ws-prov-xquery

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

July 28, 2006

Process Documentation Query Protocol

Status of this Memo

This document provides information to the community regarding the specification of a protocol for querying the process documentation contained in a provenance store and has the status of a working draft. The document represents a proposal to be used as input to a standardisation process. It is possible that the contents will change during such a process, but should nonetheless provide a stable reference for discussion. It does not define any standards or technical recommendations. Distribution is unlimited.

Copyright Notice

Abstract

A related document [MGJ+06] defines schemas to be used for documentation about the execution of a process, *process documentation*, and introduce a *provenance store*, a type of Web Service with the capability for storing and giving access to process documentation. In particular, process documentation has a defined schema, the *p-structure*, which clients of a provenance store can navigate in queries to extract particular pieces of process documentation. In this document, we specify a protocol by which a querying actor and provenance store can communicate in performing a process documentation query. This primarily takes the form of an abstract WSDL interface defining messages to be accepted and produced by a provenance store.

Contents

1 Introduction

Every provenance store supplies a Web Service interface for querying process documentation by navigating the p-structure. It has a single operation, *query*, that takes an XQuery expression as input and returns a set of XML documents, whose schemas are dependent on the XQuery, as result. This document defines the schema for the request and response messages. The WSDL 1.1 description of the interface is given in Appendix B.

1.1 Goals and Requirements

The goal of this document is to define the protocol for querying process documentation contained in a provenance store using XQuery.

1.1.1 Requirements

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

- Define the schema of the query request message sent to the provenance store.
- Give the additional requirements on the XQuery expression for navigating process documentation.
- Define the behaviour of a provenance store on receiving a query request.
- Define the schema of the query response message returned by the provenance store.

1.1.2 Non-Requirements

This document specifies a synchronous version of the query protocol. Other documents may specify asynchronous querying.

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/xquery/XQuery.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)
xq	`http://www.pasoa.org/schemas/version023s1/xquery/XQuery.xsd`	[PDXQuery]
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 documents 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.

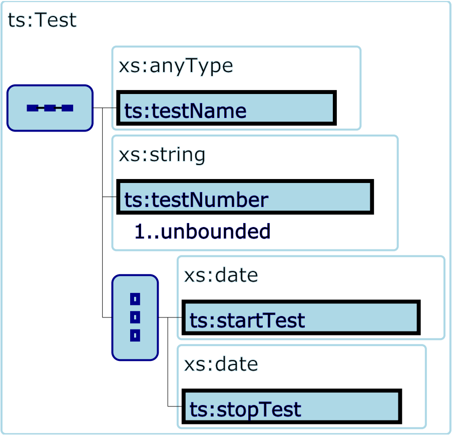
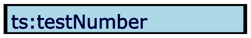


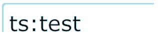
Figure 1: An example XML Schema diagram

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.

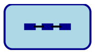
Below is a simple of description of each of the parts of the XML Schema diagram format.



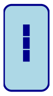
An element (instance) is represented by the qualified name of the element in the box. By default an element must occur once and only once. Where this restriction does not hold, the text "1..unbounded", "0..unbounded", "0..N", "1..N" (where N is an integer) appears under the element box. The left hand 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 Process Documentation Query

We specify below the request document schema, query expression form, provenance store behaviour and response document schema for a process documentation query. The full schema document, in which request and response message structures are defined, is given in Appendix A. The WSDL 1.1 description of the interface taking and producing these messages is given in Appendix B.

3.1 Request

A process documentation query request is a message sent by a *querying actor* to a *provenance store* to perform an XQuery over the contents of the store. It is instantiated as a document conforming to the schema depicted in Figure 2.

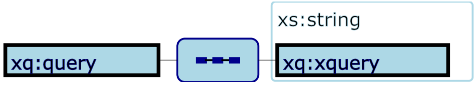


Figure 2: Process Documentation Query Request

The document contents are described in detail as follows.

`/xq:query`

> This element represents a request for a provenance store to perform an XQuery operation over its contents, and return the results.

`/xq:query/xq:xquery`

> This element contains the XQuery expression itself, conforming to the details given in the next section.

3.2 Query Expressions

The query operation treats the contents of a provenance store as a single XML document conforming to the p-structure XML Schema. The expression should follow the XQuery 1.0 specification [BCF+06], but with the following caveats.

First, the result of evaluating the expression should be one or more XML documents, i.e. not just string, integer or other literal values. This is to allow the results to be returned unambiguously in a response document.

Second, as an XQuery expression can be performed over multiple documents or document nodes, queriers need to specify which document they wish to query. By default all provenance stores implementing the process documentation query interface must recognise the pre-bound variable $ns:pstruct in an XQuery expression, where ns is any prefix bound to the namespace `http://www.pasoa.org/schemas/version023s1/PStruct.xsd`. This variable should be bound to a document node that follows the schema of the *pstruct* element defined in the p-structure data model [MGJ+06], with the expectation that the contents of the document is a p-structure containing the full contents of the provenance store. Particular implementations may provide bindings to other documents where applicable.

Specific provenance-store implementation may provide built-in XQuery functions for common or advanced traversal of the store contents.

3.2.1 Examples

The following example returns the whole provenance store contents as a p-structure.

```
declare namespace ps = "http://www.pasoa.org/schemas/version023s1/PStruct.xsd";

$ps:pstruct
```

The following example returns an HTML unordered list summarising the relationship p-assertions in a store. The format of each item is *subject relation object1 object2...* The subjects and objects are identified by the interaction ID of the interaction record in which the data item is contained.

```
declare namespace ps = "http://www.pasoa.org/schemas/version023s1/PStruct.xsd";

<UL> {
  for $r in $ps:pstruct//ps:relationshipPAssertion
  return <LI> {
    (data($r/../../ps:interactionKey/ps:interactionId),
     ' ', data($r/ps:relation), ' ')}
    {for $o in $r/ps:objectId return (data($o/ps:interactionKey/ps:interactionId), ' ')
  }</LI>
}</UL>
```

3.3 Behaviour

On receiving a process documentation query request, a provenance store is expected to evaluate the query and synchronously return the result in a response message. The result of the query is exactly that document returned by the XQuery submitted.

3.4 Response

A process documentation query response is sent by a provenance store to the querying actor that issued the corresponding request. It is instantiated as a document conforming to the schema depicted in Figure 3.

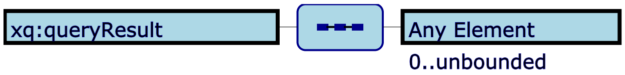


Figure 3: Process Documentation Query Response

The document contents are described in detail as follows.

`/xq:queryResult`

This element represents the results of an XQuery.

`/xq:queryResult/xs:any`

Each child element contains a document fragment produced by evaluating the request's XQuery expression.

3.5 Faults

To be specified in conjunction with IBM.

4 Default Port Name

In order to aid addressing of provenance stores, we require that each interface that can be assumed to be present in every provenance store be given a default port context name [MTG$^+$06], i.e. the last part of the URL addressing the port supporting that interface. For the process documentation query interface, the default port name is `xquery`.

5 Security Considerations

This specification defines the process documentation query request and response messages for any provenance store supporting the process documentation query interface. In this context, there are two categories of security aspects that need to be considered: (a) securing the message exchanges and (b) securing the provenance store contents.

5.1 Securing Message Exchanges

When messages are exchanged between a querier and provenance store in a process documentation query, it is recommended that the communication be secured using the mechanisms described in WS-Security [Var04]. In order to properly secure messages, the message body (query expression or results) and all relevant headers need to be included in the digital signature so as to prove the integrity of the message. In the event that a querier frequently performs process documentation queries on a store it is recommended that a security context be established using the mechanisms described in WS-Trust [Var05b] and WS-SecureConversation [Var05a], allowing for potentially more efficient means of authentication.

5.2 Securing Provenance Store Contents

Since this specification defines a mechanism to retrieve the contents of provenance stores, security policies should be established that ensure that only authorized queriers can access the p-assertions.

A Process Documentation Query Schema

Below we give the full schema for process documentation queries.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://www.pasoa.org/schemas/version023s1/xquery/XQuery.xsd"
  xmlns:xq="http://www.pasoa.org/schemas/version023s1/xquery/XQuery.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.pasoa.org/schemas/version023s1/xquery/XQuery.xsd"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation>
      An XQuery interface for Web Services
      Author: Simon Miles
      Last Modified: 28 Feb 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:element name="query" type="Query">
    <xs:annotation>
      <xs:documentation>Query operation</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="Query">
    <xs:annotation>
      <xs:documentation>Query operation type</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="xquery" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="queryResult" type="QueryResult">
    <xs:annotation>
      <xs:documentation>Query results</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="QueryResult">
    <xs:annotation>
      <xs:documentation>Query results type</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

B Process Documentation Query WSDL

Below we give the WSDL document for process documentation query and response messages.

```
<?xml version="1.0"?>
<definitions name="XQuery"
  targetNamespace="http://www.pasoa.org/schemas/version023s1/xquery/XQuery.wsdl"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://www.pasoa.org/schemas/version023s1/xquery/XQuery.wsdl"
  xmlns:xq="http://www.pasoa.org/schemas/version023s1/xquery/XQuery.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns="http://schemas.xmlsoap.org/wsdl/">
  <documentation>
    An XQuery interface for Web Services
    Author: Simon Miles
    Last Modified: 28 Feb 2006
    Copyright (c) 2006 University of Southampton
    See pasoalicense.txt for license information.
    http://www.opensource.org/licenses/mit-license.php
  </documentation>
  <import namespace="http://www.pasoa.org/schemas/version023s1/xquery/XQuery.xsd" location="./XQuery.xsd"/>
  <message name="Query">
    <part name="body" element="xq:query"/>
  </message>
  <message name="QueryAck">
    <part name="body" element="xq:queryAck"/>
  </message>
  <portType name="XQueryPortType">
    <operation name="Query">
      <input message="tns:Query"/>
      <output message="tns:QueryAck"/>
    </operation>
  </portType>
</definitions>
```

References

[BCF^{+}06] Scott Boag, Don Chamberlin, Mary F. Fernandez, Daniela Florescu, Jonathan Robie, and Jerome Simeon. XQuery 1.0: An XML Query Language. http://www.w3.org/TR/xquery, 2006.

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

[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.

[MTG^{+}06] Steve Munroe, Victor Tan, Paul Groth, Sheng Jiang, Simon Miles, and Luc Moreau. WSRF Data Model Profile for Distributed Provenance. 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.

[Var04] Various authors. Web Services Security. http://www-128.ibm.com/developerworks/library/specification/ws-secure/, 2004.

[Var05a] Various authors. Web Services Secure Conversation Language. http://www-128.ibm.com/developerworks/webservices/library/specification/ws-secon/, 2005.

[Var05b] Various authors. Web Services Trust Language. http://www-128.ibm.com/developerworks/library/specification/ws-trust/, 2005.

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