UDAP Schema

Editors:	Ali Anjomshoaa	EPCC
	Philipp Wieder	FZJ

Date	Author	Comments	Version	Status
Mar 16, 2007	Philipp Wieder	Initial Version	V0.1	Draft
…	…	…	V0.2	Draft
Sep, 2007	Philipp Wieder	Response to internal review	V0.3	Stable draft

1 Introduction

Looking at state-of-the-art Grid architectures, information about an activity is fragmented and dispersed. Activity information, such as resource usage, security data, activity state, data requirements, and much more, is currently captured using a variety of different schemata (formatted in a various number of ways) and the information is stored in different ways and by different components. This dispersion of activity information leads to management, security and logistical overheads in discovering, accessing and using that information. This uncoordinated scheme for finding activity information leads to the need for a lot of the Grid components and Grid system coordinators. It results in an environment where activity information is managed by many systems. This makes it difficult for Grid components and users that rely on that information to know where the information is, leading to them having to search for and to monitor many sources of activity information. This understanding let to the development of UDAP, the Universal Dynamic Activity Package, a model that helps to manage and discover activity information in a more unified and co-ordinated fashion.

1.1 Profile Overview

The UDAP Generalised Specification consists of a UDAP schema document (the one at hand) and a UDAP use case document. There is no UDAP Profile as such, mainly since UDAP is not based on any other existing specification, but defining something new.

1.2 Relationships to Other Profiles

Potential implementations of UDAP have to follow the UDAP Basic Profile V1.0 [10].

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 2.

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 Basic Profile 1.0

Prefix	Namespace
ogsa-bp	http://schemas.ggf.org/ogsa/2006/01/wsrf-bp
soap	http://schemas.xmlsoap.org/soap/envelope

bsp-core	http://schemas.ggf.org/ogsa/2006/01/bsp-core
wsdl	http://schemas.xmlsoap.org/wsdl
xsd	http://www.w3.org/2001/XMLSchema
xsi	http://www.w3.org/2001/XMLSchema-instance
wsa	http://www.w3.org/2005/08/addressing
wsi	http://ws-i.org/schemas/conformanceClaim
wsrf-rp	http://docs.oasis-open.org/wsrf/rp-2
wsrf-rpw	http://docs.oasis-open.org/wsrf/rpw-2
wsrf-rl	http://docs.oasis-open.org/wsrf/rl-2
wsrf-rlw	http://docs.oasis-open.org/wsrf/rlw-2
wsrf-bf	http://docs.oasis-open.org/wsrf/bf-2
wsnt	http://docs.oasis-open.org/wsn/b-2
wsntw	http://docs.oasis-open.org/wsn/bw-2
nextgrid-bp	http://www.nextgrid.org/basicprofile/v-1

1.4 Profile Identification and Versioning

Profile identification and versioning uses the style described in the Profile Identification and Versioning section of WS-I Basic Profile 1.1 [6] and abides by the normative descriptions contained therein. The name of this Profile is “UDAP Schema” and version number is “1.0.”

2 Profile Conformance

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

2.1 Conformance Targets

This profile does not define any conformance targets.

2.2 Claiming Conformance

Claims of conformance to the Profile and the attachments mechanisms are the same as normatively described in the Claiming Conformance section of WS-I Basic Profile 1.1 [6]. .

The conformance claim URI for this Profile is “http://www.nextgrid.org/udapschema/v1”.

3 UDAP Schema

3.1 Pseudo Schema

```
<UDAP>
  <ActivityID/>
  <ActivityDescription Dialect="">
    <xsd:any##other/>*
  </ActivityDescription>
  <Record>
    <Entry Category="">*
      <TimeStamp>
        <xsd:dateTime/>
      </TimeStamp>
      <State/>
      <Resource>*
        <xsd:any##other/>*
      </Resource>
      <Context>*
        <xsd:any##other/>*
      </Context>
      <Dependency>*
        <xsd:any##other/>*
      </Dependency>
    </Entry>
  </Record>
  <Result>*
    <xsd:any##other/>*
  </Result>
</UDAP>
```

3.2 UDAP Schema Elements

3.2.1 The `UDAP` Element

3.2.1.1 Definition

This is the root element of a UDAP Document. It represents a unique, atomic activity and captures all the information about that activity.

3.2.1.2 Multiplicity

The multiplicity of this element is one.

3.2.1.3 Type

This element has a complex type. It must support the following elements:

- `ActivityID`
- `ActivityDescription`
- `Record`
- `Result`

3.2.1.4 Attributes

This element has no defined attributes.

3.2.2 The `ActivityID` Element

3.2.2.1 Definition

This element holds the unique ID of the unique, atomic activity whose information is captured by the UDAP Document.

3.2.2.2 Multiplicity

The multiplicity of this element is one.

3.2.2.3 Type

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

3.2.2.4 Attributes

This element has no defined attributes.

3.2.3 The `ActivityDescription` Element

3.2.3.1 Definition

This element describes the activity in a dialect specified by the `Dialect` attribute of the element.

3.2.3.2 Multiplicity

The multiplicity of this element is one.

3.2.3.3 Type

This element has a complex type. It must support the following elements:

- `xsd:any##other`

3.2.3.4 Attributes

The following attributes are defined:

- `Dialect` – the dialect of the activity description captured by this element.

3.2.4 The `Record` Element

3.2.4.1 Definition

This element captures a record of the activity information throughout its lifetime. Hence, a running history of the activity is captured by a number of `Entry` elements.

3.2.4.2 Multiplicity

The multiplicity of this element is one.

3.2.4.3 Type

This element has a complex type. It must support the following elements:

- `Entry`

3.2.4.4 Attributes

This element has no defined attributes.

3.2.5 The `Entry` Element

3.2.5.1 Definition

This element is an entry in the activity's record of existence. It captures a snapshot of the activity's information at a point in time.

3.2.5.2 Multiplicity

The multiplicity of this element is zero or more.

3.2.5.3 Type

This element has a complex type. It must support the following elements:

- `TimeStamp`
- `Resource`

- `Context`
- `State`
- `Dependency`

3.2.5.4 Attributes

The following attributes are defined:

- `Category` – the category of the Entry element specifies whether the entry is an original, current, or archived entry.

3.2.6 The `TimeStamp` Element

3.2.6.1 Definition

This element captures the time and date of an activity's record entry, i.e the point in time this record has been created.

3.2.6.2 Multiplicity

The multiplicity of this element is one.

3.2.6.3 Type

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

3.2.6.4 Attributes

This element has no defined attributes.

3.2.7 The `State` Element

3.2.7.1 Definition

This element captures the state of the activity at the time the entry it appears in is entered into the activity's record.

3.2.7.2 Multiplicity

The multiplicity of this element is one.

3.2.7.3 Type

The type of this element is `xsd:string`, which is restricted to the following values:

- `pending`
- `running`
- `runningPreempted`
- `runningMigrating`
- `runningPaused`
- `cancelled`
- `failed`

- `finished`

3.2.7.4 Attributes

This element has no defined attributes.

3.2.8 The `Resource` Element

3.2.8.1 Definition

This element specifies all of the resource information required by and associated to the activity.

3.2.8.2 Multiplicity

The multiplicity of this element is zero or more.

3.2.8.3 Type

This element has a complex type. It must support the following elements:

- `xsd:any##other`

3.2.8.4 Attributes

This element has no defined attributes.

3.2.9 The `Context` Element

3.2.9.1 Definition

This element specifies all of the contextual information, such as policy information, domain specific information, etc., required by and associated to the activity.

3.2.9.2 Multiplicity

The multiplicity of this element is zero or more.

3.2.9.3 Type

This element has a complex type. It must support the following elements:

- `xsd:any##other`

3.2.9.4 Attributes

This element has no defined attributes.

3.2.10 The `Dependency` Element

3.2.10.1 Definition

This element specifies some dependency information, such as dependencies on data, other activities, state of the system, etc., required by and associated to the activity.

3.2.10.2 Multiplicity

The multiplicity of this element is zero or more.

3.2.10.3 Type

This element has a complex type. It must support the following elements:

- `xsd:any##other`

3.2.10.4 Attributes

This element has no defined attributes.

3.2.11 The `Result` Element

3.2.11.1 Definition

This element captures a result of the activity.

3.2.11.2 Multiplicity

The multiplicity of this element is zero or more.

3.2.11.3 Type

This element has a complex type. It must support the following elements:

- `xsd:any##other`

3.2.11.4 Attributes

This element has no defined attributes.

3.3 Schema

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     targetNamespace="http://nextgrid.org/2007/01/udap"
xmlns:udap="http://nextgrid.org/2007/01/udap"
     attributeFormDefault="unqualified"
elementFormDefault="qualified">

     <!-- ****** The UDAP document ****** -->
     <xsd:element name="UDAP">
```

```
          <xsd:complexType>
               <xsd:sequence>
                    <xsd:element name="ActivityID"
type="xsd:string" minOccurs="1" maxOccurs="1"/>
                    <xsd:element name="ActivityDescription"
type="udap:DescriptionType" minOccurs="1" maxOccurs="1"/>
                    <xsd:element name="Record"
type="udap:RecordType" minOccurs="1" maxOccurs="1"/>
                    <xsd:element name="Result"
type="udap:ResultType" minOccurs="0" maxOccurs="unbounded"/>
               </xsd:sequence>
          </xsd:complexType>
     </xsd:element>

     <!-- The DescriptionType describes the activity. This is
           basically xsd:any, allowing to embed any activity
           description according to the "Dialect"
     -->
     <xsd:complexType name="DescriptionType">
          <xsd:sequence>
               <xsd:any namespace="##other" minOccurs="0"
processContents="lax"/>
          </xsd:sequence>
          <xsd:attribute name="Dialect" type="xsd:string"
use="required"/>
     </xsd:complexType>

     <!-- The RecordType -->
     <xsd:complexType name="RecordType">
          <xsd:sequence>
               <xsd:element name="Entry"
type="udap:RecordEntryType" minOccurs="0" maxOccurs="unbounded"/>
          </xsd:sequence>
     </xsd:complexType>

     <!-- The RecordEntryType -->
     <xsd:complexType name="RecordEntryType">
          <xsd:sequence>
               <xsd:element name="TimeStamp" type="xsd:dateTime"
minOccurs="1" maxOccurs="1"/>
               <xsd:element name="State" type="udap:StateType"
minOccurs="1" maxOccurs="1"/>
               <xsd:element name="Resource"
type="udap:ResourceType" minOccurs="0" maxOccurs="unbounded"/>
               <xsd:element name="Context" type="udap:ContextType"
minOccurs="0" maxOccurs="unbounded"/>
               <xsd:element name="Dependency"
type="udap:DependencyType" minOccurs="0" maxOccurs="unbounded"/>
```

```
          </xsd:sequence>
          <xsd:attribute name="Category"
type="udap:RecordEntryCategoryType"/>
     </xsd:complexType>

     <!-- Activity state -->
     <xsd:simpleType name="StateType">
          <xsd:restriction base="xsd:string">
               <xsd:enumeration value="pending"/>
               <xsd:enumeration value="running"/>
               <xsd:enumeration value="runningPreempted"/>
               <xsd:enumeration value="runningMigrating"/>
               <xsd:enumeration value="runningPaused"/>
               <xsd:enumeration value="cancelled"/>
               <xsd:enumeration value="failed"/>
               <xsd:enumeration value="finished"/>
          </xsd:restriction>
     </xsd:simpleType>

     <!-- The ResourceType defines the resource-related
requirements of a UDAP document -->
     <xsd:complexType name="ResourceType">
          <xsd:sequence>
               <xsd:any namespace="##other" minOccurs="0"
processContents="lax"/>
          </xsd:sequence>
     </xsd:complexType>

     <!-- The ContextType describes policy constraints, QoS,
security characteristics
           related to an activity -->
     <xsd:complexType name="ContextType">
          <xsd:sequence>
               <xsd:any namespace="##other" minOccurs="0"
processContents="lax"/>
          </xsd:sequence>
     </xsd:complexType>

     <!-- Activity dependency -->
     <xsd:complexType name="DependencyType">
          <xsd:sequence>
               <xsd:any namespace="##other" minOccurs="0"
processContents="lax"/>
          </xsd:sequence>
     </xsd:complexType>

     <!-- An entry in the record can be:
```

```
          - the original one, i.e. the one submitted to the UDAP
manager,
          - the current one, which represents the current valid
document, and
          - an archive one, provided for monitoring and
persistence. -->
     <xsd:simpleType name="RecordEntryCategoryType">
          <xsd:restriction base="xsd:string">
               <xsd:enumeration value="original"/>
               <xsd:enumeration value="current"/>
               <xsd:enumeration value="archive"/>
          </xsd:restriction>
     </xsd:simpleType>

     <!-- The ResultType -->
     <xsd:complexType name="ResultType">
          <xsd:sequence>
               <xsd:any namespace="##other" minOccurs="0"
processContents="lax"/>
          </xsd:sequence>
     </xsd:complexType>

</xsd:schema>
```

4 References

[1] D. Snelling, M. Fisher, A. Basermann, F. Wray, Ph. Wieder, M. Surridge, NextGRID Vision and Architecture White Paper V5, NextGRID, May 2007

[2] I. Foster, T. Maguire, D. Snelling, OGSA WSRF Basic Profile 1.0, Public Review Draft, April 26, 2006, https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.ogsa-wg/docman.root.working_drafts.basic_profile_1_0/doc13542/43

[3] T. Mori, F. Siebenlist, OGSA Basic Security Profile – Core, Public Comment Draft February 9, 2006, https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.ogsa-wg/docman.root.working_drafts.security_profile_1_0/doc13561/7

[4] T. Mori, F. Siebenlist, OGSA Basic Security Profile – Secure Channel, Public Comment Draft February 9, 2006, https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.ogsa-wg/docman.root.working_drafts.security_profile_1_0/doc13560/18

[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.ws-i.org/Profiles/BasicProfile-1.1.html

[7] D. Box and F. Curbera (ed.): Web Services Addressing 1.0 – Core (WS-Addressing), W3C Last Call, 31 March 2005. http://www.w3.org/TR/2005/WD-ws-addr-core-20050331

[8] S. Graham and J. Treadwell (ed.) Web Services Resource Properties 1.2 (WS-ResourceProperties), OASIS Standard, 1 April 2006. http://docs.oasis-open.org/wsrf/wsrf-ws_resource_properties-1.2-spec-os.pdf

[9] S. Graham, D. Hull and B. Murray (ed.) Web Services Base Notification 1.3 (WS-BaseNotification), OASIS Public Review Draft 02, 28 November 2005. http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-pr-02.pdf

[10] V. Li and D. Snelling (eds.), NextGRID Basic Profile, V1.0, September 2006, http://www.nextgrid.org/basicprofile10.pdf