014 Specification (gfd.149.pdf)

Excerpt by Morris Riedel

BEFORE

Author: M. Drescher, A. Anjomshoaa, G. Williams

Title: JSDL Parameter Sweep Job Extension

Source: Open Grid Forum, Grid Final Document (GFD)

Date of excerpt: 2009-05-19

My experience in advance:

- Co-Chair GIN & PGI group

Which questions should the text answer?

1: What elements are covered?

2: Is the standard applicable to real Grid middleware in production?

3: What is the benefit from the standard?

DURING

The most important notes:

- "The Function element is the source of the values that are assigned to Parameters in a Parameter Sweep, as described below."
- "Like the Function element, the Parameter element is defined as an abstract XML Schema element."
- "The Assignment element associates one Function substituent with one or more Parameter substituents."
- "The Sweep element describes how to modify an existing JSDL Job Template in order to get a Parameter Sweep JSDL Job Template."
- "The DocumentNode element selects the contents of an XML element or attribute in the enclosing JSDL Job Template as the target for Parameter Sweep Function values."
- "The FileSweep element defines the syntax and semantics for Paraemeter Sweeps that require fixed input files with incrementally changing contents."
- "The Values Function provides an ordered list of otherwise unconnected, or independent values."
- "The LoopInteger function provides an ordered list of integer values within the inclusive range specified by the start and end values."

Remarks:

A useful specification for one specific purpose related to parameter jobs.

Something very special:

- “The syntax and semantics defined in this document provide an alternative to explicitly submitting thousands of individual JSDL submissions of the same base job template, except with a different set of parameters for each.”

AFTER

What are the answers to the questions above?

1: Parameter sweeps in job submissions described via JSDL documents.

2: The standard is applicable and useful for certain scientific applications.

3: A precise definition how parameter sweep jobs can be defined in a well-structured manner.

Which kind of approach?

New XML elements define Functions, Parameters and both are linked with Assignments. All of these elements can be well embedded in the existing JSDL schemas.

Interesting Publications to follow?

None

OVERALL EVALUATION:

Interesting standard that focuses on one particular issue in the job submission area related to the execution of executables with different parameter sets.

Cites	***Keywords***	***Pg.***
Abstract		1
"The syntax and semantics defined in this document provide an alternative to explicitly submitting thousands of individual JSDL submissions of the same base job template, except with a different set of parameters for each."	Different set of parameters for same base job	1
1 Introduction		3
"A Parameter Sweep is a job that internally defines a collection of jobs."	Collection of jobs	3
"For each of the jobs in the collection, the value of one or more of the job parameters may be changed in some preordained fashion, from the previous job in the collection."	Value of job parameters change	3
"Hence, an array of values may be assigned to a parameter whose value is to be changed to successive elements in that array, for each consecutive job in the collection."	Array of values assigned to a parameter	3
"In the definition of a Parameter Sweep for submission of a collection of jobs, the goal is to capture the array of values for each parameter, whose value is to be changed, in a succinct and systematic manner."	Definition in a succinct and systematic manner	3
"This document does not define how to instantiate and manage the described Parameter Sweep, which may be left as an implementation detail, or defined in Profiles on other specifications that deal with processing Job Submissions in Grids."	Not define how to instantiate and manage the sweeps	3
"Specifically, each member of an element's [children] or [attributes] properties, is described using an XPath-like notation."	Xpath	3
"Pseudo-schemas are provided for each component, before the description of the component. They use BNF-style conventions for attributes and elements:"	Pseudo-schemas, BNF-style	3
2 Parameter Sweep		4
"This section defines the core information elements of the Parameter Sweep extension to JSDL 1.0."	Core information elements	4
2.1 Function		
"The Function element is the source of the values that are assigned to Parameters in a Parameter Sweep, as described below."	Function element	4
"A Function is not necessarily a function in a mathematical sense."	Not necessarily a mathematical function	4
"Rather, it is better described as a mathematical sequence."	Mathematical sequence	4
"A Function yields a finite set of values"	Finite set of values	4
"A Function has a first value"	First value	4
"A Function has a last value"	Last value	4
"A Function has concepts of current and next values; and"	Current and next values	4
"A Function yields the same parameter values in the same order every time it is used from the beginning, which is defined by its initial conditions."	Same order every time	4
"The cardinality of a Function is defined as the number of values it yields."	Cardinality	4
"The Function element is defined as an abstract XML Schema element."	Abstract schema element	4
"Section 4 defines standard Function substituent's that MUST be supported by any implementation of this specification."	Standard Function substituent's	4
"Concrete Function substituents implicitly define the data type of its values."	Define data type of its values	4
"The data type of a Function substituent in a XML document MUST	Matching data types	4

match the data type of the associated Parameter substituent."		
2.2 Parameter		5
"Like the Function element, the Parameter element is defined as an abstract XML Schema element."	Abstract Schema element	5
"Section 3 defines standard Parameter substituent's that MUST be supported by any implementation of this specification."	Standard Parameter Substituent's	5
"Concrete Parameter substituent's define how exactly a value is applied when evaluating the Parameter Sweep."	Define how exactly a value is applied	5
"Parameter substituents implicitly define a data type."	Data type definition	5
"The Parameter substituent's data type MUST match the data type of the associated Function substituent."	Matching data types	5
2.3 Assignment		5
"The Assignment element associates one Function substituent with one or more Parameter substituents."	Associates Function with Parameters	5
"Each value yielded by a Function substituent MUST be assigned or applied to each associated Parameter substituent in due course during the evaluation of the Parameter Sweep."	Assignment idea	5
"For example, if a Function substituent yields five values, then each of the five values must be applied to the subsequent five JSDL job submissions."	Example	5
"For evaluation and validation purposes the Assignment element inherits the properties of its child Function element…"	Evaluation, validation	5
"The cardinality of the Assignment element is defined as the number of value assignments it yields."	Cardinality	5
2.4 Sweep		7
"The Sweep element describes how to modify an existing JSDL Job Template in order to get a Parameter Sweep JSDL Job Template."	Sweep element, modify existing job template	7
"The Sweep element defines the coordination of Assignment element evaluation."	Coordination of Assignment element evaluation	7
"The Sweep element MUST contain at least one Assignment element."	One Assignment element	7
"Sibling Assignment child elements MUST have the same cardinality."	Sibling assignment child elements	7
"Hence the Sweep element inherits the properties of its child Assignment elements…"	Inherits properties	7
"The cardinality of the Sweep element is defined as the number of Value Assignment Sets it yields."	Cardinality of sweep elements	7
"Each Value Assignment Set MUST be applied at once; the Value Assignment Set is applied to a copy of the original JSDL Job Template."	Value Assignment Set	7
"The thus modified job template represents an individual job from the collection of jobs defined by the Parameter Sweep, and concludes an iteration over a Sweep element."	Individual job	7
"A sweep element may contain nested Sweep elements."	Nested Sweep elements	7
"The evaluation of nested Sweep elements slightly changes the evaluation rules."	Evaluation rules slightly changed	7
3. Normative Parameter substituent definitions		8
"This section defines normative Parameter substituent's that MAY appear in a Parameter Sweep."	Normative Parameter substituent's	8
3.1 DocumentNode Parameter substituent		8
"The DocumentNode element selects the contents of an XML element or attribute in the enclosing JSDL Job Template as the target for Parameter Sweep Function values."	DocumentNode Parameter	8
"The type of the DocumentNode's Match element is an Xpath	DocumentNode Xpath	8

“While this is mostly for historical reasons, it is valid to treat the jsdl:JobDescription element as the real JSDL Job Template defining the nuts and bolts of the job that needs to be executed, and use the jsdl:JobDefinition element as a scoping closure to anything a JSDL Job Template is composed with.”	JobDescription, JobDefinition ideas	20
“Hence, the most appropriate way of composing a JSDL Job Template with a Parameter Sweep definition is to mix-in the Parameter Sweep as a direct child element of the jsdl:JobDefinition”	Min-in as a child of JobDefinition	20
6 Examples		20
“All examples given in this section use the same JSDL Job template”	Same JSDL job template	20
7 FileSweep Examples		32
See example below		
8 Security Considerations		39
“It does not define any security related semantics such as format and possible contents of security tokens.”	No security defined	39
“The only consideration is that it is possible to use this mechanism to generate a very large number of jobs; implementations should take care to manage any intermediate storage of generated documents carefully so that processing a Parameter Sweep does not act as a denial-of-service attack on the processing system.”	Denial-of-service attack	39

EXAMPLE

```
<sweep:Assignment>
   <sweep:DocumentNode>
      <sweep:NamespaceBinding
            ns="http://schemas.ggf.org/jsdl/2005/11/jsdl-posix"
            prefix="jsdl-posix" />
      <sweep:Match>
         /*//jsdl-posix:POSIXApplication[1]/jsdl-posix:Argument[2]
      </sweep:Match>
   </sweep:DocumentNode>
   <sweep:DocumentNode>
      <sweep:NamespaceBinding
            ns="http://schemas.ggf.org/jsdl/2005/11/jsdl-posix"
            prefix="jsdl-posix" />
      <sweep:Match>
         /*//jsdl-posix:POSIXApplication[1]/jsdl-posix:Argument[4]
      </sweep:Match>
   </sweep:DocumentNode>
   <sweepfunc:Values>
      <sweepfunc:Value>foo</sweepfunc:Value>
      <sweepfunc:Value>bar</sweepfunc:Value>
      <sweepfunc:Value>baz</sweepfunc:Value>
   </sweepfunc:Values>
</sweep:Assignment>
```

EXAMPLE

```
<sweep:Sweep>
   <sweep:Assignment>
      <sweep:DocumentNode>
         <sweep:NamespaceBinding
               ns="http://schemas.ggf.org/jsdl/2005/11/jsdl-posix"
               prefix="jsdl-posix" />
         <sweep:Match>
            /*//jsdl-posix:POSIXApplication[1]/jsdl-posix:Argument[2]
         </sweep:Match>
      </sweep:DocumentNode>
      <sweepfunc:LoopInteger start="1" end="10" />
   </sweep:Assignment>
</sweep:Sweep>
```

In this example, one *Parameter* is associated with one *Function*: The second *jsdl-posix:Argument* of the first *jsdl-posix:POSIXApplication* element found at an arbitrary depth in the JSDL Template document being addressed, will receive the values 1, 2, 3, 4, etc. up to 10, constituting a collection of jobs defined by this [parameter sweep] definition with 10 individual jobs.

EXAMPLE: DocumentNode

```
<jsdl:JobDefinition>
   <jsdl:JobDescription>
      <jsdl:Application>
         <jsdl-posix:POSIXApplication>
            <jsdl-posix:Executable>
               /bin/some_exe
            </jsdl-posix:Executable>
            <jsdl-posix:Argument>-infile</jsdl-posix:Argument>
            <jsdl-posix:Argument>in.NNN.dat</jsdl-posix:Argument>
         </jsdl-posix:POSIXApplication>
      </jsdl:Application>
   </jsdl:JobDescription>
   <sweep:Sweep>
      <sweep:Assignment>
         <sweep:DocumentNode>
            <sweep:NamespaceBinding
                  ns="http://schemas.ggf.org/jsdl/2005/11/jsdl-posix"
                  prefix="jsdl-posix" />
            <sweep:Match>
               substring(/*//jsdl-posix:Argument[2], 4, 3)
            </sweep:Match>
         </sweep:DocumentNode>
         <sweepfunc:Values>
            <sweepfunc:Value>001</sweepfunc:Value>
            <sweepfunc:Value>002</sweepfunc:Value>
            <sweepfunc:Value>003</sweepfunc:Value>
            <sweepfunc:Value>004</sweepfunc:Value>
            <sweepfunc:Value>005</sweepfunc:Value>
         </sweepfunc:Values>
      </sweep:Assignment>
   </sweep:Sweep>
</jsdl:JobDefinition>
```

The above yields the following 5 jobs:

```
/bin/some_exe -infile in.001.dat
/bin/some_exe -infile in.002.dat
/bin/some_exe -infile in.003.dat
/bin/some_exe -infile in.004.dat
/bin/some_exe -infile in.005.dat
```

EXAMPLE: FileSweep TemplateFile

```
<file-sweep:TemplateFile>
   <jsdl:FileName>data1.dat</jsdl:FileName>
</file-sweep:TemplateFile>
```

In this example, the template file 'data1.dat' is relative to the working job directory as determined by the consuming system.

EXAMPLE: Values Function

```
<sweepfunc:Values>
   <sweepfunc:Value> The </sweepFunc:Value>
   <sweepfunc:Value> quick </sweepFunc:Value>
   <sweepfunc:Value> brown </sweepFunc:Value>
   <sweepfunc:Value> fox </sweepFunc:Value>
   <sweepfunc:Value> jumps </sweepFunc:Value>
   <sweepfunc:Value> over </sweepFunc:Value>
   <sweepfunc:Value> the </sweepFunc:Value>
   <sweepfunc:Value> lazy </sweepFunc:Value>
   <sweepfunc:Value> dog </sweepFunc:Value>
<sweepfunc:Values>
```

In this example, the *Values* function yields nine values, which, if concatenated using whitespace, would form the string "The quick brown fox jumps over the lazy dog".

EXAMPLE: LoopInteger Function

```
<sweepfunc:LoopInteger start="1" end="10">
   <sweepfunc:Exception> 5 </sweepfunc:Exception>
   <sweepfunc:Exception> 7 </sweepfunc:Exception>
</sweepfunc:LoopInteger>
```

In this example, the *LoopInteger* function yields eight values: “1”, “2”, “3”, “4”, “6”, “8”, “9”, “10”.

EXAMPLE: LoopDouble Function

```
<sweepfunc:LoopDouble start="1000.0" end="1400.0" step="50.0">
   <sweepfunc:Exception>1000.0</sweepfunc:Exception>
   <sweepfunc:Exception>1100.0</sweepfunc:Exception>
</sweepfunc:LoopDouble>
```

In this example, the *LoopDouble* function yields the seven values "1050.0", "1150.0", "1200.0", "1250.0", "1300.0", "1350.0", "1400.0".

EXAMPLE: JSDL with Sweeps

```
<?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:sweep="http://schemas.ogf.org/jsdl/2009/03/sweep"
      xmlns:sweepfunc=
            "http://schemas.ogf.org/jsdl/2009/03/sweep/functions">

   <jsdl:JobDescription>
      <jsdl:Application>
         <jsdl-posix:POSIXApplication>
            <jsdl-posix:Executable>/bin/echo</jsdl-posix:Executable>
            <jsdl-posix:Argument>The</jsdl-posix:Argument>
            <jsdl-posix:Argument>quick</jsdl-posix:Argument>
            <jsdl-posix:Argument>brown</jsdl-posix:Argument>
            <jsdl-posix:Argument>fox</jsdl-posix:Argument>
            <jsdl-posix:Argument>jumps</jsdl-posix:Argument>
            <jsdl-posix:Argument>over</jsdl-posix:Argument>
            <jsdl-posix:Argument>the</jsdl-posix:Argument>
            <jsdl-posix:Argument>lazy</jsdl-posix:Argument>
            <jsdl-posix:Argument>dog</jsdl-posix:Argument>
         </jsdl-posix:POSIXApplication>
      </jsdl:Application>
   </jsdl:JobDescription>

   <sweep:Sweep>
      <sweep:Assignment>
         <sweep:DocumentNode>
            <sweep:NamespaceBinding
                  ns="http://schemas.ggf.org/jsdl/2005/11/jsdl-posix"
                  prefix="jsdl-posix" />
            <sweep:Match>
               //jsdl-posix:Argument[4]
            </sweep:Match>
         </sweep:DocumentNode>
         <sweepfunc:Values>
            <sweepfunc:Value>cat</sweepfunc:Value>
            <sweepfunc:Value>dog</sweepfunc:Value>
            <sweepfunc:Value>bird</sweepfunc:Value>
         </sweepfunc:Values>
      </sweep:Assignment>
   </sweep:Sweep>
</jsdl:JobDefinition>
```

JSDL Integration

```
<jsdl:JobDefinition>
    <jsdl:JobDescription/>
    <sweep:Sweep/>+
<jsdl:JobDefinition>
```