurs="0" />
                    <xs:element name="stdout"
                        type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stderr"
                        type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stdin"
```

```
                        type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="maxTime"
                        type="PosintOrVariableType" minOccurs="0" />
                    <xs:element name="notify" type="xs:string" minOccurs="0" />
                </xs:sequence>
                <xs:attribute name="jobName" type="xs:string" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="arcFileTransferType">
        <xs:sequence>
            <xs:element name="transfer">
                <xs:complexType>
                    <xs:attribute name="filename" type="xs:string" />
                    <xs:attribute name="location" type="xs:string" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

    <xs:element name="arcGroup" type="arcProcessType"
        substitutionGroup="abstractShElement">
    </xs:element>

</xs:schema>
```

A.4.5 CGSP China Grid

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="cgspProcessType">
        <xs:complexContent>
            <xs:extension base="abstractGroupSchedulerElementType">
                <xs:sequence>
                    <xs:element name="count" type="PosintOrVariableType" minOccurs="0" />
                    <xs:element name="directory" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stdout" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stderr" type="pathElementGeneralType" minOccurs="0" />
                </xs:sequence>
                <xs:attribute name="hostname" type="xs:string" />
                <xs:attribute name="queue" type="xs:string" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="cgspGroup" type="cgspProcessType" substitutionGroup="abstractShElement"/>

</xs:schema>
```

A.4.6 EGEE gLite

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="gLiteProcessType">
        <xs:complexContent>
            <xs:extension base="abstractGroupElementType">
                <xs:sequence>
                    <xs:element name="inputData" type="gLiteInputDataType" minOccurs="0" />
                    <xs:element name="rank" type="gLiteRankType" />
                    <xs:element name="stdout" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stderr" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stdin" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="configFile" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="arguments" type="argumentListType" minOccurs="0" />
                </xs:sequence>
                <xs:attribute name="hostname" type="xs:string" />
                <xs:attribute name="virtualOrganisation" type="xs:string" use="required" />
                <xs:attribute name="Type" type="xs:string" default="job" />
                <xs:attribute name="executable" type="xs:string" default="/bin/sh" />
                <xs:attribute name="jobType" type="xs:string" default="normal" />
                <xs:attribute name="nodeNumber" type="xs:string" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <!-gLiteRankType-->
    <xs:complexType name="gLiteRankType">
        <xs:simpleContent>
            <xs:extension base="TextOrVariableType">
                <xs:attribute name="fuzzyrank" type="BooleanOrVariableType" default="false" />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <!-gLiteInputDataType-->
    <xs:complexType name="gLiteInputDataType">
        <xs:attribute name="dataAccessProtocol" type="xs:string" />
        <xs:attribute name="storageIndex" type="xs:string" />
        <xs:attribute name="dataCatalog" type="xs:string" />
    </xs:complexType>
    <!-end of gLiteProcessType -->

    <xs:element name="gLiteGroup" type="gLiteProcessType" substitutionGroup="abstractShElement"/>

</xs:schema>
```

A.4.7 Fura / InnerGrid

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="furaProcessType">
        <xs:complexContent>
            <xs:extension base="abstractGroupSchedulerElementType">
                <xs:sequence>
                    <xs:element name="description" type="xs:string" minOccurs="0" />
                    <xs:element name="stdout" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stderr" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stdin" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="maxTime" type="PosintOrVariableType" minOccurs="0" />
                </xs:sequence>
                <xs:attribute name="jobName" type="xs:string" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="furaGroup" type="furaProcessType" substitutionGroup="abstractShElement"/>

</xs:schema>
```

A.4.8 GLOBUS

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="globusProcessType">
        <xs:complexContent>
            <xs:extension base="abstractGroupSchedulerElementType">
                <xs:sequence>
                    <xs:element name="count" type="PosintOrVariableType" minOccurs="0" />
                    <xs:element name="maxTime" type="PosintOrVariableType" minOccurs="0" />
                    <xs:element name="directory" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stdout" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stderr" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stdin" type="pathElementGeneralType" minOccurs="0" />
                </xs:sequence>
                <xs:attribute name="hostname" type="xs:string" use="required" />
                <xs:attribute name="queue" type="xs:string" use="optional" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:element name="globusGroup" type="globusProcessType" substitutionGroup="abstractShElement"/>
</xs:schema>
```

A.4.9 GridBus

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="gridbusProcessType">
        <xs:complexContent>
            <xs:extension base="abstractGroupSchedulerElementType">
                <xs:sequence>
                    <xs:element name="arguments" type="argumentListType" minOccurs="0" />
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="gridbusGroup" type="gridbusProcessType"
substitutionGroup="abstractShElement"/>

</xs:schema>
```

A.4.10 IBM load leveler

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="loadLevelerProcessType">
        <xs:complexContent>
            <xs:extension base="abstractGroupSchedulerElementType">
                <xs:sequence>
```

```
                    <xs:element name="directory" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stdout" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stderr" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="resources" type="xs:string" minOccurs="0" />
                    <xs:element name="maxTime" type="xs:string" minOccurs="0" />
                    <xs:element name="arguments" type="argumentListType" minOccurs="0" />
                    <xs:element name="taskRepartition" type="LoadLevelerTaskRepartitionType"
minOccurs="1" />
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="LoadLevelerTaskRepartitionType">
            <xs:choice>
                <xs:element name="modeBlock" type="LoadLevelerBlockingGroup" />
                <xs:element name="modeTotalTasks" type="LoadLevelerTotalTasksGroup" />
                <xs:element name="modeTasksPerNode" type="LoadLevelerTasksPerNodeGroup" />
                <xs:element name="modeGeometry" type="LoadLevelerGeometryGroup" />
            </xs:choice>
    </xs:complexType>

    <xs:complexType name="LoadLevelerBlockingGroup">
      <xs:sequence>
            <xs:element name="totalTasks" type="xs:nonNegativeInteger" />
            <xs:element name="blocking" type="xs:nonNegativeInteger" />
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="LoadLevelerTotalTasksGroup">
      <xs:sequence>
            <xs:element name="totalTasks" type="xs:nonNegativeInteger" />
            <xs:element name="node" type="xs:nonNegativeInteger" />
      </xs:sequence>
    </xs:complexType>

    <xs:complexType name="LoadLevelerTasksPerNodeGroup">
      <xs:sequence>
            <xs:element name="tasksPerNode" type="xs:nonNegativeInteger" />
            <xs:element name="node" type="xs:nonNegativeInteger" />
      </xs:sequence>
    </xs:complexType>

   <xs:complexType name="LoadLevelerGeometryGroup">
      <xs:sequence>
            <xs:element name="taskGeometry" type="xs:string" />
      </xs:sequence>
    </xs:complexType>

    <xs:element name="loadLevelerGroup" type="loadLevelerProcessType"
substitutionGroup="abstractShElement" />
</xs:schema>
```

A.4.11 LSF

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd" />

    <xs:complexType name="lsfGroupType">
        <xs:complexContent>
            <xs:extension base="abstractGroupSchedulerElementType">
                <xs:sequence>
                    <xs:element name="resources" type="lsfResourcesType" minOccurs="1" />
                    <xs:element name="stdout" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stderr" type="pathElementGeneralType" minOccurs="0" />
                </xs:sequence>
                <xs:attribute name="interactive" type="BooleanOrVariableType" use="optional" />
                <xs:attribute name="jobName" type="xs:string" use="optional" />
                <xs:attribute name="queue" type="xs:string" use="optional" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="lsfGroup" substitutionGroup="abstractShElement" type="lsfGroupType" />

    <xs:complexType name="lsfResourcesType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="processorNumber" type="PosintOrVariableType" use="optional"/>
                <xs:attribute name="walltime" use="optional">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:pattern value="\d\d(:\d\d)?" />
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
```

A.4.12 Microsoft CCS

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="ccsProcessType">
        <xs:complexContent>
            <xs:extension base="abstractGroupSchedulerElementType">
                <xs:sequence>
                    <xs:element name="resources" type="ccsResourcesType" minOccurs="1" />
                    <xs:element name="stdout" type="pathElementGeneralType" minOccurs="0" />

                    <xs:element name="stderr" type="pathElementGeneralType" minOccurs="0" />
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="ccsGroup" type="ccsProcessType" substitutionGroup="abstractShElement" />

    <xs:complexType name="ccsResourcesType">
        <xs:complexContent>
            <xs:restriction base="xs:anyType">
                <xs:attribute name="cpus" type="PosintOrVariableType"/>
                <xs:attribute name="runtime" type="TimeType" use="optional"/>
            </xs:restriction>
        </xs:complexContent>
    </xs:complexType>

</xs:schema>
```

A.4.13 OAR

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="oarGroupType">
        <xs:complexContent>
            <xs:extension base="abstractGroupSchedulerElementType">
```

```
                <xs:sequence>
                    <xs:element name="resources" type="oarResourcesType" minOccurs="1" />
                    <xs:element name="directory" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stdout" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stderr" type="pathElementGeneralType" minOccurs="0" />
                </xs:sequence>
                <xs:attribute name="interactive" type="BooleanOrVariableType" use="optional" />
                <xs:attribute name="queue" type="xs:string" use="optional" />
                <xs:attribute name="type" use="optional">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:enumeration value="deploy" />
                            <xs:enumeration value="besteffort" />
                            <xs:enumeration value="cosystem" />
                            <xs:enumeration value="checkpoint" />
                            <xs:enumeration value="timesharing" />
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:complexType name="oarResourcesType">
        <xs:simpleContent>
            <xs:extension base="TextOrVariableType">
                <xs:attribute name="nodes"    type="PosintOrVariableType" use="optional"/>
                <xs:attribute name="cpu"      type="PosintOrVariableType" use="optional"/>
                <xs:attribute name="core"     type="PosintOrVariableType" use="optional"/>
                <xs:attribute name="walltime" type="TimeType"             use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>

    <xs:element name="oarGroup" type="oarGroupType" substitutionGroup="abstractShElement" />
</xs:schema>
```

A.4.14 PBS / Torque

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <!-PBSProcess-->
    <xs:complexType name="pbsProcessType">
        <xs:complexContent>
            <xs:extension base="abstractGroupSchedulerElementType">
                <xs:sequence>
                    <xs:element name="resources" type="pbsResourcesType" minOccurs="1" />
                    <xs:element name="mailWhen" type="xs:string" minOccurs="0" />
                    <xs:element name="mailTo" type="xs:string" minOccurs="0" />
                    <xs:element name="joinOutput" type="BooleanOrVariableType" minOccurs="0" />
                    <xs:element name="stdout" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stderr" type="pathElementGeneralType" minOccurs="0" />
                </xs:sequence>
                <xs:attribute name="queue" type="xs:string" use="optional" />
                <xs:attribute name="jobName" type="xs:string" use="optional" />
                <xs:attribute name="interactive" type="xs:boolean" use="optional" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="pbsGroup" type="pbsProcessType" substitutionGroup="abstractShElement" />

    <xs:complexType name="pbsResourcesType">
        <xs:simpleContent>
            <xs:extension base="TextOrVariableType">
                <xs:attribute name="nodes" type="PosintOrVariableType" use="optional"/>
                <xs:attribute name="ppn" type="PosintOrVariableType" use="optional"/>
                <xs:attribute name="walltime" type="TimeType" use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
```

A.4.15 PRUN

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <xs:complexType name="prunProcessType">
        <xs:complexContent>
            <xs:extension base="abstractGroupSchedulerElementType">
                <xs:sequence>
                    <xs:element name="resources" type="prunResourcesType" minOccurs="1" />
                    <xs:element name="stdout" type="pathElementGeneralType" minOccurs="0" />

                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="prunGroup" type="prunProcessType" substitutionGroup="abstractShElement" />

        <xs:complexType name="prunResourcesType">
        <xs:simpleContent>
            <xs:extension base="TextOrVariableType">
                <xs:attribute name="nodes" type="PosintOrVariableType" use="optional"/>
                <xs:attribute name="ppn" type="PosintOrVariableType" use="optional"/>
                <xs:attribute name="walltime" type="TimeType" use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>

</xs:schema>
```

A.4.16 Sun Grid Engine

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:gcm:deployment:1.0"
    xmlns="urn:gcm:deployment:1.0"
    elementFormDefault="qualified">

    <xs:include schemaLocation="../../schema/DeploymentDescriptorSchema.xsd"/>

    <!-gridEngine process -->
    <xs:complexType name="gridEngineProcessType">
        <xs:complexContent>
            <xs:extension base="abstractGroupSchedulerElementType">
                <xs:sequence>
                    <xs:element name="resources" type="gridEngineResourcesType" minOccurs="1"/>
                    <xs:element name="directory" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stdout" type="pathElementGeneralType" minOccurs="0" />
                    <xs:element name="stderr" type="pathElementGeneralType" minOccurs="0" />
                </xs:sequence>
                <xs:attribute name="queue" type="xs:string" use="optional" />
                <xs:attribute name="jobName" type="xs:string" use="optional" />
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="gridEngineGroup" type="gridEngineProcessType"
substitutionGroup="abstractShElement"/>

    <xs:complexType name="gridEngineResourcesType">
        <xs:simpleContent>
```

```
            <xs:extension base="TextOrVariableType">
                <xs:attribute name="nodes" type="PosintOrVariableType" use="optional"/>
                <xs:attribute name="parallelEnvironment" tySpe="TextOrVariableType" use="optional"/>
                <xs:attribute name="walltime" type="TimeType" use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
```

Annex B (informative): Examples of deployment descriptor

B.1 Example 1

```
<?xml version="1.0" encoding="UTF-8"?>
<GCMDeployment
    xmlns="urn:gcm:deployment:1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:gcm:deployment:1.0
http://proactive.inria.fr/schemas/gcm/1.0/ExtensionSchemas.xsd  ">

    <environment>
        <javaPropertyVariable name="user.home"/>
    </environment>

    <resources>
        <group refid="gINRIA">
            <host refid="hSchubby" />
        </group>
    </resources>

    <infrastructure>
        <hosts>
            <host id="hSchubby" os="unix" hostCapacity="1" vmCapacity="1">
                <homeDirectory base="root" relpath="${user.home}" />
            </host>
        </hosts>

        <groups>
            <sshGroup id="gINRIA" hostList="schubby cheypa duff" username="cmathieu" />
            <rshGroup id="grINRIA" hostList="schubby cheypa duff" username="cmathieu" />
        </groups>
    </infrastructure>

</GCMDeployment>
```

B.2 Example 2

```
<?xml version="1.0" encoding="UTF-8"?>
<GCMDeployment
    xmlns="urn:gcm:deployment:1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:gcm:deployment:1.0
http://proactive.inria.fr/schemas/gcm/1.0/ExtensionSchemas.xsd">

    <environment>
        <descriptorVariable name="USERNAME" value="jsmith" />
    </environment>

    <resources>
        <bridge refid="GATEWAY" />

        <group refid="COMPANY_LAN">
            <host refid="BIG_HOST" />
        </group>
    </resources>

    <infrastructure>
        <hosts>
            <host id="BIG_HOST" os="unix" hostCapacity="4"
                vmCapacity="2">
                <homeDirectory base="root"
                    relpath="/user/${USERNAME}/home" />
                <tool id="java" path="/usr/java/j2sdk/bin/java" />
                <tool id="proactive"
                    path="$HOME/myApps/ProActive_3.2.1" />
                <networkInterface name="eth0"/>
            </host>
        </hosts>

        <groups>
            <lsfGroup id="COMPANY_LAN" bookedNodesAccess="rsh" queue="top">
                <resources processorNumber="12" walltime="01:00"/>
                <stdout>output_trace</stdout>
                <stderr>error_trace</stderr>
            </lsfGroup>
        </groups>

        <bridges>
            <sshBridge id="GATEWAY" hostname="sea.inria.fr"
                username="${USERNAME}" commandPath="" />
        </bridges>
    </infrastructure>

</GCMDeployment>
```

Annex C (normative): Custom bridge and group types

The schema is designed to allow easy integration of custom extensions for any new bridge or group protocols. The standard way to do this is by creating a custom XML Schema element type as an extension of the appropriate abstract type abstractBridgeElementType, abstractGroupElementType, or abstractGroupSchedulerElementType.

The type should then be put in its own schema file, which will reference the main schema one.

The following example shows an extension schema with a custom group type and a custom bridge type:

```
<xs:schema
    targetNamespace="urn:gcm:deployment:extension"
    xmlns="urn:gcm:deployment:extension"
    xmlns:gcmdep="urn:gcm:deployment:1.0"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">

    <xs:import
        schemaLocation="DeploymentDescriptorSchema.xsd"
        namespace="urn:gcm:deployment:1.0"/>
    <!-
     custom group
      -->
    <xs:complexType name="myGroupType">
        <xs:complexContent>
            <xs:extension base="gcmdep:abstractGroupElementType">
                <xs:attribute name="someattr" type="xs:string"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="myGroup" substitutionGroup="gcmdep:abstractShElement" type="myGroupType" />

    <!-
     custom bridge
      -->
    <xs:complexType name="myBridgeType">
        <xs:complexContent>
            <xs:extension base="gcmdep:abstractBridgeElementType">
                <xs:attribute name="someattr" type="xs:string"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>

    <xs:element name="myBridge" substitutionGroup="gcmdep:abstractBridgeElement" type="myBridgeType"
/>
</xs:schema>
```

Schema extension example - SampleDeploymentExtension.xsd

The schema can then be used as an auxiliary schema in a descriptor, along with the main schema. Here is a usage example:

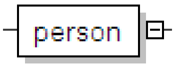
```
<?xml version="1.0" encoding="UTF-8"?>
<GCMDeployment
    xmlns="urn:gcm:deployment:1.0"
    xmlns:pauext="urn:gcm:deployment:extension"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:gcm:deployment:1.0
http://proactive.inria.fr/schemas/gcm/1.0/ExtensionSchemas.xsd
    urn:gcm:deployment:extension SampleDeploymentExtension.xsd
    "
>
    <environment>
        <javaPropertyVariable name="user.home"/>
        <javaPropertyVariable name="user.name"/>
    </environment>

    <resources>
        <group refid="gINRIAEXT">
            <host refid="hSchubby" />
        </group>
    </resources>

    <infrastructure>
        <hosts>
            <host id="hSchubby" os="unix" hostCapacity="1"
                vmCapacity="1">
                <homeDirectory base="root" relpath="${user.home}" />
            </host>
        </hosts>

        <groups>
            <sshGroup id="LAN_INRIA" hostList="cheypa schubby petawawa naruto"
username="${user.name}"
                commandPath="/usr/local/bin/ssh">
            </sshGroup>
            <pauext:myGroup id="gINRIAEXT" someattr="foobar" />
        </groups>

        <bridges>
            <pauext:myBridge id="bINRIAEXT" someattr="somevalue" />
        </bridges>
    </infrastructure>

</GCMDeployment>
```

Schema extension-example - SampleDescriptor.xml

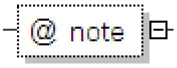
Annex D (informative): Symbols used in the schema diagrams

The following is taken from the <oXygen/> XML Editor User guide (with permission):

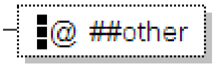
http://www.oxygenxml.com/doc/ug-oxygen/xml-schema-diagram-symbols.html



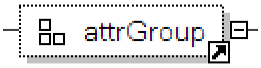
A rectangle containing a string represents an xs:element with the name attribute having the value equal to the string from the rectangle.



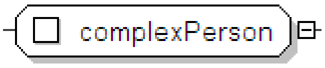
An xs:attribute with the name attribute having the value equal to the string from the rectangle.



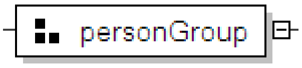
An xs:anyAttribute with the namespace attribute having the value equal to the string from the rectangle.



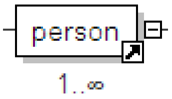
An xs:attributeGroup with the ref attribute having the value equal to the string from the rectangle.



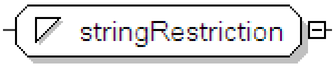
An xs:complexType with the name attribute having the value equal to the string from the rectangle.



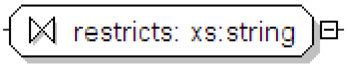
An xs:group with the name attribute having the value equal to the string from the rectangle.



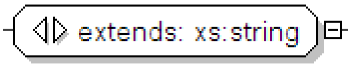
An xs:element with the ref attribute having the value equal to the string from the rectangle.



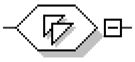
An xs:simpleType with the name attribute having the value equal to the string from the rectangle.



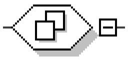
An xs:restriction with the base attribute having the value equal to the string from the rectangle.



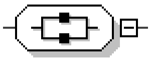
An xs:extension with the base attribute having the value equal to the string from the rectangle.



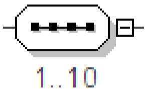
An xs:simpleContent.



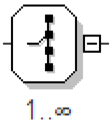
An xs:complexContent.



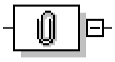
An xs:all.



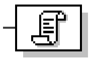
An xs:sequence with the minOccurs and minOccurs attributes specified under the symbol.



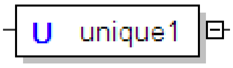
An xs:choice with the minOccurs and minOccurs attributes specified under the symbol.



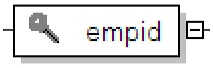
An xs:annotation.



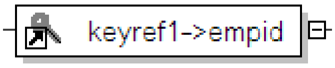
An xs:documentation.



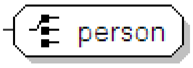
An xs:unique with the name attribute having the value equal to the string from the rectangle.



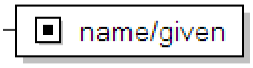
An xs:key with the name attribute having the value equal to the string from the rectangle.



An xs:keyref with the name and refer attributes having the value equal to the string from the rectangle.



An xs:selector with the xpath attribute having the value equal to the string from the rectangle; it is used inside an xs:unique, an xs:key or an xs:keyref.



An xs:field with the xpath attribute having the value equal to the string from the rectangle; it is used inside an xs:unique, an xs:key or an xs:keyref.

Annex E (informative): Bibliography

- ProActive: Programming, Composing, Deploying for the Grid,
  Baude F., Baduel L., Caromel D., Contes A., Huet F., Morel M. and Quilici R., in "GRID COMPUTING: Software Environments and Tools", Jose C. Cunha and Omer F. Rana (Eds), Springer Verlag, January 2006.
  http://proactive.inria.fr/userfiles/file/papers/ProgrammingComposingDeploying.pdf

- CoreGrid Deliverable D.PM.04: "Basic Features of the Grid Component Model (assessed)",
  http://www.coregrid.net/mambo/images/stories/Deliverables/d.pm.04.pdf

- GridCOMP Deliverable D.CFI.03: "Architectural design of the component framework"
  http://gridcomp.ercim.org/images/stories/Deliverables/d.cfi.03-final.pdf

- ETSI Plugtest report: "3rd Grid Plugtest report"
  http://portal.etsi.org/docbox/GRID/Open/GRID_Plugtests_2007/III-GRID-Plugtests-Report-2006.pdf

- "ProActive User Group"
  http://www-sop.inria.fr/oasis/plugtest2007/ProActiveGCMUserGroup.html

- CoreGRID Network of Excellence. http://www.coregrid.net/

History

Document history		
V1.1.1	August 2008	Publication