ws-prov-link

Authors:
Steve Munroe, U. Southampton
Paul Groth, U. Southampton
Sheng Jiang, U. Southampton
Simon Miles, U. Southampton
Victor Tan, U. Southampton
Luc Moreau, U. Southampton
John Ibbotson, IBM
Javier Vazquez, UPC

August 24, 2006

A WS-Addressing Profile for Distributed Process Documentation

Status of this Memo

This document provides the specification of a data model for distributed process documentation and has the status of a working draft. It does not define any standards or technical recommendations. Distribution is unlimited.

Copyright Notice

Abstract

Process documentation can often be distributed across different provenance stores. To enable the discovery of related process documentation, a mechanism is required to link disparate but related process documentation to enable the effective collection of such documentation in order to answer provenance queries. This document represents a WS-addressing profile on distributed process documentation that provides mechanisms to solve this problem.

Contents

1 Introduction

Using the provenance recording protocol [GTM$^+$06] actors may record p-assertions to any number of different provenance stores. This means that the documentation of a process [MGJ$^+$06] that led to a result can be distributed at different locations. Given this, there must be some mechanism to retrieve and reconstruct these p-assertions in order to validate, visualise or replay the represented process. To facilitate such retrieval, the notion of a *link* is introduced which, intuitively, is a pointer to a provenance store. This document defines the schema necessary to represent linking for distributed process documentation. The XML document that describes this model is presented in Appendix A.

1.1 Goals and Requirements

The goal of this document is to define a linking mechanism that allows distributed process documentation to be discovered by queriers [MMG$^+$06]. This document exists as part of a family of documents specifying provenance. It is most closely associated to [MMG$^+$06], which describes how queries over process documentation can be achieved.

1.1.1 Requirements

In meeting this goal, this document must address the following requirements:

- Define the schema for linking different but related actor views for a given interaction.
- Define the schema to define how to locate p-assertions [MGJ$^+$06] that appear as objects in a relation.

1.1.2 Non-Requirements

This document does not intend to meet the following requirements:

- Supply definitions and scope of data provenance. This is covered in [TMG$^+$06c].
- Supply a model for the transformation of process documentation — called documentation styles. This aspect of data provenance is covered in [TMG$^+$06a].
- Supply a model querying process documentation. This aspect of data provenance is described in [MMG$^+$06].
- Supply a model for Provenance security. This aspect of data provenance is described in [TMG$^+$06b].

2 Terminology and Notation

All definitions for the concepts and structures found within this document can be found in [TGJ+06].

2.1 XML Namespaces

The XML Namespace URI that MUST be used by implementations of this specification is: `http://www.pasoa.org/schemas/version023s1/PLinks.xsd`

Table 1 lists XML namespaces that are used in this specification. The choice of any namespace prefix is arbitrary and not semantically significant.

Prefix	XML Namespace	Specification(s)
pl	`http://www.pasoa.org/schemas/version023s1/PLinks.xsd`	[PLinks]
wsa	`http://schemas.xmlsoap.org/ws/2004/03/addressing`	[WS-addressing]
xs	`http://www.w3.org/2001/XMLSchema`	[XMLSchema]

Table 1: Prefixes and XML Namespaces used in this specification

2.2 Notational Conventions

The keywords "MUST ", "MUSTNOT ", "REQUIRED ", "SHALL ", "SHALLNOT ", "SHOULD ", "SHOULDNOT ", "RECOMMENDED ", "MAY ", and "OPTIONAL " in this document are to be interpreted as described in [Bra97].

2.3 XML Schema Diagrams

This document adopts a graphical notation to describe XML Schema. Figure 1 gives an example of a small XML Schema displayed as a diagram, which is now explained with reference to the figure.

Figure 1 defines the structure of type `ts:Test`. The type Test contains a sequence of elements, which we now detail. One element in the sequence is `ts:testName`, which can be any type and must occur once and only once in an instance of `ts:Test`. `ts:Name` is followed by element `ts:testNumber`, which must contain a string. The `ts:testNumber` element must occur at least once and can occur as many times as needed. This is denoted by the "1..unbounded" under the element. Finally, the sequence contains a choice between two elements, `ts:startTest` and `ts:stopTest`, either of which must contain a date.

Below is a simple of description of the XML Schema diagram format.

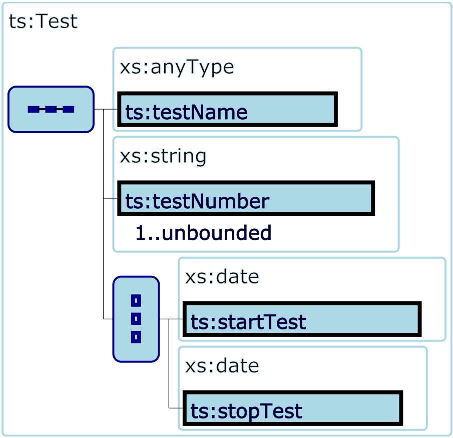
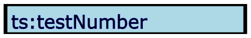
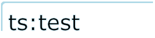


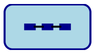
Figure 1: An example XML Schema diagram



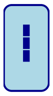
An element (instance) is represented by the qualified name of the element in the box. By default an element must occur once and only once. Where this restriction does not hold, the text “1..unbounded”, “0..unbounded”, “0..N”, “1..N” (where N is an integer) appears under the element box. The left hand number is the minimum occurrences of the element at the position in the XML document, the right hand number is the maximum (with “unbounded” for no maximum).



A complex type is denoted by a lightly marked box with the qualified name of the type at the top left. The structure of the type is given by the elements, types and control structures within the box.



A horizontal sequence of dots represents a sequence of elements or control structures, that must appear in an element conforming to the type in the surrounding type box.



A vertical sequence of dots represents a choice between elements or control structures, that must appear in an element conforming to the type in the surrounding type box.

2.4 XPath notation

In addition to the XML Schema diagrams, an XPath notation [W3C99] is used to identify each individual element in the specification along with its context, in order to describe precisely its meaning and use.

3 Linking

Process documentation may be distributed across any number of provenance stores, and must be retrieved in order to answer provenance queries. A querier, finding a fragment of process documentation, must be able to locate related fragments. This is achieved by providing explicit *links* to locations where other, related process documentation can be found.

This document provides a linking mechanism to achieve this using the Web-Service addressing's endpoint reference mechanism [GHR06]. Thus, provenance stores are modelled as Web Services that are locatable at an endpoint reference.

Two forms of linking are required: Linking together process documentation that represent different *views* of an interaction, and providing links to data items that play the role of *objects* in a relation. Each mechanism is described below.

3.1 Linking Together Different Views of an Interaction

The different views of actors in an interaction can be linked together by providing links from one set of interaction p-assertions to the location of the associated set (i.e. the set belonging to the other actor in the interaction). This is achieved by providing textttviewLinks, which are links that can be found within p-assertions pointing to the provenance store holding the other view of an interaction.

Linking information is passed between actors as part of an interaction's meta-data, and may be contained within a *p-header* (a full description of which is given in [MTG$^+$06]). The p-header allows for the provision of extra information about an interaction in its `interactionMetaData` element, which contains an extensibility element that can be used to provide application specific information within interactions, and is shown in the xpath expression below.

```
/ps:pheader/ps:interactionMetaData/xs:Any
```

One such use of this extensibility element is to provide the ability to connect process documentation that is spatially distributed via the use of the link mechanism. This can be achieved by redefining the `Any` element as a `viewLink`, resulting in the following xpath expression:

```
/ps:pheader/ps:interactionMetaData/pl:viewLink
```

This model is described by a schema document represented by Figure 2.

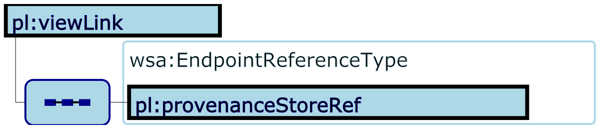


Figure 2: A WS-addressing endpoint reference pointing to the location of another view on an interaction

The document content is further described below.

`/pl:viewLink`

This element represents a link to the location of an actor's view of an interaction.

`/pl:viewLink/pl:provenanceStoreRef`

A `viewLink` is instantiated as a WS-addressing `EndpointReference` pointing to the provenance store where the associated view for an interaction can be found.

The instantiation of an EndpointReference can be achieved by utilising WS-Addressing's `ReferenceParameters` element. This is described in Section 4.

3.2 Linking to Objects of a Relationship P-Assertion

Relationship p-assertions allow uni-directional relationships between both messages and data to be expressed. Relationship p-assertions are modelled as one-to-many triples between data or messages, where the domain of a relationship is called the *subject* and the range is the set of *objects* (the complete model is presented in the Data Model for Process Documentation document [MGJ$^+$06]). The triple consists of a subject identifier (`subjectId`), a relation, and several object identifiers (`objectIds`).

Relationship p-assertions express causal relationships, where the subject of the relationship is a data item in a sent message, i.e. an output, and the objects are entities in messages received by the same actor, i.e. inputs, where the inputs had caused the output to be as it is. An `objectId` element contains several child elements and is fully defined in [MGJ$^+$06]. In distributed process documentation scenarios, given that an actor can use multiple provenance stores to document its involvement in a process, an `objectId` element will contain an `objectLink` element giving the address of the provenance store in which the data item acting as an object in a relationship is kept. In the complete data model for process documentation presented in [MGJ$^+$06], this element instantiates the element in the `objectId` structure contained within a relationship p-assertion as shown by the xpath below:

```
/ps:relationshipPAssertion/ps:objectId/xs:Any
```

The Any element is then instantiated in the WS-addressing profile by an object link as follows:

```
/ps:relationshipPAssertion/ps:objectId/pl:objectLink
```

Object links are modelled as shown in Figure 3. The model is further described as follows.

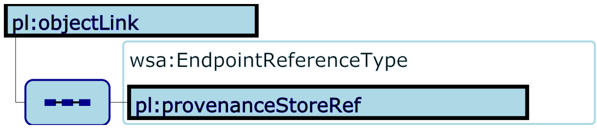


Figure 3: An WS-addressing endpoint reference pointing to the location of an object in a relation

`/pl:objectLink`

This element represents a link to the location of a data item that acts as an object in a relationship p-assertion.

`/pl:objectlink/pl:provenanceStoreRef`

The link to the location of a data item acting as an object in a relationship p-assertion is instantiated as a WS-addressing `EndpointReference` pointing to the provenance store where the data item can be found.

4 Identifying Provenance Store Service Ports

Provenance stores offer different functionality via their public interfaces exposed as ports. When following a link to a provenance store it is necessary to be able to identify the correct port in order to perform the requested operation. The WS-addressing's EndpointReference provides a number of characteristics to enable such identification via `ReferenceParameters`. The `ReferenceParameter` element is defined as an xs:Any, thus it allows for extensibility and enables the specification of a `PortContext` type to provide a way to identify which port is to be used in a communication with a provenance store. Below is shown an instance of a `PortContext` type that defines a key-value pair. The key is given as the portName and the value is given as the context of the port, i.e. the identifier used to identify the port.

```
<pl:portContext>
    <pl:portName>XQuery</pl:portName>
    <pl:context>xquery</pl:context>
</pl:portContext>
```

A ProvenanceStoreRef is now defined in which the ReferenceParameters element is used to identify two ports, one for recording and one for querying, using the portContext type.

```
<wsa:EndpointReference>
    <wsa:Address>http://www.pasoa.org/provenancestore1/</wsa:Address>

    <wsa:ReferenceParameters>
        <pl:portContext>
            <pl:portName>Record</pl:portName>
            <pl:context>myrecord</pl:context>
       </pl:portContext>
       <pl:portContext>
            <pl:portName>XQuery</pl:portName>
            <pl:context>superxquery</pl:context>
      </pl:portContext>
      ...
    </wsa:ReferenceParameters>
</wsa:EndpointReference>
```

In this way connecting to a provenance store entails specifying the EndpointReference of the provenance store plus the context for the required port. The following example shows an EndpointReference that identifies a record port.

```
http://www.pasoa.org/provenancestore1/myrecord
```

The formal model of this is further described below.

/pl:provenanceStoreRef/wsa:ReferenceParameters/pl:portContext

This element holds the information about one of this provenance store's ports.

/pl:provenanceStoreRef/wsa:ReferenceParameters/pl:portContext/pl:portName

The name of the port for this provenance store.

/pl:provenanceStoreRef/wsa:ReferenceParameters/pl:portContext/pl:context

The context of the specified port for this provenance store.

It should be noted that in cases where process documentation must be kept for long periods of time, it is recommended that the references used for machines and ports should use virtual URI's that can the n be mapped onto real IP addresses, since physical machines may be upgraded or changed over time.

5 Conclusion

This document has provided a WS-addressing profile for the distribution of process documentation. It provides a detailed data model of a linking mechanism instantiated using WS-addressing `EndpointReferences`. This enables process documentation necessary to connect different views of an interaction to be found, and to enable data items acting as objects in a relationship p-assertion to be discovered.

Other forms of linking could be considered where process documentation is distributed but not stored in provenance stores. Specific profiles should be defined for each of these.

Appendix A

The following XML document describes the linking types and elements used in this document.

```
<?xml version="1.0" encoding="UTF-8"?> <xs:schema
xmlns:pl="http://www.pasoa.org/schemas/version023s1/distribution/PLinks.xsd"
                      xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
                    xmlns:xs="http://www.w3.org/2001/XMLSchema"
                    targetNamespace="http://www.pasoa.org/schemas/version023s1/PLinks.xsd"
                    elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:annotation>
        <xs:documentation>
            Defines extensions to the P-Structure that allow for linking between
            provenance stores from within the p-structure.
            Author: Paul Groth
            Last Modified:15 May 2006

                  Copyright (c) 2006 University of Southampton
                  See pasoalicense.txt for license information.
                  http://www.opensource.org/licenses/mit-license.php

              </xs:documentation>
    </xs:annotation>

      <xs:import namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing"
        schemaLocation="../wsaddressing.xsd"/>

    <xs:element name="viewLink" type="pl:Link">
        <xs:annotation>
            <xs:documentation>
                 A link between this provenance store and another provenance
                 store. Where the provenance store linked to will contain
                 the other view of the interaction record.
             </xs:documentation>
        </xs:annotation>
    </xs:element>

    <xs:element name="objectLink" type="pl:Link">
        <xs:annotation>
            <xs:documentation>
                 A link between this provenance store and another provenance
                 store. Where the provenance store linked to will contain the p-assertion identified by the
                 p-assertion data key in the object id where this link is located.
             </xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:complexType name="Link">
        <xs:sequence>
            <xs:element name="provenanceStoreRef" type="wsa:EndpointReferenceType">
                <xs:annotation>
                    <xs:documentation>
                        The actual link to the Provenance Store.
                    </xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

References

[Bra97] Scott Bradner. Key words for use in RFCs to indicate requirement levels. http://www.ietf.org/rfc/rfc2119.txt, 1997.

[GHR06] Martin Gudgin, Marc Hadley, and Tony Rogers. Web services addressing 1.0 - core w3c recommendation 9 may 2006. http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/, 2006.

[GTM$^+$06] Paul Groth, Victor Tan, Steve Munroe, Sheng Jiang, Simon Miles, and Luc Moreau. Process Documentation Recording Protocol. Technical report, University of Southampton, June 2006.

[MGJ$^+$06] Steve Munroe, Paul Groth, Sheng Jiang, Simon Miles, Victor Tan, and Luc Moreau. Data model for Process Documentation. Technical report, University of Southampton, June 2006.

[MMG$^+$06] Simon Miles, Steve Munroe, Paul Groth, Sheng Jiang, Victor Tan, John Ibbotson, and Luc Moreau. Process Documentation Query Protocol. Technical report, University of Southampton, June 2006.

[MTG$^+$06] Steve Munroe, Victor Tan, Paul Groth, Sheng Jiang, Simon Miles, and Luc Moreau. A SOAP Binding For Process Documentation. Technical report, University of Southampton, June 2006.

[TGJ$^+$06] Victor Tan, Paul Groth, Sheng Jiang, Simon Miles, Steve Munroe, and Luc Moreau. WS Provenance Glossary. Technical report, Electronics and Computer Science, University of Southampton, 2006.

[TMG$^+$06a] Victor Tan, Steve Munroe, Paul Groth, Sheng Jiang, Simon Miles, and Luc Moreau. Basic Transformation Profile for Documentation Style. Technical report, University of Southampton, June 2006.

[TMG$^+$06b] Victor Tan, Steve Munroe, Paul Groth, Sheng Jiang, Simon Miles, and Luc Moreau. Data Model for Provenance Security. Technical report, University of Southampton, June 2006.

[TMG$^+$06c] Victor Tan, Steve Munroe, Paul Groth, Sheng Jiang, Simon Miles, and Luc Moreau. The Provenance Standardisation Vision. Technical report, University of Southampton, June 2006.

[W3C99] W3C. XML Path Language (XPath) Version 1.0. W3C Recommendation 16 November 1999. http://www.w3.org/TR/xpath, 1999.