



NextGRID SLA Template Repository Profile

Editors:	Roland Kübert	HLRS

Date	Author	Comments	Version	Status
10.07.2007	Roland Kübert	Initial Version	0.1	Draft
11.10.2007	Roland Kübert	Update incorporating comments from the internal review	0.2	Draft
19.10.2007	Roland Kübert	Response to second review	0.3	Stable draft
07.01.2008	Roland Kübert	Added WSDL information	0.4	Stable draft
29.01.2008	Roland Kübert	Updated WSDL and use cases	0.5	Stable draft
05.02.2008	David Snelling	Final QA	1.0	Final



CONTENTS

1 INTRODUCTION 3

1.1 Profile Overview..... 3

1.2 Relationship to Use Cases 4

1.3 Notational Conventions..... 4

1.4 Profile Identification and Versioning 5

2 PROFILE CONFORMANCE..... 5

2.1 Conformance Targets 5

2.2 Claiming Conformance..... 5

3 NEXTGRID SLA TEMPLATE REPOSITORY FUNCTIONS..... 6

3.1 Requirements from Use Cases..... 6

3.2 Implementation 6

3.3 Operations..... 6

 3.3.1 getTemplateByName 7

 3.3.2 getTemplateForBid..... 8

 3.3.3 getTemplatesForBid 8

 3.3.4 queryTemplates 9

 3.3.5 publishTemplate 10

 3.3.6 revokeTemplate 11

3.4 Port Type 12

3.5 WSDL Specification 13

4 REFERENCES 18



1 Introduction

This document defines the NextGRID SLA Template Repository Profile 1.0 (hereafter, “this Profile”). It consists of a set of de facto, institutional or evolving institutional Web services specifications, along with clarifications, refinements, interpretations and amplifications of those specifications that promote interoperability among implementations of those specifications within the context of the NextGRID architecture.

This Profile forms part of the NextGRID Generalized Specifications, which aim at capturing NextGRID architectural concepts in a set of profiles that may be composed together. These profiles are specified in such a way that they could be implemented in terms of other well known specifications. While overall consistency is achieved at the conceptual level, and captured through the motivating Use Cases accompanying each Generalized Specification, system implementation in terms of other specifications may not be consistent between different profiles. Thus, each profile defines a realisation of the underlying concept that can be implemented. However, implementers of the full NextGRID architecture may need to support multiple competing underlying specifications.

An introduction to this Profile and its relationship to the other NextGRID profiles are given in this section.

Section 2, “Profile Conformance”, explains what it means to be conformant to this Profile.

Each subsequent section in turn addresses a component of this Profile, and consists of two parts:

- An overview section detailing the component specifications and their extensibility points; followed by
- A number of subsections that address individual parts of the component specifications.

Note that there is no relationship between the section numbers in this document and those in the specifications that it references.

1.1 Profile Overview

This Profile is intended for use when implementing services that are concerned with distributed computing in line with the concepts of the NextGRID project. A service implementation that uses the specifications referenced herein in a manner conformant with this Profile may be said to be an “implementation of the NextGRID SLA Template Repository Profile 1.0” or, more informally, to be a “NextGRID SLA Template Repository Service”.

The primary issues addressed in the profile are as follows:

- *NextGRID Basic Profile* [3]. This Profile mandates the use of aspects of the NextGRID Basic Profile.
- *OGSA WSRF Basic Profile* [2]. This Profile mandates the use of aspects of the OGSA WSRF Basic Profile.

In NextGRID, the SLA Template Repository Service is implemented as a WSRF resource.

1.2 Relationship to Use Cases

The use case part of the SLA Template Repository Generalised Specification [4] lists a number of use cases which are in general abstract and not taken from any of NextGRID's experiments or demonstrations. The following use cases are listed:

- Publish template;
- Revoke template;
- Query by name;
- Query random template for a bid;
- Query all templates for a given bid; and
- Query templates.

1.3 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 [5].

Normative statements of requirements in the Profile are presented in the manner detailed in the WS-I Basic Profile 1.1 Conformance Requirements section [6].

Both requirement statements and extensibility statements can be considered namespace-qualified. This specification uses a number of namespace prefixes throughout; their associated URIs are listed below. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table 1 Namespaces used by NextGRID SLA Template Repository Profile 1.0.

Prefix	Namespace
soap	http://schemas.xmlsoap.org/soap/envelope
wSDL	http://schemas.xmlsoap.org/wSDL
xsd	http://www.w3.org/2001/XMLSchema
wsa	http://www.w3.org/2005/08/addressing

wsrp-rp	http://docs.oasis-open.org/wsrp/rp-2
wsrp-rpw	http://docs.oasis-open.org/wsrp/rpw-2
wsrp-bf	http://docs.oasis-open.org/wsrp/bf-2

1.4 Profile Identification and Versioning

Profile identification and versioning use the style described in the WS-I Basic Profile 1.1 [6] and abides by the normative descriptions contained therein. The name of this Profile is “NextGRID SLA Template Repository Profile” and the version number is “1.0”.

2 Profile Conformance

Conformance to the Profile is defined normatively in the WS-I Basic Profile 1.1 [6], section 2, “Profile Conformance”. This Profile abides by those definitions.

2.1 Conformance Targets

Since the Profile is an extension of the WS-I Basic Profile 1.1 [6], the OGSA WSRF Basic Profile 1.0 [2], and the NextGRID Basic Profile 1.0 [3], it may place further restrictions on conformance targets defined therein.

The following conformance targets are used in this Profile:

- **DESCRIPTION** – descriptions of types, messages, interfaces and their concrete protocol and data format bindings, and the network access points associated with Web services (e.g., WSDL descriptions) (from WS-I Basic Profile 1.1).
- **INSTANCE** – software that implements a wsdl:port (from WS-I Basic Profile 1.1, without “bindingTemplate” from the namespace urn:uddi-org:api_v2).
- **RECEIVER** – software that consumes a message according to the protocol(s) associated with that message (e.g., SOAP processors) (from WS-I Basic Profile 1.1).
- **MESSAGE** – protocol elements that transport the serialization of the soap:Envelope element and its content (e.g., SOAP/HTTP messages).

2.2 Claiming Conformance

Claims of conformance to the Profile and the attachments mechanisms are the same as normatively described in the WS-I Basic Profile 1.1 [6]. The conformance claim URI for this Profile is: “<http://www.nextgrid.org/slatemplaterepositoryprofile/v-1>”.

3 NextGRID SLA Template Repository Functions

3.1 Requirements from Use Cases

A NextGRID Template Repository should provide the following functions in order to fulfil the use cases [4]:

- Publish a template into a Template Repository;
- Revoke a template from a Template Repository;
- Query a template by name;
- Query a random template for a given bid;
- Query all templates for a given bid; and
- Query templates.

The use cases can be divided into two parts, where one is accessible by the service provider operating the repository and the other is accessible by anyone.

The functions *publish* and *revoke* are only accessible by the service provider, while all the query functions can be accessed by anyone, although the service provider is of course free to restrict access to the Template Repository in ways which do not concern this specification.

3.2 Implementation

An SLA Template Repository implementation must realise the functions shown in Section 3.1.

The NextGRID SLA Template Repository exposes Web Service interfaces in compliance with WS-RF standards and conforming to the NextGRID Basic Profile [3]. While designing WS-RF Web Services, a clear distinction has to be made upon the selection of stateful and stateless Web Services. In the NextGRID implementation, a Template Repository Service is exposed as a single-resource stateful Web Service.

3.3 Operations

The WSDL for the interface is given in full below in section 3.5. Informally, the SLA Template Repository operations are:

- GetTemplateByNameResponse getTemplateByName(GetTemplateByNameRequest)
- GetTemplateForBidResponse getTemplateForBid(GetTemplateForBidRequest)
- GetTemplatesForBidResponse getTemplatesForBid(GetTemplatesForBidRequest)
- QueryTemplatesResponse queryTemplates (QueryTemplatesRequest)
- PublishTemplateResponse publishTemplate(PublishTemplateRequest)

- RevokeTemplateResponse revokeTemplate(RevokeTemplateRequest)

As can be seen easily, each operation has its argument type and its return value type set according to its name, e.g. operation *foo* has an argument of type *fooRequest* and returns an instance of type *fooResponse*. Each operation may also throw an exception of type *TemplateRepositoryFaultType*; this message must contain a message detailing the exact cause of the error. The operation *queryTemplates* may additionally throw an exception of type *UnsupportedQueryDialect* if the query dialect is not supported; the exception must contain a message stating the error and informing about supported query dialects.

3.3.1 getTemplateByName

The operation *getTemplateByName* uses the following messages:

```
<operation name="getTemplateByName">
  <input message="tns:GetTemplateByNameInputMessage" />
  <output message="tns:GetTemplateByNameOutputMessage" />
  <fault name="templateRepositoryFault"
    message="tns:TemplateRepositoryFaultMessage">
  </fault>
</operation>
```

The operation takes the following input and output messages:

```
<message name="GetTemplateByNameOutputMessage">
  <part name="parameters" element="tns:GetTemplateByNameResponse" />
</message>
<message name="GetTemplateForBidInputMessage">
  <part name="parameters" element="tns:GetTemplateForBidRequest" />
</message>
```

The input message element contains the name of the template which shall be obtained:

```
<xsd:element name="GetTemplateByNameRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="templateName" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The output message element contains exactly one *SLAType* object, the requested template:

```
<xsd:element name="GetTemplateByNameResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="template" type="sla:SLAType" minOccurs="1"
        maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

3.3.2 getTemplateForBid

The operation `getTemplateForBid` uses the following messages:

```
<operation name="getTemplateForBid">
  <input message="tns:GetTemplateForBidInputMessage" />
  <output message="tns:GetTemplateForBidOutputMessage" />
  <fault name="templateRepositoryFault"
    message="tns:TemplateRepositoryFaultMessage">
  </fault>
</operation>
```

The operation takes the following input and output messages:

```
<message name="GetTemplateForBidInputMessage">
  <part name="parameters" element="tns:GetTemplateForBidRequest" />
</message>
<message name="GetTemplateForBidOutputMessage">
  <part name="parameters" element="tns:GetTemplateForBidResponse" />
</message>
```

The input message element contains the name of the template which shall be obtained:

```
<xsd:element name="GetTemplateForBidRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="bid" type="sla:SLAType" minOccurs="1"
        maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The output message element contains exactly one `SLAType` object storing the requested template:

```
<xsd:element name="GetTemplateForBidResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="template" type="sla:SLAType" minOccurs="1"
        maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

3.3.3 getTemplatesForBid

The operation `getTemplatesForBid` uses the following messages:

```
<operation name="getTemplatesForBid">
  <input message="tns:GetTemplatesForBidInputMessage" />
  <output message="tns:GetTemplatesForBidOutputMessage" />
  <fault name="templateRepositoryFault"
    message="tns:TemplateRepositoryFaultMessage">
  </fault>
</operation>
```

```
</fault>
</operation>
```

The operation takes the following input and output messages:

```
<message name="GetTemplatesForBidInputMessage">
  <part name="parameters" element="tns:GetTemplatesForBidRequest" />
</message>
<message name="GetTemplatesForBidOutputMessage">
  <part name="parameters" element="tns:GetTemplatesForBidResponse" />
</message>
```

The input message element contains exactly one bid for which templates shall be obtained:

```
<xsd:element name="GetTemplatesForBidRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="bid" type="sla:SLAType" minOccurs="1"
        maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The output message element contains at least one template matching the given bid:

```
<xsd:element name="GetTemplatesForBidResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="templates" type="sla:SLAType"
        minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

3.3.4 queryTemplates

The operation queryTemplates uses the following messages:

```
<operation name="queryTemplates">
  <input message="tns:QueryTemplatesInputMessage"></input>
  <output message="tns:QueryTemplatesOutputMessage"></output>
  <fault name="templateRepositoryFault" message="tns:TemplateRepositoryFaultMessage">
  </fault>
  <fault name="unsupportedDialectFault" message="tns:UnsupportedDialectFaultMessage">
  </fault>
</operation>
```

The operation takes the following input and output messages:

```
<message name="QueryTemplatesInputMessage">
  <part name="parameters" element="tns:QueryTemplatesRequest" />
</message>
```

```
<message name="QueryTemplatesOutputMessage">
  <part name="parameters" element="tns:QueryTemplatesResponse" />
</message>
```

The input message element contains the query dialect and the query itself:

```
<xsd:element name="QueryTemplatesRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="query" type="xsd:string" minOccurs="1" maxOccurs="1" />
      <xsd:element name="dialect" type="xsd:string" minOccurs="1"
        maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

A NextGRID Template Repository must support at least the query dialect XPath [7], specified by the case-insensitive string “xpath”. The support of other query dialects is optional.

The output message element contains a list of templates matching the given query:

```
<xsd:element name="QueryTemplatesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="templates" type="sla:SLAType"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

3.3.5 publishTemplate

The operation publishTemplate uses the following messages:

```
<operation name="publishTemplate">
  <input message="tns:PublishTemplateInputMessage" />
  <output message="tns:PublishTemplateOutputMessage" />
  <fault name="templateRepositoryFault"
    message="tns:TemplateRepositoryFaultMessage">
  </fault>
</operation>
```

The operation takes the following input and output messages:

```
<message name="PublishTemplateInputMessage">
  <part name="parameters" element="tns:PublishTemplateRequest" />
</message>
<message name="PublishTemplateOutputMessage">
  <part name="parameters" element="tns:PublishTemplateResponse" />
</message>
```

NextGRID SLA Template Repository Profile

The input message element contains the name of the template which shall be published and the template itself:

```
<xsd:element name="PublishTemplateRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="templateName" type="xsd:string" minOccurs="1"
        maxOccurs="1" />
      <xsd:element name="template" type="sla:SLAType" minOccurs="1"
        maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

The output message element contains a flag which shows if the template could be published. Since it is in most cases not sufficient to only return a *false* value without giving further reasons, a `TemplateRepositoryFaultType` should be returned if any error occurs.

```
<xsd:element name="PublishTemplateResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="success" type="xsd:boolean" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

3.3.6 revokeTemplate

The operation `revokeTemplate` uses the following messages:

```
<operation name="revokeTemplate">
  <input message="tns:RevokeTemplateInputMessage" />
  <output message="tns:RevokeTemplateOutputMessage" />
  <fault name="templateRepositoryFault"
    message="tns:TemplateRepositoryFaultMessage">
  </fault>
</operation>
```

The operation takes the following input and output messages:

```
<message name="RevokeTemplateInputMessage">
  <part name="parameters" element="tns:RevokeTemplateRequest" />
</message>
<message name="RevokeTemplateOutputMessage">
  <part name="parameters" element="tns:RevokeTemplateResponse" />
</message>
```

The input message element contains the name of the template which shall be revoked from the repository:

```
<xsd:element name="RevokeTemplateRequest">
  <xsd:complexType>
```

```

    <xsd:sequence>
      <xsd:element name="templateName" type="xsd:string" minOccurs="1"
        maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

The output message element is empty:

```

<xsd:element name="RevokeTemplateResponse">
  <xsd:complexType />
</xsd:element>

```

3.4 Port Type

The SLA Template Repository consists of a single port type which provides all operations given in section 3.3.

```

<portType name="SLATemplateRepositoryPortType"
  wsdlpp:extends="wsrpw:GetSLATemplateRepositoryProperty"
  wsrp:ResourceProperties="tns:SLATemplateRepositoryResourceProperties">

  <operation name="revokeTemplate">
    <input message="tns:RevokeTemplateInputMessage" />
    <output message="tns:RevokeTemplateOutputMessage" />
    <fault name="templateRepositoryFault"
      message="tns:TemplateRepositoryFaultMessage">
    </fault>
  </operation>

  <operation name="publishTemplate">
    <input message="tns:PublishTemplateInputMessage" />
    <output message="tns:PublishTemplateOutputMessage" />
    <fault name="templateRepositoryFault"
      message="tns:TemplateRepositoryFaultMessage">
    </fault>
  </operation>

  <operation name="getTemplateByName">
    <input message="tns:GetTemplateByNameInputMessage" />
    <output message="tns:GetTemplateByNameOutputMessage" />
    <fault name="templateRepositoryFault"
      message="tns:TemplateRepositoryFaultMessage">
    </fault>
  </operation>

  <operation name="getTemplateForBid">
    <input message="tns:GetTemplateForBidInputMessage" />

```

NextGRID SLA Template Repository Profile

```

        <output message="tns:GetTemplateForBidOutputMessage" />
        <fault name="templateRepositoryFault "
            message="tns:TemplateRepositoryFaultMessage">
        </fault>
    </operation>

    <operation name="getTemplatesForBid">
        <input message="tns:GetTemplatesForBidInputMessage" />
        <output message="tns:GetTemplatesForBidOutputMessage" />
        <fault name="templateRepositoryFault "
            message="tns:TemplateRepositoryFaultMessage">
        </fault>
    </operation>

    <operation name="queryTemplates">
        <input message="tns:QueryTemplatesInputMessage"></input>
        <output message="tns:QueryTemplatesOutputMessage">
        </output>
        <fault name="templateRepositoryFault "
            message="tns:TemplateRepositoryFaultMessage">
        </fault>
        <fault name="unsupportedDialectFault "
            message="tns:UnsupportedDialectFaultMessage">
        </fault>
    </operation>
</portType>

```

All operations may return a *TemplateRepositoryFault*, which must detail on exact failure circumstances; the operation *queryTemplates* may additionally return a fault of type *UnsupportedDialectFault*, which must detail on exact failure circumstances and provide information on query dialects supported by the template repository. Furthermore, for each operation named *foo*, the input message is named *fooInputMessage* and the output message is named *fooOutputMessage*.

3.5 WSDL Specification

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions name="SLATemplateRepositoryService"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    targetNamespace="http://www.nextgrid.org/namespaces/services/SLATemplateRepositoryService"
    xmlns:wsbf="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-BaseFaults-1.2-draft-01.xsd"
    xmlns:wsbfw="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-BaseFaults-1.2-draft-01.wsdl"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:sla="http://www.nextgrid.org/sla/2006/08"
    xmlns:wsrp="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ResourceProperties-1.2-draft-01.xsd"
    xmlns:wsrpw="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-ResourceProperties-1.2-draft-01.wsdl"
    xmlns:wsdldpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor"
    xmlns:tns="http://www.nextgrid.org/namespaces/services/SLATemplateRepositoryService">
    <import
        namespace="http://docs.oasis-open.org/wsr/2004/06/wsr/WS-BaseFaults-1.2-draft-01.wsdl"

```

NextGRID SLA Template Repository Profile

```

        location="../../../schema/wsrf/faults/WS-BaseFaults.wsdl" />
    <import
draft-01.wsdl"
        namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-
        location="../../../schema/wsrf/properties/WS-ResourceProperties.wsdl" />

    <!--=====
        T Y P E S
        =====-->
    <types>
        <xsd:schema

targetNamespace="http://www.nextgrid.org/namespaces/services/SLATemplateRepositoryService"

xmlns:tns="http://www.nextgrid.org/namespaces/services/SLATemplateRepositoryService"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing">

    <!-- Import SLAType -->
    <xsd:import namespace="http://www.nextgrid.org/sla/2006/08"
        schemaLocation="../../../sla/SLA-070707-DS-v4-5-1.xsd" />

    <!-- FAULTS -->
    <xsd:complexType name="TemplateRepositoryFaultType">
        <xsd:complexContent>
            <xsd:extension base="wsbf:BaseFaultType">
                <xsd:sequence>
                    <xsd:element name="details"
                        type="xsd:string" />
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="templateRepositoryFault"
        type="tns:TemplateRepositoryFaultType" />

    <xsd:complexType name="UnsupportedDialectFaultType">
        <xsd:complexContent>
            <xsd:extension base="wsbf:BaseFaultType">
                <xsd:sequence>
                    <xsd:element name="details"
                        type="xsd:string" />
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="unsupportedDialectFault"
        type="tns:UnsupportedDialectFaultType" />

    <!-- REQUESTS AND RESPONSES -->

    <!--
        Set a template to the repository.
    -->
    <xsd:element name="PublishTemplateRequest">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="templateName"
                    type="xsd:string" minOccurs="1" maxOccurs="1"

                    <xsd:element name="template" type="sla:SLAType"
                        minOccurs="1" maxOccurs="1" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>

    <xsd:element name="PublishTemplateResponse">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="success" type="xsd:boolean" />
            </xsd:sequence>

```

NextGRID SLA Template Repository Profile

```

        </xsd:complexType>
    </xsd:element>

    <xsd:element name="RevokeTemplateRequest">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="templateName"
                    type="xsd:string" minOccurs="1" maxOccurs="1"
                />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="RevokeTemplateResponse">
        <xsd:complexType />
    </xsd:element>

    <!--
        Get a template from the repository by name.
    -->
    <xsd:element name="GetTemplateByNameRequest">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="templateName"
                    type="xsd:string" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="GetTemplateByNameResponse">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="template" type="sla:SLAType"
                    minOccurs="1" maxOccurs="1" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <!--
        Interface for the negotiator to obtain a single template which
        corresponds to the given bid.
    -->
    <xsd:element name="GetTemplateForBidRequest">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="bid" type="sla:SLAType"
                    minOccurs="1" maxOccurs="1" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="GetTemplateForBidResponse">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="template" type="sla:SLAType"
                    minOccurs="1" maxOccurs="1" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <!--
        Interface for the negotiator to obtain all templates which
        correspond to the given bid.
    -->
    <xsd:element name="GetTemplatesForBidRequest">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="bid" type="sla:SLAType"
                    minOccurs="1" maxOccurs="1" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

```

NextGRID SLA Template Repository Profile

```

<xsd:element name="GetTemplatesForBidResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="templates" type="sla:SLAType"
        minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!--
      Obtain SLA templates by performing an query
-->
<xsd:element name="QueryTemplatesRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="query" type="xsd:string"
minOccurs="1" maxOccurs="1" />
      <xsd:element name="dialect" type="xsd:string"
minOccurs="1" maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="QueryTemplatesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="templates" type="sla:SLAType"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

</xsd:schema>
</types>

<!--=====
      M E S S A G E S
=====-->
<message name="TemplateRepositoryFaultMessage">
  <part name="fault" element="tns:templateRepositoryFault" />
</message>
<message name="UnsupportedDialectFaultMessage">
  <part name="fault" element="tns:unsupportedDialectFault" />
</message>

<message name="PublishTemplateInputMessage">
  <part name="parameters" element="tns:PublishTemplateRequest" />
</message>
<message name="PublishTemplateOutputMessage">
  <part name="parameters" element="tns:PublishTemplateResponse" />
</message>
<message name="RevokeTemplateInputMessage">
  <part name="parameters" element="tns:RevokeTemplateRequest" />
</message>
<message name="RevokeTemplateOutputMessage">
  <part name="parameters" element="tns:RevokeTemplateResponse" />
</message>
<message name="GetTemplateByNameInputMessage">
  <part name="parameters" element="tns:GetTemplateByNameRequest" />
</message>
<message name="GetTemplateByNameOutputMessage">
  <part name="parameters" element="tns:GetTemplateByNameResponse" />
</message>
<message name="GetTemplateForBidInputMessage">
  <part name="parameters" element="tns:GetTemplateForBidRequest" />
</message>
<message name="GetTemplateForBidOutputMessage">
  <part name="parameters" element="tns:GetTemplateForBidResponse" />
</message>
<message name="GetTemplatesForBidInputMessage">

```

NextGRID SLA Template Repository Profile

```

    <part name="parameters" element="tns:GetTemplatesForBidRequest" />
</message>
<message name="GetTemplatesForBidOutputMessage">
    <part name="parameters"
        element="tns:GetTemplatesForBidResponse" />
</message>
<message name="QueryTemplatesInputMessage">
    <part name="parameters"
        element="tns:QueryTemplatesRequest" />
</message>
<message name="QueryTemplatesOutputMessage">
    <part name="parameters"
        element="tns:QueryTemplatesResponse" />
</message>

<!--=====
P O R T T Y P E
=====-->
<portType name="SLATemplateRepositoryPortType"
    wsdlpp:extends="wsrpw:GetSLATemplateRepositoryProperty"
    wsrp:ResourceProperties="tns:SLATemplateRepositoryResourceProperties">

    <operation name="revokeTemplate">
        <input message="tns:RevokeTemplateInputMessage" />
        <output message="tns:RevokeTemplateOutputMessage" />
        <fault name="templateRepositoryFault"
            message="tns:TemplateRepositoryFaultMessage">
        </fault>
    </operation>

    <operation name="publishTemplate">
        <input message="tns:PublishTemplateInputMessage" />
        <output message="tns:PublishTemplateOutputMessage" />
        <fault name="templateRepositoryFault"
            message="tns:TemplateRepositoryFaultMessage">
        </fault>
    </operation>

    <operation name="getTemplateByName">
        <input message="tns:GetTemplateByNameInputMessage" />
        <output message="tns:GetTemplateByNameOutputMessage" />
        <fault name="templateRepositoryFault"
            message="tns:TemplateRepositoryFaultMessage">
        </fault>
    </operation>

    <operation name="getTemplateForBid">
        <input message="tns:GetTemplateForBidInputMessage" />
        <output message="tns:GetTemplateForBidOutputMessage" />
        <fault name="templateRepositoryFault"
            message="tns:TemplateRepositoryFaultMessage">
        </fault>
    </operation>

    <operation name="getTemplatesForBid">
        <input message="tns:GetTemplatesForBidInputMessage" />
        <output message="tns:GetTemplatesForBidOutputMessage" />
        <fault name="templateRepositoryFault"
            message="tns:TemplateRepositoryFaultMessage">
        </fault>
    </operation>

    <operation name="queryTemplates">
        <input message="tns:QueryTemplatesInputMessage"></input>
        <output
            message="tns:QueryTemplatesOutputMessage">
        </output>
        <fault name="templateRepositoryFault"
            message="tns:TemplateRepositoryFaultMessage">
        </fault>
        <fault name="unsupportedDialectFault"
            message="tns:UnsupportedDialectFaultMessage">
        </fault>
    </operation>

```

</portType>

</definitions>

4 References

- [1] D. Snelling and M. Fisher and A. Basermann and F. Wray and P. Wieder and M. Surrige (eds.), NextGRID White Paper V5, NextGRID Project, July 25, 2005, http://www.nextgrid.org/download/publications/NextGRID_Architecture_White_Paper.pdf
- [2] I. Foster, T. Maguire, D. Snelling, OGSA WSRF Basic Profile 1.0, GFD.72, GF 2006. <http://www.ogf.org/documents/GFD.72.pdf>
- [3] V. Li and D. Snelling (eds.), NextGRID Basic Profile, V1.0, NextGRID Project, January 2008, http://www.nextgrid.org/GS/management_systems/basic/NextGRID_basic_profile.pdf.
- [4] NextGRID SLA Template Repository Use Cases, NextGRID, January, 2008, http://www.nextgrid.org/GS/management_systems/SLA_template_repository/NextGRID_SLA_template_repository_usecases.pdf.
- [5] S. Bradner (ed.): Key words for use in RFCs to Indicate Requirement Levels, The Internet Engineering Task Force Best Current Practice, March 1997. <http://www.ietf.org/rfc/rfc2119>
- [6] K. Ballinger, D. Ehnebuske, C. Ferris, M. Gudgin, C.K. Liu, M. Nottingham, and P. Yendluri (ed.): Basic Profile Version 1.1, Web Services Interoperability Organization Final Material, 24 August 2004. <http://www.wsi.org/Profiles/BasicProfile-1.1.html>
- [7] XML Path Language (XPath) Version 1.0, W3C Recommendation, November, 1999; <http://www.w3.org/TR/xpath>