

JSDL SPMD Application Extension, Version 1.0

Status of This Document

This document provides information to the Grid community on the definition of a parallel (single executable) application extension for JSDL 1.0. Distribution is unlimited.

Copyright Notice

Copyright © Open Grid Forum (2006, 2007). All Rights Reserved.

Abstract

This document specifies the structure and semantics of a parallel application extension to JSDL 1.0. This is a single-program-multiple-data (SPMD) extension. It is limited to a single executable and re-uses a number of elements already defined in the POSIXApplication extension to JSDL 1.0. The document includes the normative XML schema for the extension as well as informative examples.

Contents

Abstract	1
1. Introduction	2
2. Notational Conventions	2
3. Types used by the JSDL SPMD extension	3
4. Parallel environment names.....	3
5. SPMD Application Extension	4
5.1 SPMDApplication Element	4
5.2 NumberOfProcesses Element.....	6
5.3 ProcessesPerHost Element.....	8
5.4 ThreadsPerProcess Element	9
5.5 SPMDVariation Element	10
6. Security Considerations.....	11
7. Contributors	11
8. Intellectual Property Statement.....	12
9. Disclaimer	12
10. Full Copyright Notice	12
11. References	12
Appendix A. SPMD Application Normative Schema.....	13
Appendix B. Extended Informative Examples	15

1. Introduction

This document specifies the structure and semantics of a parallel SPMD (single program multiple data) application extension to JSDL 1.0. This extension is limited to a single executable. It re-uses a number of elements already defined in the POSIXApplication extension to JSDL 1.0[GFD.56].

This specification intentionally defines only a small set of new elements—elements thought to cover many common parallel job descriptions while being supported by a large number of existing systems. This set of elements was selected after a use case session held at GGF17¹ and extended after further discussion at OGF19.

2. Notational Conventions

The key words ‘MUST,’ ‘MUST NOT,’ ‘REQUIRED,’ ‘SHALL,’ ‘SHALL NOT,’ ‘SHOULD,’ ‘SHOULD NOT,’ ‘RECOMMENDED,’ ‘MAY,’ and ‘OPTIONAL’ are to be interpreted as described in RFC 2119 [RFC2119].

Pseudo-schemas are provided for each component, before the description of the component. They use BNF-style conventions for attributes and elements: ‘?’ denotes zero or one occurrences; ‘*’ denotes zero or more occurrences; ‘+’ denotes one or more occurrences. Attributes are conventionally assigned a value which corresponds to their type, as defined in the normative schema.

```
<!-- sample pseudo-schema -->
<defined_element
  required_attribute_of_type_string="xsd:string"
  optional_attribute_of_type_int="xsd:int"? >
  <required_element />
  <optional_element />?
  <one_or_more_of_this_element />+
</defined_element>
```

This specification uses the namespace prefixes in Table 2-1. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table 2-1: Prefixes and namespaces used in this specification.

Prefix	Namespace
xsd	http://www.w3.org/2001/XMLSchema
xsi	http://www.w3.org/2001/XMLSchema-instance
jsdl	http://schemas.ggf.org/jsdl/2005/11/jsdl
jsdl-posix	http://schemas.ggf.org/jsdl/2005/11/jsdl-posix
jsdl-spmc	http://schemas.ogf.org/jsdl/2007/02/jsdl-spmc

The terms *JSDL element* and *JSDL attribute* indicate that the corresponding language construct is represented as an XML element and XML attribute in the normative JSDL schema.

The term *JSDL document* refers to a well formed XML document that can be validated against the normative JSDL Schema definition [GFD.56].

¹ The presentations are available online at http://www.ogf.org/gf/event_schedule/index.php?id=253

The key word *present*, when used to refer to a JSDL element, means that an instance of that element appears in a JSDL document.

The key word *support*, with respect to a JSDL consuming system supporting the JSDL specification and the language constructs, refers to the ability of that system to parse a JSDL document. That is, the consuming system **MUST** be able to interpret the language constructs and assign to them the semantics described in this specification and the values assigned to them in that JSDL document. All JSDL compliant consuming systems must, therefore, support all of the JSDL language constructs.

See also [GFD.56] Notational Conventions.

3. Types used by the JSDL SPMD extension

The following normative XML Schema (xsd) types are used by the JSDL SPMD extension.

Table 3-1 Normative XML Schema Types

XSD types			
xsd:anyURI	xsd:boolean	xsd:positiveInteger	Complex

4. Parallel environment names

The JSDL SPMD extension defines a set of URIs that can be used to refer to a number of parallel environments. The URIs can be used as values of the SPMDVariation element (see §5.5).

Table 4-1 URIs identifying parallel environments

Normative Name	Explanation
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/MPI	Any MPI environment
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/GridMPI	The GridMPI environment ²
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/IntelMPI	The Intel MPI environment ³
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/LAM-MPI	The LAM/MPI environment ⁴
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/MPICH1	The MPICH version 1 environment ⁵
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/MPICH2	The MPICH version 2 environment ⁶
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/MPICH-GM	The MPICH-GM environment ⁷
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/MPICH-MX	The MPICH-MX environment ⁸
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/MVAPICH	The MVAPICH (MPI-1) environment ⁹
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/MVAPICH2	The MVAPICH2 (MPI-2) environment ¹⁰
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/OpenMPI	The Open MPI environment ¹¹
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/POE	The POE (IBM MPI) environment ¹²
http://www.ogf.org/jsdl/2007/02/jsdl-spmc/PVM	A Parallel Virtual Machine environment ¹³

The list is not meant to be exhaustive. Other URIs may be defined and used in addition to the ones in Table 4-1 but they **MUST NOT** be defined in the JSDL SPMD namespace.

² <http://www.gridmpi.org/>

³ <http://www.intel.com/cd/software/products/asm-na/eng/308295.htm>

⁴ <http://www.lam-mpi.org/>

⁵ <http://www-unix.mcs.anl.gov/mpi/mpich1/>

⁶ <http://www-unix.mcs.anl.gov/mpi/mpich2/>

⁷ <http://www.myri.com/scs/download-mpichgm.html>

⁸ <http://www.myri.com/scs/download-mpichmx.html>

⁹ <http://nowlab.cse.ohio-state.edu/projects/mpi-iba/>

¹⁰ <http://nowlab.cse.ohio-state.edu/projects/mpi-iba/>

¹¹ <http://www.open-mpi.org/>

¹² <http://www-03.ibm.com/systems/p/software/pe.html>

¹³ <http://www.csm.ornl.gov/pvm/>

5. SPMD Application Extension

This normative extension defines a schema describing a POSIX single-executable multiple-process application execution.

The namespace prefix used for this schema in the specification is “jsdl-spmd.” The normative namespace for this schema is given in Table 2-1.

5.1 *SPMDApplication Element*

5.1.1 Definition

This element describes an SPMD (single program multiple data) parallel application and its requirements. It re-uses a number of elements already defined in the POSIXApplication extension: Executable, Argument, Input, Output, Error, WorkingDirectory, Environment and UserName. The definitions of these elements are not changed and are not reproduced here, see [GFD.56] §8.1 for details. Normative restrictions, or clarifications, on the usage of these POSIXApplication elements in the context of the SPMDApplication element are included in this section.

In contrast to POSIXApplication the multiplicity of the Executable element is zero or one—it is not required to be present in a JSDL document by this schema. JSDL documents may be processed by any number of intermediaries and the value of this element may not be known until the final submission to an execution environment. The Executable element **MUST** be present at the time of the final submission.

The Executable element **MUST** contain the name of the executable that will be used to start a single process, see §5.2. A wrapper script **MUST NOT** be specified as the value of the Executable element. It is the responsibility of the consuming system to map the SPMDApplication definition to the execution environment. Taking an MPI invocation as an example, the Executable is started using an appropriate MPI execution command (e.g., mpirun) with the argument value for the number of processes taken from the NumberOfProcesses element and the environment of the execution defined by the Environment elements of the SPMDApplication. If the executable is not specified as an absolute path then setting up the environment might also include making sure that the executable can be found by setting an appropriate PATH environment variable.

A parallel application may execute on a number of different hosts. The semantics of WorkingDirectory may imply, but do not require, that the working directory is on a shared filesystem. Implementations may provide the location specified by WorkingDirectory for SPMDApplications on a shared filesystem¹⁴ or, in the case of an MPI environment may provide it as the directory of the rank zero process.

A number of new elements are defined: NumberOfProcesses, ProcessesPerHost, ThreadsPerProcess, and SPMDVariation.

It is worth emphasizing that these SPMDApplication elements describe the parallel application; they do not specify resource requirements. These elements can be specified to depend on resource requirements but the opposite is not true. If the appropriate resource requirements elements (jsdl:TotalResourceCount, jsdl:TotalCPUCount, etc) are not specified the consuming system is free to allocate any number of resources (hosts, CPUs, etc) to the execution potentially resulting in what might appear as, but is not, incorrect behavior. See also §5.3.7.

¹⁴ Errata note: The JSDL Resources FileSystem element is specified to contain a MountSource sub-element which could be used to specify a shared filesystem. Due to an error in the specification this element is missing from the normative JSDL schema.

If the SPMDApplication element is present as a sub-element of the JSDL Application element it MUST appear only once.

5.1.2 Multiplicity

The multiplicity of this element is zero or one in the context of the JSDL Application element.

5.1.3 Type

This is a complex type. It MUST support the following elements from POSIXApplication:

- Executable
- Argument
- Input
- Output
- Error
- WorkingDirectory
- Environment
- UserName

It MUST also support the following elements defined in this specification:

- NumberOfProcesses
- ProcessesPerHost
- ThreadsPerProcess
- SPMDVariation

5.1.4 Attributes

The following attributes are defined:

- name—an optional name for the SPMDApplication element. Its type is xsd:NCName so that it can be reused and referred to from outside the containing JSDL document.

5.1.5 Pseudo Schema

```
<SPMDApplication name="xsd:NCName"?>
  <jSDL-posix:Executable ... />?
  <jSDL-posix:Argument ... />*
  <jSDL-posix:Input ... />?
  <jSDL-posix:Output ... />?
  <jSDL-posix:Error ... />?
  <jSDL-posix:WorkingDirectory ... />?
  <jSDL-posix:Environment ... />*
  <jSDL-posix:UserName ... />?
  <NumberOfProcesses ... />?
  <ProcessesPerHost ... />?
  <ThreadsPerProcess ... />?
```

```

    <SPMDVariation .../>
    <xsd:any##other/>*
  </SPMDApplication>

```

5.1.6 Security Considerations

None.

5.1.7 Examples

A parallel invocation using 'any' MPI environment:

```

<jSDL:Application>
:
  <jSDL-spmd:SPMDApplication>
    <jSDL-posix:Executable>a.out</jSDL-posix:Executable>
    <jSDL-posix:Input>input.dat</jSDL-posix:Input>
    <jSDL-posix:Output>output.dat</jSDL-posix:Output>
    <jSDL-spmd:NumberOfProcesses>8</jSDL-spmd:NumberOfProcesses>
    <jSDL-spmd:SPMDVariation>http://www.ogf.org/jSDL/2007/02/jSDL-
spmd/MPI</jSDL-spmd:SPMDVariation>
  </jSDL-spmd:SPMDApplication>
:
</jSDL:Application>

```

5.2 *NumberOfProcesses Element*

5.2.1 Definition

This element specifies the number of instances of the Executable that the consuming system **MUST** start when starting this parallel application. If this element is not present then it is not defined and the consuming system **MAY** choose any value.

If the `actualTotalCPUCount` attribute is set to true the element **MUST** have the nil value. Setting this element to the nil value under other circumstances is undefined.

It is incorrect to specify both `NumberOfProcesses` with `actualTotalCPUCount` as true and `ThreadsPerProcess` with `actualIndividualCPUCount` as true.

5.2.2 Multiplicity

The multiplicity of this element is zero or one.

5.2.3 Type

The type of this element is `xsd:positiveInteger`.

5.2.4 Attributes

The following attributes are defined:

- `actualTotalCPUCount`—An optional attribute. If true, the value of the total number of CPUs *allocated* to the job is used as the value of the `NumberOfProcesses`.

5.2.5 Pseudo Schema

```

<NumberOfProcesses actualTotalCPUCount="xsd:boolean"? nillable="true">

```

```
xsd:positiveInteger
<NumberOfProcesses/?>
```

5.2.6 Security Considerations

None.

5.2.7 Examples

A parallel application with a total of 8 processes:

```
<jsd1:Application>
:
<jsd1-spmd:SPMDApplication>
  <jsd1-spmd:NumberOfProcesses>8</jds1-spmd:NumberOfProcesses>
  :
</jds1-spmd:SPMDApplication>
:
</jds1:Application>
```

A parallel application with a total of 8 processes that requires 8 CPUs, so each process is expected to have one CPU available:

```
<jsd1:Application>
:
<jsd1-spmd:SPMDApplication>
  <jsd1-spmd:NumberOfProcesses>8</jds1-spmd:NumberOfProcesses>
  :
</jds1-spmd:SPMDApplication>
:
</jds1:Application>
<jsd1:Resources>
  <jsd1:TotalCPUCount>
    <jsd1:Exact>8.0</jds1:Exact>
  </jds1:TotalCPUCount>
</jds1:Resources>
```

A parallel application where the number of processes will be equal to the number of CPUs allocated to it:

```
<jsd1:Application>
:
<jsd1-spmd:SPMDApplication>
  <jsd1-spmd:NumberOfProcesses actualTotalCPUCount="true"
    xsi:nil="true"></jds1-spmd:NumberOfProcesses>
  :
</jds1-spmd:SPMDApplication>
:
</jds1:Application>
```

5.3 ProcessesPerHost Element

5.3.1 Definition

This element specifies the number of instances of the Executable that the consuming system **MUST** start per host. (See also §5.2.) If this element is not present then it is not defined and the consuming system **MAY** choose any value.

A host here refers to a single computational resource, for example, a single node in a cluster. The `jsdl:TotalResourceCount` element, if present, determines the total number of hosts available to the parallel application.

5.3.2 Multiplicity

The multiplicity of this element is zero or one.

5.3.3 Type

The type of this element is `xsd:positiveInteger`.

5.3.4 Attributes

No attributes are defined.

5.3.5 Pseudo Schema

```
<ProcessesPerHost> xsd:positiveInteger </ProcessesPerHost>?
```

5.3.6 Security Considerations

None.

5.3.7 Examples

A parallel application with one process per host. Since `jsdl:TotalResourceCount` is set to 8 a total of 8 processes will be started. Since there is no requirement for the number of CPUs the consuming system is free to make any allocation.

```
<jsdl:Application>
  :
  <jsdl-spmd:SPMDApplication>
    <jsdl-spmd:ProcessesPerHost>1</jsdl-spmd:ProcessesPerHost>
    :
  </jsdl-spmd:SPMDApplication>
</jsdl:Application>
<jsdl:Resources>
  <jsdl:TotalResourceCount>
    <jsdl:Exact>8.0</jsdl:Exact>
  </jsdl:TotalResourceCount>
</jsdl:Resources>
```

The values of the number of processes can be specified to depend on the resource requirements but the opposite is not true. In the following *incorrect* example, even though there is a definition for both the number of processes and number of processes per host, as long as the `jsdl:TotalResourceCount` element is not specified the consuming system is free to allocate any number of hosts to the execution resulting in what might appear as, but is not, incorrect behavior.

SPMDApplication elements describe the parallel application; they do not specify resource requirements.

```
<jSDL:Application>
  :
  <jSDL-spmd:SPMDApplication>
    <jSDL-spmd:NumberOfProcesses>8</jSDL-spmd:NumberOfProcesses>
    <jSDL-spmd:ProcessesPerHost>1</jSDL-spmd:ProcessesPerHost>
    :
  </jSDL-spmd:SPMDApplication>
</jSDL:Application>
<jSDL:Resources>
  :
</jSDL:Resources>
```

5.4 ThreadsPerProcess Element

5.4.1 Definition

This element specifies the number of threads per process (i.e., per instance of the Executable, see §5.2). If this element is not present then it is not defined and the consuming system MAY choose any value.

If the actualIndividualCPUCount attribute is set to true the element MUST have the nil value. Setting this element to the nil value under other circumstances is undefined.

It is incorrect to specify both NumberOfProcesses with actualTotalCPUCount as true and ThreadsPerProcess with actualIndividualCPUCount as true.

5.4.2 Multiplicity

The multiplicity of this element is zero or one.

5.4.3 Type

The type of this element is xsd:positiveInteger.

5.4.4 Attributes

The following attributes are defined:

- actualIndividualCPUCount—An optional attribute. If true, the value of the individual number of CPUs *allocated* to the job on each host is used as the value of the ThreadsPerProcess element.

5.4.5 Pseudo Schema

```
<ThreadsPerProcess actualIndividualCPUCount="xsd:boolean"?
  nillable="true">
  xsd:positiveInteger
</ThreadsPerProcess>?
```

5.4.6 Security Considerations

None.

5.4.7 Examples

A parallel application with 4 processes; 2 threads per process; and one process per host. Since there is no requirement for the number of CPUs the consuming system is free to make any allocation.

```
<jSDL:Application>
  :
  <jSDL-spmd:SPMDApplication>
    <jSDL-spmd:NumberOfProcesses>4</jSDL-spmd:NumberOfProcesses>
    <jSDL-spmd:ProcessesPerHost>1</jSDL-spmd:ProcessesPerHost>
    <jSDL-spmd:ThreadsPerProcess>2</jSDL-spmd:ThreadsPerProcess>
    :
  </jSDL-spmd:SPMDApplication>
</jSDL:Application>
<jSDL:Resources>
  <jSDL:TotalResourceCount>
    <jSDL:Exact>4.0</jSDL:Exact>
  </jSDL:TotalResourceCount>
</jSDL:Resources>
```

A parallel application with one process per host and one thread per CPU:

```
<jSDL:Application>
  :
  <jSDL-spmd:SPMDApplication>
    <jSDL-spmd:ProcessesPerHost>1</jSDL-spmd:ProcessesPerHost>
    <jSDL-spmd:ThreadsPerProcess actualIndividualCPUCount="true">
      </jSDL-spmd:ThreadsPerProcess>
    :
  </jSDL-spmd:SPMDApplication>
</jSDL:Application>
<jSDL:Resources>
  <jSDL:IndividualCPUCount>
    <jSDL:Exact>4.0</jSDL:Exact>
  </jSDL:IndividualCPUCount>
  <jSDL:TotalResourceCount>
    <jSDL:Exact>4.0</jSDL:Exact>
  </jSDL:TotalResourceCount>
</jSDL:Resources>
```

5.5 SPMDVariation Element

5.5.1 Definition

This element defines the type of SPMD application. This element **MUST** be present.

Names for a number of known of environments are defined in Table 4-1.

5.5.2 Multiplicity

The multiplicity of this element is one.

5.5.3 Type

The type of this element is `xsd:anyURI`.

5.5.4 Attributes

No attributes are defined.

5.5.5 Pseudo Schema

```
<SPMDVariation> xsd:anyURI </SPMDVariation>
```

5.5.6 Security Considerations

None.

5.5.7 Examples

A 'stock' MPI application:

```
<jsdl-spmd:SPMDVariation>
  http://www.ogf.org/jSDL/2007/02/jSDL-spmd/MPI
</jsdl-spmd:SPMDVariation>
```

A 'GridMPI' application:

```
<jsdl-spmd:SPMDVariation>
  http://www.ogf.org/jSDL/2007/02/jSDL-spmd/GridMPI
</jsdl-spmd:SPMDVariation>
```

A 'local' MPI variation that may refer to a customized configuration:

```
<jsdl-spmd:SPMDVariation>MPI.122</jsdl-spmd:SPMDVariation>
```

6. Security Considerations

The security considerations of JSDL 1.0 [GFD.56] also apply to this document.

Further security considerations that are specific to elements defined in this specification are detailed in each element section.

7. Contributors

Editors:

Andreas Savva
 IT Architecture Lab.
 Fujitsu Laboratories
 4-1-1, Kamikodanaka, Nakahara, Kawasaki City, Japan
 Email: andreas.savva@jp.fujitsu.com

The following persons have contributed to this specification: Michel Drescher, Donal Fellows, William Lee, Stephen McGough, Stephen Pickles, Kazushige Saga, and Chris Smith.

We are also grateful to numerous colleagues for discussions on the topics covered in this document. Thanks in particular to (in alphabetical order, with apologies to anybody we have missed) Fabio Benedetti, Andre Merzky, Ariel Oleksiak, Ivan Rodero, Ellen Stokes, Kenji Suehiro, Atsuko Takefusa, and Wolfgang Ziegler.

8. Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

9. Disclaimer

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

10. Full Copyright Notice

Copyright (C) Open Grid Forum (2006, 2007). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

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

11. References

[RFC2119] Bradner, S. Key Words for Use in RFCs to Indicate Requirement Levels, RFC 2119, March 1997.

[GFD.56] A. Anjomshoaa, F. Brisard, M. Drescher, D. Fellows, A. Ly, S. McGough, D. Pulsipher, and A. Savva (ed.) Job Submission Description Language (JSDL) Specification, Version 1.0, Global Grid Forum, Lemont, Illinois, U.S.A., GFD.56, November 2005.
<http://www.ogf.org/gf/docs/?final>

Appendix A. SPMD Application Normative Schema

This section contains the full normative XML Schema definition for JSDL SPMD Application 1.0.

For convenience, an informative copy of this schema is also available online at <https://forge.gridforum.org/sf/go/doc13970>. If, however, there are any discrepancies between the definitions in this section and those portions described in other sections in the specification, or in the online schema then the definitions in this section MUST be considered normative.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  The OGF takes no position regarding the validity or scope of any
  intellectual property or other rights that might be claimed to pertain
  to the implementation or use of the technology described in this
  document or the extent to which any license under such rights might or
  might not be available; neither does it represent that it has made any
  effort to identify any such rights. Copies of claims of rights made
  available for publication and any assurances of licenses to be made
  available, or the result of an attempt made to obtain a general license
  or permission for the use of such proprietary rights by implementers or
  users of this specification can be obtained from the OGF Secretariat.

  The OGF invites any interested party to bring to its attention any
  copyrights, patents or patent applications, or other proprietary rights
  which may cover technology that may be required to practice this
  recommendation. Please address the information to the OGF Executive
  Director.

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

  Copyright (C) Open Grid Forum (2006, 2007). All Rights Reserved.

  This document and translations of it may be copied and furnished to
  others, and derivative works that comment on or otherwise explain it or
  assist in its implementation may be prepared, copied, published and
  distributed, in whole or in part, without restriction of any kind,
  provided that the above copyright notice and this paragraph are
  included
  on all such copies and derivative works. However, this document itself
  may not be modified in any way, such as by removing the copyright
  notice
  or references to the OGF or other organizations, except as needed for
  the purpose of developing Grid Recommendations in which case the
  procedures for copyrights defined in the OGF Document process must be
  followed, or as required to translate it into languages other than
  English.

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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:jSDL-posix="http://schemas.ggf.org/jSDL/2005/11/jSDL-posix"
```

```

xmlns="http://schemas.ogf.org/jsdl/2007/02/jsdl-spm�"
xmlns:jsdl-spm�="http://schemas.ogf.org/jsdl/2007/02/jsdl-spm�"
targetNamespace="http://schemas.ogf.org/jsdl/2007/02/jsdl-spm�"
  elementFormDefault="qualified">
  <xsd:import namespace="http://schemas.ogf.org/jsdl/2005/11/jsdl-
posix"/>
  <!--=====-->
  <xsd:complexType name="NumberOfProcesses_Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:positiveInteger">
        <xsd:attribute name="actualtotalcpucount"
          type="xsd:boolean" use="optional"/>
        <xsd:anyAttribute namespace="##other"
          processContents="lax"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <!--=====-->
  <xsd:complexType name="ProcessesPerHost_Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:positiveInteger">
        <xsd:anyAttribute namespace="##other"
          processContents="lax"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <!--=====-->
  <xsd:complexType name="ThreadsPerProcess_Type">
    <xsd:simpleContent>
      <xsd:extension base="xsd:positiveInteger">
        <xsd:attribute name="actualindividualcpucount"
          type="xsd:boolean" use="optional"/>
        <xsd:anyAttribute namespace="##other"
          processContents="lax"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <!--=====-->
  <xsd:complexType name="SPMDApplication_Type">
    <xsd:sequence>
      <xsd:element ref="jsdl-posix:Executable" minOccurs="0"/>
      <xsd:element ref="jsdl-posix:Argument" minOccurs="0"
        maxOccurs="unbounded"/>
      <xsd:element ref="jsdl-posix:Input" minOccurs="0"/>
      <xsd:element ref="jsdl-posix:Output" minOccurs="0"/>
      <xsd:element ref="jsdl-posix:Error" minOccurs="0"/>
      <xsd:element ref="jsdl-posix:WorkingDirectory" minOccurs="0"/>
      <xsd:element ref="jsdl-posix:Environment" minOccurs="0"
        maxOccurs="unbounded"/>
      <xsd:element ref="jsdl-posix:UserName" minOccurs="0"/>
      <xsd:element ref="jsdl-spm�:NumberOfProcesses" minOccurs="0"/>
      <xsd:element ref="jsdl-spm�:ProcessesPerHost" minOccurs="0"/>
      <xsd:element ref="jsdl-spm�:ThreadsPerProcess" minOccurs="0"/>
      <xsd:element ref="jsdl-spm�:SPMDVariation" minOccurs="1"
        maxOccurs="1"/>
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

```

```

    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:NCName" use="optional"/>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>
  <!--=====-->
  <xsd:element name="SPMDApplication" type="SPMDApplication_Type"/>
  <xsd:element name="NumberOfProcesses"
    type="jsdl-spmd:NumberOfProcesses_Type" nillable="true"/>
  <xsd:element name="ProcessesPerHost"
    type="jsdl-spmd:ProcessesPerHost_Type"/>
  <xsd:element name="ThreadsPerProcess"
    type="jsdl-spmd:ThreadsPerProcess_Type" nillable="true"/>
  <xsd:element name="SPMDVariation" type="xsd:anyURI"/>
</xsd:schema>

```

Appendix B. Extended Informative Examples

The following example defines an invocation of the application “XXX.” The example uses all JSDL SPMD elements.

```

<?xml version="1.0" encoding="UTF-8"?>
<jSDL:JobDefinition
  xmlns:jsdl="http://schemas.ggf.org/jsdl/2005/11/jsdl"
  xmlns:jsdl-posix="http://schemas.ggf.org/jsdl/2005/11/jsdl-posix"
  xmlns:jsdl-spmd="http://schemas.ggf.org/jsdl/2007/02/jsdl-spmd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <jSDL:JobDescription>
    <jSDL:JobIdentification>
      <jSDL:JobName>My Parallel Application</jSDL:JobName>
    </jSDL:JobIdentification>
    <jSDL:Application>
      <jSDL:ApplicationName>mympiapp</jSDL:ApplicationName>
      <jSDL-spmd:SPMDApplication>
        <jSDL-posix:Executable>a.out</jSDL-posix:Executable>
        <jSDL-posix:Argument>control.txt</jSDL-posix:Argument>
        <jSDL-posix:Input>input.dat</jSDL-posix:Input>
        <jSDL-posix:Output>output.dat</jSDL-posix:Output>
        <jSDL-spmd:NumberOfProcesses>4
        </jSDL-spmd:NumberOfProcesses>
        <jSDL-spmd:ProcessesPerHost>2
        </jSDL-spmd:ProcessesPerHost>
        <jSDL-spmd:SPMDVariation>
          http://www.ogf.org/jsdl/2007/02/jsdl-spmd/MPI
        </jSDL-spmd:SPMDVariation>
      </jSDL-spmd:SPMDApplication>
    </jSDL:Application>
    <jSDL:Resources>
      <jSDL:IndividualCPUCount>
        <jSDL:Exact>2.0</jSDL:Exact>
      </jSDL:IndividualCPUCount>
      <jSDL:TotalResourceCount>
        <jSDL:Exact>2.0</jSDL:Exact>
      </jSDL:TotalResourceCount>
    </jSDL:Resources>
  </jSDL:JobDescription>
</jSDL:JobDefinition>

```