

ws-prov-soap

**Authors:**

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

August 24, 2006

# A SOAP Binding for Provenance P-headers

## Status of this Memo

This document provides information to the community regarding the specification of a data model for process documentation used to describe a SOAP binding of the process documentation model and has the status of a working draft. It does not define any standards or technical recommendations. Distribution is unlimited.

## Copyright Notice

Copyright 2006.

## Abstract

This document describes a SOAP binding for the process documentation p-header. It presents a specification of the p-header and can be considered an extension of the process documentation data model presented in [MGJ<sup>+</sup>06].

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Goals and Requirements . . . . .	3
1.1.1	Requirements . . . . .	3
<b>2</b>	<b>Terminology and Notation</b>	<b>4</b>
2.1	XML Namespaces . . . . .	4
2.2	Notational Conventions . . . . .	4
2.3	XML Schema Diagrams . . . . .	4
2.4	XPath notation . . . . .	6
<b>3</b>	<b>The P-Header</b>	<b>6</b>
3.1	The P-header's location in a SOAP Message . . . . .	7
<b>4</b>	<b>Conclusion</b>	<b>8</b>

# 1 Introduction

In order for p-assertions to be created, asserting actors need to identify which process they are making an assertion about, which requires some *shared context* between asserting actors. As it is application actors that make assertions, a further obligation is placed on them to pass context information between each other regarding the process being executed. As this would often be achieved by putting the context information in the header of an application message, such as a SOAP message [Mit03], this information is termed the *p-header*, defined as follows.

**Definition 1 (p-header)** *The p-header of an interaction is provenance-related contextual information, sent along with the interaction's message. □*

In practise, the p-header can contain an identifier for the interaction to which the context information applies and the locations of provenance stores where p-assertions documenting the same process are stored. Additionally, the p-header can contain a set of *tracers*, which are used to demarcate where one process starts and ends. A tracer is a token added to a p-header by an application actor, where the same tracer is added to the p-headers of all interactions in the same process by the same application actor. Additionally, where a tracer is included in the p-header of a message received by an application actor, that actor is obliged to copy the tracer into the p-header of all interactions within the same process. Using tracers, a querying actor can determine which interactions were part of a single process, because their p-headers will all contain the same tracer, and whether one process is contained within another, because the tracers of the former's interactions will be a subset of the tracers of the latter's interactions. This document presents a specification of the data model for the p-header.

A full overview document is available that describes the vision for the standardisation effort [TMG<sup>+</sup>06].

## 1.1 Goals and Requirements

The goal of this document is to define an open, interoperable model for the p-header and its location in a SOAP header.

### 1.1.1 Requirements

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

- Define the data items necessary for the p-header and their logical organisation.
- Locate the p-header within a SOAP header.

- Provide the basis for an open, interoperable set of standards.
- Provide extensibility for more sophisticated and/or currently unanticipated scenarios.

## 2 Terminology and Notation

All definitions for the concepts and structures found within this document can be found in [TGJ<sup>+</sup>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/PStruct.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)
ph	<code>http://www.pasoa.org/schemas/version023s1/PHeader.xsd</code>	[P-Header]
xs	<code>http://www.w3.org/2001/XMLSchema</code>	[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 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.

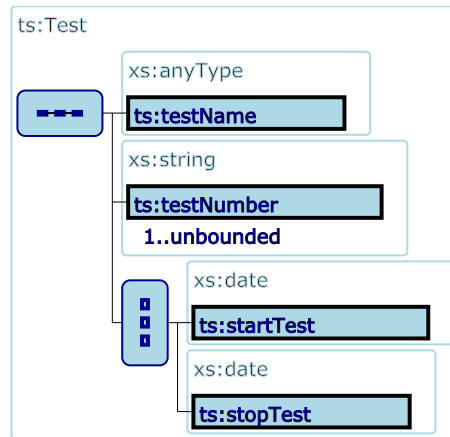


Figure 1: An example XML Schema diagram

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 The P-Header

The p-header's intended functionality is to provide a way for actors to pass meta-information about interactions. Though the model is technology neutral, we focus on the use of SOAP messages as the vehicle to pass around application messages and thus forms the basis of the interaction model in this document.

The p-header is constructed and placed within the header of a SOAP message. The information contained within a p-header is of three types. First, the p-header contains an interaction key. This key is used to link p-assertions made at different times and by different actors to a specific interaction. Any p-assertion made about a specific interaction should use the same interaction key. This enables queriers to later come along and find all those p-assertions about a given interaction by examining their interaction keys and selecting all those that have the same one. Thus, when an actor sends a message to another actor it must include an interaction key so that the receiver can use this within its own p-assertions about the interaction to point to that interaction. Along with the interaction key, a p-header contains an optional set of interaction metadata (see [MGJ<sup>+</sup>06] for the detailed specification of the elements defined within the p-header, i.e. `InteractionKey`, `InteractionMetada` and `InteractionContext`). This data contains provenance related information about the interaction such as pointers to the location where p-assertions made by the actor are stored, any tracers that are being used to denote a process and any other application specific information deemed necessary. The information described so far all refer to the current interaction, i.e. the contents of the SOAP message to which the p-header is attached.

The final form of information contained within a p-header is an optional set of interaction contexts relating to other interactions, i.e. interactions other than the one that the above discussed interaction key and interaction metadata are about. This provides the means to propagate view and object links around. This information includes the interaction key of the interaction being referred to, and other information relating to where p-assertions relating to this information are stored, i.e. interaction metadata about the interaction. The model of the p-header is shown in Figure 2.

The contents of a p-header are further described as follows:

`/ph:pheader`

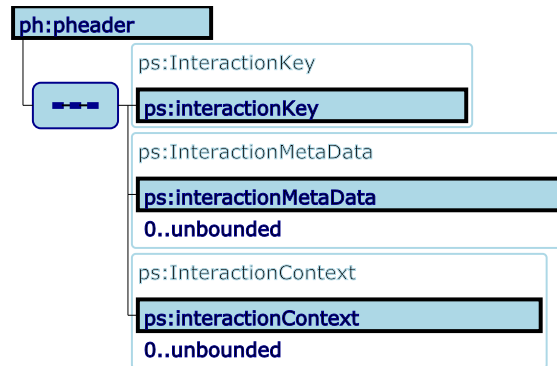


Figure 2: Model of the PHeader.

The root element of the p-header. It contains a sequence of three components that enable actors to add interaction context elements to messages: an interaction key, an OPTIONAL set of interaction metadata about the identified interaction, and an OPTIONAL set of interaction contexts about other interactions.

#### `/ph:pheader/ps:interactionKey`

The intent of this component is to uniquely identify the interaction whose message this p-header is attached. The full definition of `interactionKey` is given in [MGJ<sup>+</sup>06].

#### `/ph:pheader/ps:interactionMetaData`

The intent of this component is to hold meta data about the above identified interaction. The full description and formal definition of `interactionMetaData` is given in [MGJ<sup>+</sup>06].

#### `/ph:pheader/ps:interactionContext`

The intent of this component is to provide information about the context of the above identified interaction by identifying those other interactions that are relevant to this one. The full description and formal definition of `interactionContext` is given in [MGJ<sup>+</sup>06].

### 3.1 The P-header's location in a SOAP Message

Given the full specification of the p-header as above, its location within a SOAP message can be described. Figure 3 shows a SOAP message and its component parts. A SOAP message contains an envelope that, as its name would suggest, serves as a container for the other elements of the SOAP message. Those other

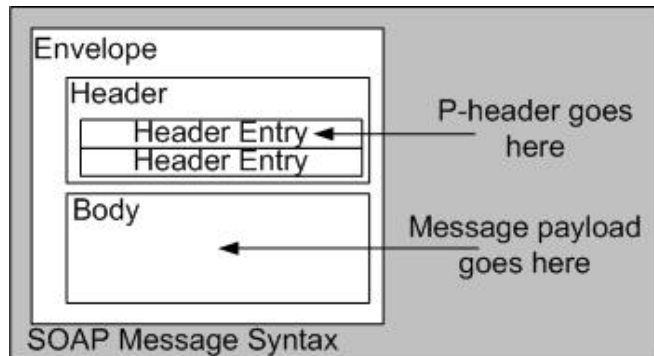


Figure 3: The p-header’s location in a SOAP message

elements comprise the message body that contains the application specific information that is to be passed, as well as the SOAP header.

The SOAP header is used to supply extra information about the message that does not properly belong in the message body. For example, the information in the message body may need to be linked to other messages — it is the purpose of the SOAP header to carry such information. Since the p-header is designed to carry context specific information about p-assertions, that is, extra information about the enclosed p-assertions, then its natural place within a SOAP message is in the SOAP header. This is shown the figure, where the location of the p-header is shown as being within one of the SOAP header entries.

The XML snippet below shows a soap message with a p-header and its elements enclosed within the SOAP message’s Header element. The namespaces shown under the Envelope element provide the prefixes for each of the parts of the p-header and the application data contained within the Body element.

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    soap:encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/">
    xmlns:ph="http://www.pasoa.org/schemas/version023s1/PHeader.xsd">
    xmlns:ps="http://www.pasoa.org/schemas/version023s1/PStructure.xsd">
    xmlns:ap="http://www.application/data/Data.xsd">
    <soap:Header>
    <ph:pheader>
    <ps:interactionKey>...</ps:interactionKey>
    <ps:interactionMetaData>...</ps:interactionMetaData>
    <ps:interactonContext>...</ps:interactionContext>
    </ph:pheader>
    </soap:Header>
    <soap:Body>
    <ap:applicationData>010011100111</ap:applicationData>
    </soap:Body>
</soap:Envelope>
```

## 4 Conclusion

In this document the specification of the process documentation p-header was presented. The p-header is used to transfer provenance-based context information within the headers of application specific messaging protocols such as SOAP. This enables actors to relate messages to other messages and thus helps to bring together different views on a given interaction.

## Appendix A

The following illustrates the p-header types and elements used in this document.

```
<?xml version="1.0" encoding="UTF-8"?> <xs:schema
targetNamespace="http://www.pasoa.org/schemas/version023s1/PHeader.xsd"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ph="http://www.pasoa.org/schemas/version023s1/PHeader.xsd"
  xmlns:ps="http://www.pasoa.org/schemas/version023s1/PStruct.xsd">

  <xs:annotation>
    <xs:documentation>
      The PHeader schema
      Author: Paul Groth

      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://www.pasoa.org/schemas/version023s1/PStruct.xsd"
    schemaLocation="./PStruct.xsd"/>

  <xs:element name="pheader" type="ph:PHeader"/>

  <xs:complexType name="PHeader">
    <xs:annotation>
      <xs:documentation>Provenance Specific Header Information</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element ref="ps:interactionKey" maxOccurs="1" minOccurs="1" />
      <xs:element ref="ps:interactionMetaData" maxOccurs="unbounded" minOccurs="0"/>
      <xs:element ref="ps:interactionContext" maxOccurs="unbounded" minOccurs="0"/>
    </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.
- [MGJ<sup>+</sup>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.
- [Mit03] N. Mitra. Soap version 1.2 part 0: Primer. <http://www.w3.org/TR/soap12-part0/>, 2003.
- [TGJ<sup>+</sup>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<sup>+</sup>06] 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.