Title: Functional Prototype

Author: Neil Hardman (IBM)

Reviewers: John Ibbotson (IBM)

Identifier: D9.3.1

Type: Deliverable

Version: 1.0

Date: 20th September 2005

Status: Public

Summary

The purpose of this document is to describe the first functional prototype for the Provenance project. It covers the design decisions, functionality and dependencies for the prototype. The first functional prototype does not include any security or scalability support.

The PROVENANCE project receives research funding from the European Community's Sixth Framework Programme

Members of the PROVENANCE consortium:

- IBM United Kingdom Limited United Kingdom
- University of Southampton United Kingdom
- University of Wales, Cardiff United Kingdom
- Deutsches Zentrum fur Luft- und Raumfahrt s.V. Germany
- Universitat Politecnica de Catalunya Spain
- Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézet Hungary

The PROVENANCE project receives research funding from the European Community's Sixth Framework Programme

Foreword

This document has been edited by Neil Hardman (IBM).

The PROVENANCE project receives research funding from the European Community's Sixth Framework Programme

Table of Contents

The PROVENANCE project receives research funding from the European Community's Sixth Framework Programme

1 Introduction

1.1 Purpose of the Document

The first deliverable of the Grid Provenance code builds upon the work done on the pre-prototype and also the work of the PASOA (www.pasoa.org) project. The aim of this first deliverable was to take the research work of PASOA and to move it into a formal implementation, to provide the basis for ongoing research and development. The task of building any software deliverable is considerable as it encompasses so many technologies and skills and the exploitation of Open Source pre-requisite products also give us their own challenges in terms of software compatibility.

1.2 Overview of the Document

This document provides a view of the development branch of the Grid Provenance project. It discusses some of the decisions that were taken in producing the current functional prototype and it also details the components in the delivery package and gives a high level view of what they do.

The PROVENANCE project receives research funding from the European Community's Sixth Framework Programme

2 Provenance Functional Prototype Design

2.1 Overview

The functional prototype implements the storage interface for recording Provenance p-assertions as defined by the Provenance Aware Service Oriented Architecture (PASOA) WSDL interface descriptions. The interface operations are made available through a provenance store Web Service which stores p-assertions in an XML database using a set of Java classes. The prototype implements Web Services that provide record and query interfaces to the provenance store.

In addition to the functionality described above, we provide a provenance aware AXIS handler to seamlessly write interaction p-assertions to the provenance store and a command line query client to query the provenance store using XPath statements.

At present the functional prototype does not implement any security, scalability, management or high level client API functionality, however these functions are currently under development

2.2 Design Decisions

All the code developed is written in Java. This was chosen to allow easier integration with existing web service container implementations such as Apache Tomcat.

The functional prototype uses a number of pre-requisite components. We chose to use the Open Middleware Infrastructure Institute (OMII) (http://www.omii.ac.uk/) container (which includes Apache Tomcat and will eventually package OGSA-DAI as well).for this and subsequent deliverables.

The functional prototype uses a data abstraction layer to isolate the interfaces from the data store implementation. This allows the use of various underlying database technologies whilst keeping the implementation code constant.

The functional prototype should only use an XML database as its data store. This decision was taken because all data flowing between client and server will be packed in XML structures, and it was considered a sensible approach to speed the development process.

2.3 Design Objectives

A number of separate objectives needed to be met in order to make the deliverable:

1. The installation and use of pre-existing software to provide the service container and the data store for Grid Provenance. The latest version of OMII base was installed on a Linux machine and then a specialised version of OGSA-DAI (modified for OMII use) was installed within OMII.
2. The installation and use of a database that permits the storage and retrieval of XML data. There are a number of candidates including Xindice (http://xml.apache.org/xindice/), eXist (http://exist.sourceforge.net/) and IBM's DB2. For the development and testing we chose eXist as it is the most capable of the open source XML database implementations. The eXist database was accessed through the OGSA-DAI data access middleware.
3. The Grid Provenance code should exploit the web service interfaces that are described by the PASOA project. PASOA provides the WSDL and associated data structures to record three

The PROVENANCE project receives research funding from the European Community's Sixth Framework Programme

types of provenance information, namely: Interaction Provenance (IP), Relationship Provenance (RP) and Actor State Provenance (ASP).

2.4 Dependencies

The Grid Provenance functional prototype depends on a number of components as listed below. The decision to use these products was taken to ease the development load by allowing us to exploit the of work that others have used in developing their components. There is no point in re-inventing when we can reuse. These components also provide us with consistent interfaces that allow faster and more reliable software development.

1. OGSA-DAI WSI 1.1 for OMII 2 - http://www.ogsadai.org.uk
2. OMII 2.0 - http://www.omii.ac.uk/
3. eXist 1.0 Beta 2 - http://exist.sourceforge.net/
4. Apache Axis 1.2 RC3 - http://ws.apache.org/axis
5. Apache Tomcat 5.0.x - http://jakarta.apache.org/tomcat

2.4.1 PASOA

The PASOA project is complementary to the Grid Provenance project. It has defined the interfaces and protocol for recording p-assertions in a provenance store. PASOA has provided the WSDL interface descriptions which are the definitions used to automatically generate the Java beans that provide the provenance store interface implementation.

In addition to the interface beans, the Grid Provenance project provides a header bean that generates the SOAP header that is passed between actors and the provenance store. The layout of the header is specified by PASOA, however because it does not implement a Web Services interface, it cannot be described by WSDL and as a consequence cannot be automatically generated, so the bean to serialize and de-serialize the header information was manually developed.

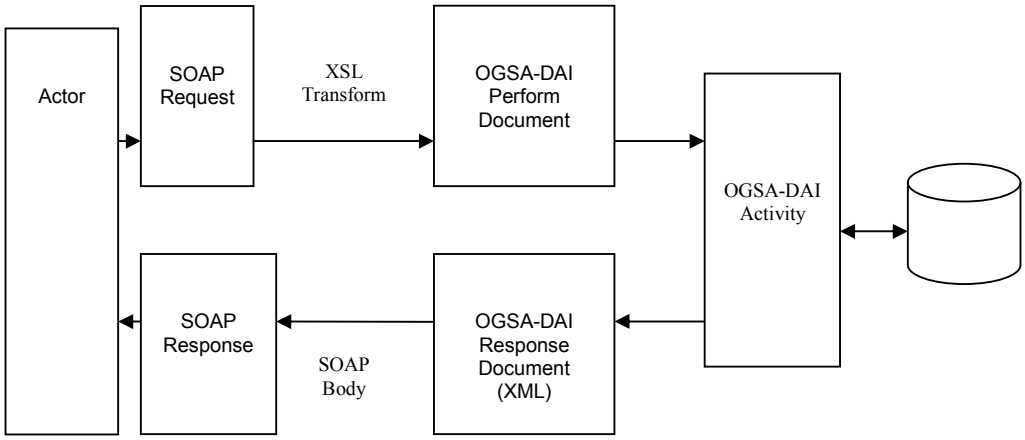
2.4.2 OGSA-DAI

The interface to OGSA-DAI is based on the submission of structured documents (known as *perform documents*) to OGSA-DAI through the web service interfaces it exposes. The *perform document* describes a set of activities that the OGSA-DAI enactment engine must perform. OGSA-DAI provides a large number of pre-written activities as well as giving the ability for developers to write their own.

For the Grid Provenance project, we have need only of the pre-written activities. We make use of the `xmlResourceManagement` activity for adding records into an XML database and the `xPathStatement` activity for querying the content of the database. The contents of a SOAP message received by the provenance store interface implementation are transformed using XSLT into an OGSA-DAI *perform document* that is submitted to the OGSA-DAI Web Service.

The flow of messages is illustrated in the following diagram. A record or query message arrives as a SOAP Request. This is transformed into an OGSA-DAI Perform Document using an XSL Transform. The Perform Document is input to an OGSA-DAI Activity which returns a response in the form of an XML document. This document is embedded in a SOAP envelope which is returned as a response to the requesting actor.

The PROVENANCE project receives research funding from the European Community's Sixth Framework Programme



The PROVENANCE project receives research funding from the European Community's Sixth Framework Programme

3 Provenance Functional Prototype Implementation

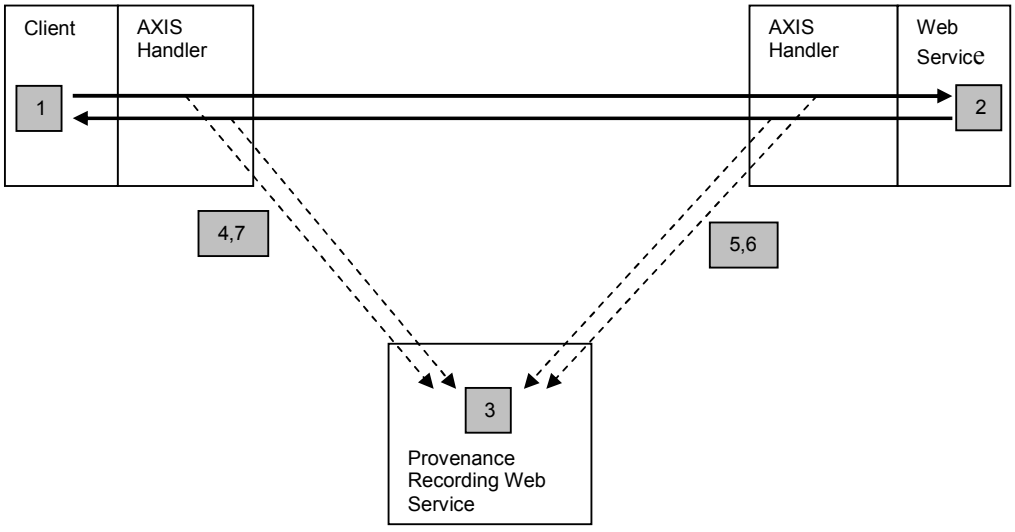
3.1 Package Structure

The functional Prototype is delivered as a single zip file, GridProvenance.zip and the entire contents of the installation package should be extracted using an unzipping tool such as WinZip. When unpacked, the directory structure contains all the installation documentation in the docs directory, the WSDL in the wsdl directory and the main body of the code in the file GridProvenance.jar in the extracted directory.

The instructions for installation of the Grid Provenance prototype can be found in doc/readme.txt. The documentation for the generated Java beans is presented as html JavaDocs. Before installing the functional prototype code, there are a number of pre-requisite pieces of software to install and configure, these include Java 1.4, OMII, OGSA-DAI and the eXist XML database. A full list of the pre-requisites and details of their configuration for Grid Provenance is within the readme. Users of the Grid Provenance package having to download the pre-requisites from their respective web sites.

3.2 AXIS handler

Within the functional prototype we have implemented an AXIS handler that allows provenance interaction p-assertions to be automatically written to the provenance store without the need to modify an existing actor. The handler must be inserted into the AXIS handler chain. It is designed to be used on either the client or service AXIS engines and the configuration of the handler allows the client and server to store records in separate Provenance Stores if required.



The handlers work as follows:

The PROVENANCE project receives research funding from the European Community's Sixth Framework Programme

1. The client [1] makes a request to the Web Service [2].
2. The client side AXIS handler intercepts the outgoing SOAP message, copies the data from the SOAP Body into an interaction p-assertion message and sends [4] it to the Provenance Web Service [3].
3. The service side AXIS handler intercepts the incoming SOAP message, copies the data from the SOAP Body into an interaction p-assertion message and sends [5] it to the Provenance Web Service [3].
4. The Web Service processes the request message and generates a response to be sent back to the client
5. The service side AXIS handler intercepts the outgoing SOAP message, copies the data from the SOAP Body into an interaction p-assertion message and sends [6] it to the Provenance Web Service [3].
6. The client side AXIS handler intercepts the incoming SOAP message, copies the data from the SOAP Body into an interaction p-assertion message and sends [7] it to the Provenance Web Service [3].

3.3 Query Interface

The PASOA project has developed a recording protocol to reliably record provenance assertions; however the query interface is less well defined. In order to use Grid Provenance we must have the ability to query the p-assertions that have been recorded. For this deliverable, we defined a simple WSDL definition for a Query Web Service which allowed an XPath query to be sent to the provenance store and a single result set consisting of any number of records to be returned..

The PROVENANCE project receives research funding from the European Community's Sixth Framework Programme

4 Conclusions and Further Work

This functional prototype is a first step on the road to a fully functional Provenance recording, querying and management infrastructure. It provides an implementation of the basic recording together with a query interface allowing recorded p-assertions to be retrieved. It does not contain any implementation of the security or scalability features. These will be added in further releases of the software.

The PROVENANCE project receives research funding from the European Community's Sixth Framework Programme