



# Workflow Language

## 1. Structure

In order to allow users to describe tasks and their dependencies, we defined a simple and flexible XML-based workflow language (xWFL). The workflow language provides the means to build new applications by linking standalone applications.

Our workflow description expressed in XML format, and its structure is shown in Figure 1. It consists of three parts, namely parameter definitions, task definitions and data link definitions.

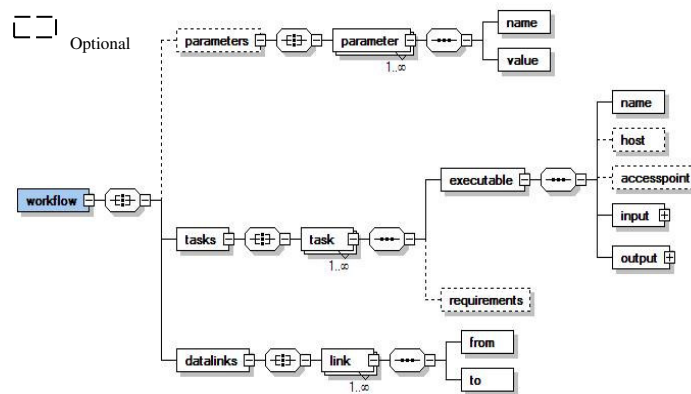


Figure 1. The Structure of Workflow Language.

## 2. Parameters

Supporting parameterization in the workflow language is very important for scientific applications. It enables scientists to do experiments on different parameters easily by changing the value of parameters without being concerned about the detailed workflow description. Multiple parameters types such as single, range, select, random, file and multi-files need to be supported. An example for a single parameter type and a range parameter type is given below:

```
<parameters>
  <para type= "single">
    <name>X</name>
    <value type=integer>10</value>
  </para>
  <para type= "range">
    <name>Y</name>
    <min>1</min>
    <max>20</max>
    <step>2</step>
  </para>
</parameters>
```

## 3. Task Definition

Basically, <tasks> is a set of tasks that are to be executed. Our workflow language supports both abstract workflow and concrete workflow. As shown in the Figure 1, host and accesspoint are optional. That is the user or higher workflow planner, can either specify the location of a particular resource providing required application services or leave it to the engine to identify their providers dynamically at run-time. In the below example, task A executes dock.exe program on host *services.gridbus.com* in

the directory `/services` and executable `dock` has two input I/O ports: port0 (a file) and port1 (a parameter value). The example shows task A only has one output named port2.

```

<tasks>
  <task name= "A">
    <executable>
      <name>dock</name>
      <host>services.gridbus.com</host>
      <accesspoint type= "GlobusGram">/dock.exe</accesspoint>
      <input>
        <port0 type= "file" url= "http://www.gridbus.org/
          datacenter/dock.in">dock.in</port0>
        <port1 type= "msg">$time_base_value</port1>
      </input>
      <output>
        <port2 type= "file">dock.out</port2>
      </output>
    </executable>
    .....
  </task>
</tasks>

```

#### 4. Data Links

Data link is used to specify the data flow of the tasks. Below example is the data flow description of Figure 3. The inputs of task B and task C rely on the output of A. The task A's output needs to be transferred to the node on which task B and task C are executed. Input could be a file, parameter value or data stream.

```

<workflow>
  <tasks>
    <task name= "A">
      .....
    </task>
    <task name= "B">
      .....
    </task>
    <task name= "C">
      .....
    </task>
    <task name= "D">
      .....
    </task>
  </tasks>
  <links>
    <link>
      <from>A:port2</from>
      <to>B:port0</to>
    </link>
    <link>
      <from>A:port2</from>
      <to>C:port0</to>
    </link>
    <link>
      <from>B:port1</from>
      <to>D:port0</to>
    </link>
    <link>
      <from>C:port2</from>
      <to>D:port1</to>
    </link>
  </links>
</workflow>

```

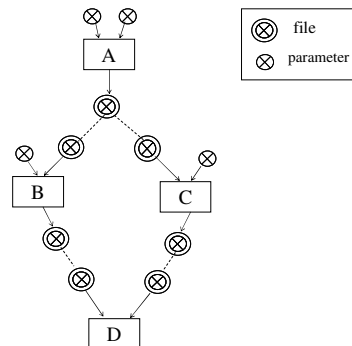


Figure 2. Flow Diagram of Task A, B, C and D.

#### 5. Example

Entire workflow description for Figure 2 is presented in Figure 3. In this example, task A and task B are specified to submit to `belle.cs.mu.oz.au` and `belle.anu.edu.au` respectively. The locations of servers to run the rest tasks will be discovered by workflow engine at the run-time via Grid Market Directory ([www.gridbus.org/gmd](http://www.gridbus.org/gmd)).

```

<workflow>
  <tasks>
    <task name="A">
      <executable>
        <name>xcalc</name>
        <host>belle.cs.mu.oz.au</host>
        <accesspoint type="GT2Gram"/>/home/bellegrid/workflowengine/calc/calc</accesspoint>
        <input>
          <port0 type="msg">3</port0>
          <port1 type="msg">0.1</port1>
        </input>
        <output>
          <port2 type="file">output</port2>
        </output>
      </executable>
    </task>
    <task name="B">
      <executable>
        <name>ycalc</name>
        <host>belle.anu.edu.au</host>
        <accesspoint type="GT2Gram"/>/data/workflowengine/ycalc/ycalc.sh</accesspoint>
        <input>
          <port0 type="file">para</port0>
          <port1 type="msg">0.5</port1>
        </input>
        <output>
          <port2 type="file">output</port2>
        </output>
      </executable>
    </task>
    <task name="C">
      <executable>
        <name>ycalc</name>
        <input>
          <port0 type="file">para</port0>
          <port1 type="msg">1</port1>
        </input>
        <output>
          <port2 type="file">output</port2>
        </output>
      </executable>
    </task>
    <task name="D">
      <executable>
        <name>addcalc</name>
        <input>
          <port0 type="file">para1</port0>
          <port1 type="file">para2</port1>
        </input>
        <output>
          <port2 type="file">output</port2>
        </output>
      </executable>
    </task>
  </tasks>
  <links>
    <link>
      <from>A:port2</from>
      <to>B:port0</to>
    </link>
    <link>
      <from>A:port2</from>
      <to>C:port0</to>
    </link>
    <link>
      <from>B:port2</from>
      <to>D:port0</to>
    </link>
    <link>
      <from>C:port2</from>
      <to>D:port1</to>
    </link>
  </links>
</workflow>

```