Collaborative Climate Community Data and Processing Grid (C3Grid)

Generation 0:
Workflow Scheduler Web Service Definition

Work Package:	WP6 - Coordinated Grid Scheduling
Author(s):	Christian Grimme, Alexander Papaspyrou
Version:	C3Grid-T6.1-001-Final
Publishing Date:	2006-10-15
WP-Coordination:	Robotics Research Institute, Dortmund University
Partners:	ZIB, GKSS, Uni Köln, FUB
Contact:	Christian Grimme, Alexander Papaspyrou
Email:	{firstname.surname}@udo.edu

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Contents

1 Introduction

1.1 About this Document

In this document a Web Service definition for the C3Grid Workflow Scheduling Service (WSS) is presented. The interfaces in this paper are designed for two main purposes:

1. Passing tasks or jobs to the C3Grid Scheduler
2. Monitoring job and workflow states in the Grid

However, these main aspects can be specified in more detail which will be done in the following sections together with the explanation of the Web Service definition.

Note: Those elements in this definition which include the string 'job' in their name refer to workflows of interdependend tasks whereas those elements with the string 'task' in their name refer to single, atomical execution entities.

1.2 General Comments on the Workflow Scheduling Service

The Workflow Scheduling Service (WSS) is a central component in the C3Grid architecture. Its main purpose is to accept and handle atomic tasks or jobs (tasks with interdependencies) form the user portal and coordinate their efficient execution in the Grid. To this end, the WSS has to provide mechanisms to orchestrate task execution on Grid resources as well as planning data access in cooperation with the C3Grid Data Management System (DMS). Figure 1.1 exemplarily depicts the described scenario: the user submits an atomic task or a job through a portal. To describe tasks for submission to the WSS the C3Grid Community agreed to use the Job Specification and Description Language (JSDL) standardized by the OGF [JSDL05]. As this specification does not support the annotation of task dependencies to realize jobs (workflows), a proprietary Workflow Specification Language (WSL) was developed to support basic dependencies. However, in WSL cyclic or conditional dependencies are not allowed [WSL06]. The portal passes the job to the *Portal interface* of the Scheduler (WSS) that coordinates the job's execution. Simultanously, the Portal interface provides functionality to monitor and administer the before submitted job.

The focus of this document is to specify the interface marked with G in Figure 1.1 allowing the portal to submit, control, and monitor tasks or jobs. As such, two port types (WorkflowControlPort and WorkflowMonitoringPort) were defined to logically separate the control from monitoring functionality. A survey of the complete functionality of the WSS is given in Figure 1.2 and will be detailed lateron.

The remainder of this document is organized as follows: in the next Section the fundamental namespaces and types used in this definition are presented. Then in Section 2.2 the WorkflowControlPort definition is detailed. Afterwards Section 2.3 explains the iterface for monitoring the execution in the Grid. Finally, Section 2.4 describes the enumeration types used as status messages.

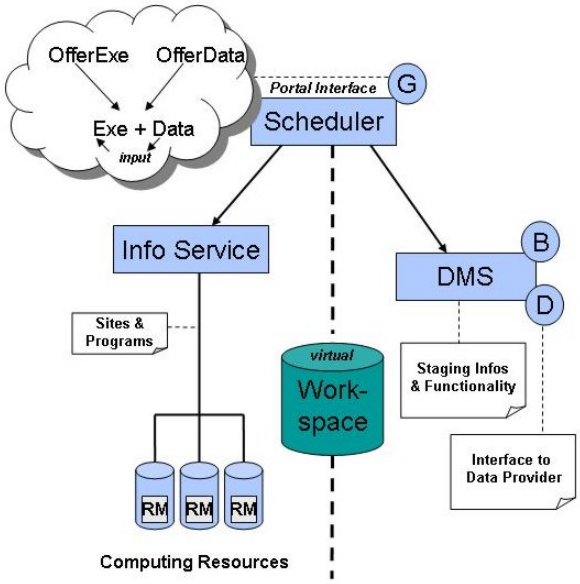

Figure 1.1: Schematic depiction for the collaboration scenario of Portal, WSS and DMS

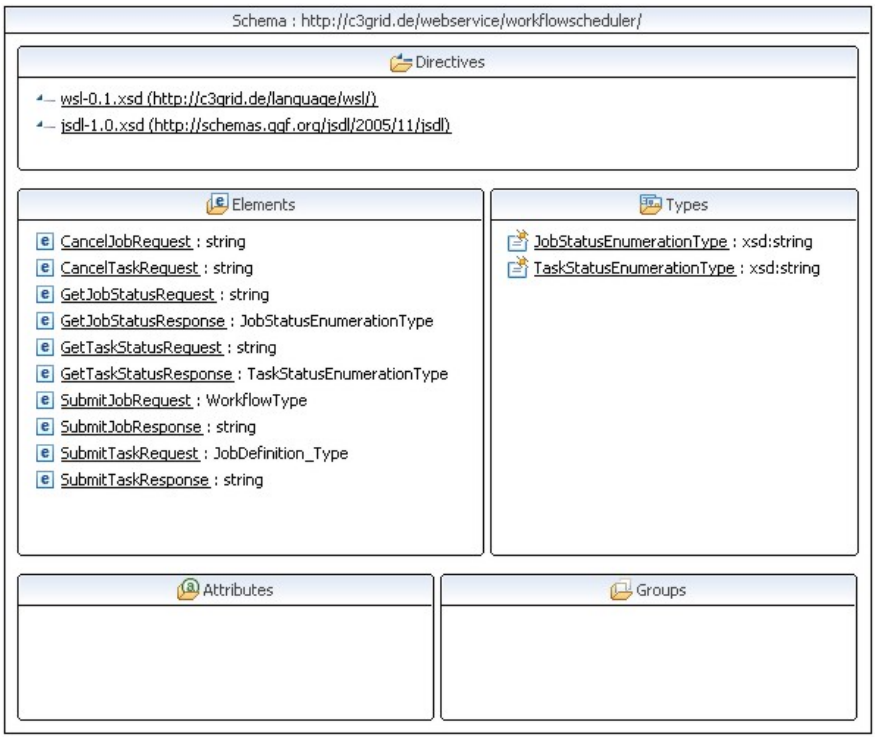

Figure 1.2: Summarizing depiction of the WSS functionality.

2 Specification Details

2.1 Types

Table 2.1 contains the namespaces which are used for the definition of this service, while Table 2.2 shows all types used in this definition. As these types are identically used in the WorkflowMonitoring and WorkflowControl port we will describe them here only once.

Tag	Namespace
wft	`http://c3grid.de/webservice/workflowscheduler/`
wsl	`http://www.c3grid.de/language/wsl`
jsdl	`http://schemas.ggf.org/jsdl/2005/11/jsdl`

Table 2.1: Namespaces used in the definition.

Type	Description	Namespace
JobStatusEnumerationType	Defines the states a job can reach during execution. The different states are detailes in Section 2.4	wft
TaskStatusEnumerationType	Defines those states a task can reach during its execution. The detailed description can be found in Section 2.4.	wft
WorkflowType	This is an external type from the WSL job definition (job with interdependent tasks).	wsl
JobDefinitionType	Type from the JSDL definition for defining an atomic task.	jsdl

Table 2.2: Types defined and used in this definition.

2.2 WorkflowControlPort

The WorkflowControl definition of the WSS provides several operations to start and control job or task execution. These operations and the types employed as well as the return values are schematically depicted in Figure 2.1 and detailed in the following paragraphs.

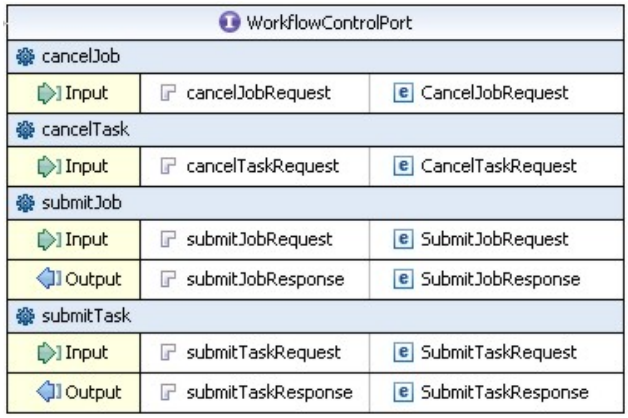

Figure 2.1: Schematic of the WorkflowControl Definition.

2.2.1 submitJob

Description	input parameter	output
This method submits a job defined in WSL to the Scheduling Service and returns a handle to refer to the submitted job lateron, e.g. for status requests.	WorkflowType	xsd:String

2.2.2 cancelJob

Description	input parameter	output
This method initiates the cancelling of a job which has been submitted for execution, i.e. all dependent and running atomic tasks are cancelled. The handle of the job to be cancelled is required as parameter.	xsd:String	

2.2.3 submitTask

Description	input parameter	output
This method submits a task defined in JSDL to the WSS and retrieves a handle to refer to the submitted task lateron.	JobDefinitionType	xsd:String

2.2.4 cancelTask

Description	input parameter	output
This method cancels a task submitted to the WSS. The handle of the task to be cancelled is required as parameter.	xsd:String	

2.3 WorkflowMonitoringPort

The WorkflowMonitoring definition provides only two operations. Both operations ask for the actual status of a job or a task execution. They are formally described in the following sections and depicted in Figure 2.2.

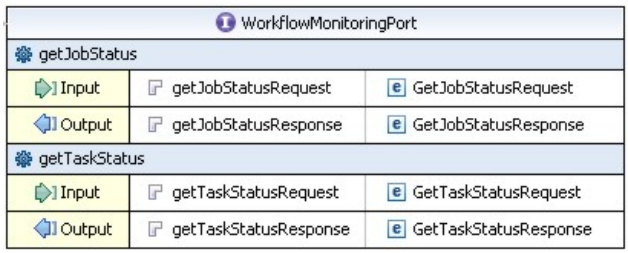

Figure 2.2: Schematic of the WorkflowMonitoring Definition.

2.3.1 getJobStatus

Description	input parameter	output
This method is used to get the status of a job's execution. As parameter the handle of the monitored job is needed. The return value is one of the enumeration values from the JobStatusEnumerationType which is detailed in Section 2.4.	xsd:String	JobStatusEnumerationType

2.3.2 getTaskStatus

Description	input parameter	output
This method is used to get the status of a task's execution. As parameter the handle of the monitored job is needed. The return value is one of the enumeration values from the TaskStatusEnumerationType which is detailed in Section 2.4.	xsd:String	TaskStatusEnumerationType

2.4 Enumeration Types

In this Section the possible values of the enumeration types defined in Section 2.1 are presented and explained. Table 2.3 shows those values JobStatusEnumeration supports for jobs, while Tables 2.4, 2.5 and 2.6 give the values supported by TaskStatusEnumeration.

JobStatusEnumeration	Description
submitted	The job has been successfully submitted to the WSS.
active	The job, i.e atomic tasks of the job are currently executed.
aborted	The job's execution was cancelled by the WSS due to an fatal error during processing.
cancelled	The job's execution was cancelled from the outside, i.e. through cancelJob at the WSS interface.
completed	The execution of the job is completed.

Table 2.3: Enumeration values for JobEnumerationType.

TaskStatusEnumeration (status messages)	Description
waiting	The task is waiting for execution by the WSS.
queued	The task is queued as the WSS cannot find an adequate resource to execute the task on.
initializing	The task's environment is initialized.
beginstaging	The WSS prepares for the staging of data described by the task.
performstaging	The WSS executes the data staging process described by the task.
endstaging	The staging process has been completed.
begintransfer	The WSS initiates a transport described by the task.
performtransfer	The transport described by the task is performed.
endtransfer	The transport performed is completed.
stepin	The task's execution begins with necessary setups before start.
running	The task is being executed.
cleanup	The task's environment is cleaned up.
stepout	The tasks execution has finished and the environment is destroyed.
beginpublish	Results of the task's execution are prepared to be published.
performpublish	Results of the task's execution are published.
endpublish	Publishing of task execution's results is completed.
finished	The execution of the task has been finished successfully.

Table 2.4: Enumeration values of the TaskEnumerationType concerning the processing status.

TaskStatusEnumeration (cancel/abort messages)	Description
aborted	The task's execution has been cancelled internally due to a fatal error.
cancelled	The task has been cancelled externally, i.e. through cancelTask at the WSS interface.

Table 2.5: Enumeration values of the TaskEnumerationType concerning the reason for abort.

TaskStatusEnumeration (error messages)	Description
erroronstaging	An error occured during the staging process.
errorontransfer	An error occured during transport.
erroronexecution	An error occured during task execution.
erroronpublish	An error occured during the publishing of execution results.

Table 2.6: Enumeration values the TaskEnumerationType concerning error types.

Bibliography

[JSDL05] A. Anjomshoaa, F. Brisard, M. Drescher, D. Fellows, A. Ly, S. McGough, D. Pulsipher, A. Savva, *Job Submission Description Language (JSDL) Specification*, Version 1.0, www.ggf.org, 11/2005

[WSL06] C. Grimme, A. Papaspyrou, *Workflow Specification Language*, Version 0.1, C3Grid Internal Report, 2006

3 APPENDIX

3.1 WorkflowSchedulerTypes Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  targetNamespace="http://c3grid.de/webservice/workflowscheduler/"
  xmlns:wss="http://c3grid.de/webservice/workflowscheduler/"
  xmlns:jsdl="http://schemas.ggf.org/jsdl/2005/11/jsdl"
  xmlns:wsl="http://c3grid.de/language/wsl/"
  >
  <xsd:import namespace="http://c3grid.de/language/wsl/"
    schemaLocation="wsl-0.1.xsd">
  </xsd:import>

  <xsd:import namespace="http://schemas.ggf.org/jsdl/2005/11/jsdl"
    schemaLocation="jsdl-1.0.xsd">
  </xsd:import>


  <xsd:simpleType name="TaskStatusEnumerationType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="waiting" />
      <xsd:enumeration value="queued" />
      <xsd:enumeration value="running" />
      <xsd:enumeration value="aborted" />
      <xsd:enumeration value="cancelled" />
      <xsd:enumeration value="finished" />
      <xsd:enumeration value="initializing" />
      <xsd:enumeration value="stepin" />
      <xsd:enumeration value="cleanup" />
      <xsd:enumeration value="stepout" />
      <xsd:enumeration value="beginstaging" />
      <xsd:enumeration value="performstaging" />
      <xsd:enumeration value="endstaging" />
      <xsd:enumeration value="begintransport" />
      <xsd:enumeration value="performtransport" />
      <xsd:enumeration value="endtransport" />
      <xsd:enumeration value="beginpublish" />
      <xsd:enumeration value="performpublish" />
      <xsd:enumeration value="endpublish" />
      <xsd:enumeration value="erroronstaging" />
```

```
      <xsd:enumeration value="errorontransport" />
      <xsd:enumeration value="erroronexecution" />
      <xsd:enumeration value="erroronpublish" />
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="JobStatusEnumerationType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="submitted" />
      <xsd:enumeration value="active" />
      <xsd:enumeration value="aborted" />
      <xsd:enumeration value="cancelled" />
      <xsd:enumeration value="completed" />
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:element name="CancelJobRequest" type="xsd:string"></xsd:element>

  <xsd:element name="CancelTaskRequest" type="xsd:string"></xsd:element>

  <xsd:element name="GetJobStatusRequest" type="xsd:string"></xsd:element>

  <xsd:element name="GetJobStatusResponse"
    type="wss:JobStatusEnumerationType">
  </xsd:element>

  <xsd:element name="GetTaskStatusRequest" type="xsd:string"></xsd:element>

  <xsd:element name="GetTaskStatusResponse"
    type="wss:TaskStatusEnumerationType">
  </xsd:element>

  <xsd:element name="SubmitJobRequest" type="wsl:WorkflowType"></xsd:element>

  <xsd:element name="SubmitJobResponse" type="xsd:string"></xsd:element>

  <xsd:element name="SubmitTaskRequest" type="jsdl:JobDefinition_Type"></xsd:element>

  <xsd:element name="SubmitTaskResponse" type="xsd:string"></xsd:element>

</xsd:schema>
```

3.2 WorkflowControlPort

```
<?xml version="1.0" encoding="UTF-8"?>
 <wsdl:definitions name="WorkflowControlService"
  targetNamespace="http://c3grid.de/webservice/workflowscheduler/control"
```

```
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wfsc="http://c3grid.de/webservice/workflowscheduler/control"
xmlns:wft="http://c3grid.de/webservice/workflowscheduler/">
<wsdl:types>
  <schema
    targetNamespace="http://c3grid.de/webservice/workflowscheduler/"
    xmlns="http://www.w3.org/2001/XMLSchema">

    <include schemaLocation="WorkflowSchedulerTypes.xsd"></include>
  </schema>
</wsdl:types>
<wsdl:message name="cancelJobRequest">
  <wsdl:part name="cancelJobRequest" element="wft:CancelJobRequest"></wsdl:part>
</wsdl:message>
<wsdl:message name="cancelTaskRequest">
  <wsdl:part name="cancelTaskRequest" element="wft:CancelTaskRequest"></wsdl:part>
</wsdl:message>
<wsdl:message name="submitJobResponse">
  <wsdl:part name="submitJobResponse" element="wft:SubmitJobResponse"></wsdl:part>
</wsdl:message>
<wsdl:message name="submitJobRequest">
  <wsdl:part name="submitJobRequest" element="wft:SubmitJobRequest"></wsdl:part>
</wsdl:message>
<wsdl:message name="submitTaskResponse">
  <wsdl:part name="submitTaskResponse" element="wft:SubmitTaskResponse"></wsdl:part>
</wsdl:message>
<wsdl:message name="submitTaskRequest">
  <wsdl:part name="submitTaskRequest"
    element="wft:SubmitTaskRequest">
  </wsdl:part>
</wsdl:message>
<wsdl:portType name="WorkflowControlPort">
  <wsdl:operation name="cancelJob">
    <wsdl:input message="wfsc:cancelJobRequest"></wsdl:input>
  </wsdl:operation>
  <wsdl:operation name="cancelTask">
    <wsdl:input message="wfsc:cancelTaskRequest"></wsdl:input>
  </wsdl:operation>
  <wsdl:operation name="submitJob">
    <wsdl:input message="wfsc:submitJobRequest"></wsdl:input>
    <wsdl:output message="wfsc:submitJobResponse"></wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="submitTask">
    <wsdl:input message="wfsc:submitTaskRequest"></wsdl:input>
    <wsdl:output message="wfsc:submitTaskResponse"></wsdl:output>
```

```
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="WorkflowControlBinding"
  type="wfsc:WorkflowControlPort">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="cancelJob">
    <soap:operation
      soapAction="http://c3grid.de/webservice/workflowscheduler/control/cancelJob" />
    <wsdl:input>
      <soap:body use="literal" parts=" cancelJobRequest" />
    </wsdl:input>
  </wsdl:operation>
  <wsdl:operation name="cancelTask">
    <soap:operation
      soapAction="http://c3grid.de/webservice/workflowscheduler/control/cancelTask" />
    <wsdl:input>
      <soap:body use="literal" parts=" cancelTaskRequest" />
    </wsdl:input>
  </wsdl:operation>
  <wsdl:operation name="submitJob">
    <soap:operation
      soapAction="http://c3grid.de/webservice/workflowscheduler/control/submitJob" />
    <wsdl:input>
      <soap:body use="literal" parts=" submitJobRequest" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" parts=" submitJobResponse" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="submitTask">
    <soap:operation
      soapAction="http://c3grid.de/webservice/workflowscheduler/control/submitTask" />
    <wsdl:input>
      <soap:body use="literal" parts=" submitTaskRequest" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" parts=" submitTaskResponse" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="WorkflowControlService">
  <wsdl:port name="WorkflowControlPort"
    binding="wfsc:WorkflowControlBinding">
    <soap:address location="http://www.example.org/" />
  </wsdl:port>
</wsdl:service>
```

```
</wsdl:definitions>
```

3.3 WorkflowMonitoringPort

```
<?xml version="1.0" encoding="UTF-8"?>
 <wsdl:definitions name="WorkflowMonitoringService"
  targetNamespace="http://c3grid.de/webservice/workflowscheduler/monitoring"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wfsc="http://c3grid.de/webservice/workflowscheduler/monitoring"
  xmlns:wft="http://c3grid.de/webservice/workflowscheduler/">

  <wsdl:types>
    <schema
      targetNamespace="http://c3grid.de/webservice/workflowscheduler/"
      xmlns="http://www.w3.org/2001/XMLSchema">

      <include schemaLocation="WorkflowSchedulerTypes.xsd"></include>
    </schema>
  </wsdl:types>

  <wsdl:message name="getJobStatusResponse">
    <wsdl:part name="getJobStatusResponse"
      element="wft:GetJobStatusResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getJobStatusRequest">
    <wsdl:part name="getJobStatusRequest" element="wft:GetJobStatusRequest"></wsdl:part>
  </wsdl:message>
  <wsdl:message name="getTaskStatusResponse">
    <wsdl:part name="getTaskStatusResponse"
      element="wft:GetTaskStatusResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getTaskStatusRequest">
    <wsdl:part name="getTaskStatusRequest" element="wft:GetTaskStatusRequest"></wsdl:part>
  </wsdl:message>

  <wsdl:portType name="WorkflowMonitoringPort">
    <wsdl:operation name="getJobStatus">
      <wsdl:input message="wfsc:getJobStatusRequest"></wsdl:input>
      <wsdl:output message="wfsc:getJobStatusResponse"></wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getTaskStatus">
      <wsdl:input message="wfsc:getTaskStatusRequest"></wsdl:input>
      <wsdl:output message="wfsc:getTaskStatusResponse"></wsdl:output>
```

```
    </wsdl:operation>
  </wsdl:portType>

  <wsdl:binding name="WorkflowMonitoringBinding"
    type="wfsc:WorkflowMonitoringPort">
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="getJobStatus">
      <soap:operation
        soapAction="http://c3grid.de/webservice/workflowscheduler/monitoring/getJobStatus" />
      <wsdl:input>
        <soap:body use="literal" parts=" getJobStatusRequest" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" parts=" getJobStatusResponse" />
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getTaskStatus">
      <soap:operation
        soapAction="http://c3grid.de/webservice/workflowscheduler/monitoring/getTaskStatus" />
      <wsdl:input>
        <soap:body use="literal" parts=" getTaskStatusRequest" />
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal" parts=" getTaskStatusResponse" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>

  <wsdl:service name="WorkflowMonitoringService">
    <wsdl:port name="WorkflowMonitoringPort"
      binding="wfsc:WorkflowMonitoringBinding">
      <soap:address location="http://www.example.org/" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```