PROVENANCE
Enabling and Supporting Provenance in Grids for Complex Problems

Contract Number: 511085

An Architecture for Provenance Systems
Executive Summary

Authors:	Paul Groth Sheng Jiang Simon Miles Steve Munroe Victor Tan Sofia Tsasakou Luc Moreau
Reviewers:	All project partners
Identifier:	D3.1.1 (Final Architecture)
Type:	Deliverable
Version:	0.5
Version:	February 27, 2006
Status:	public

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

Executive Summary

Provenance Definition

According to the Oxford English Dictionary, provenance is defined as *(i) the fact of coming from some particular source or quarter; origin, derivation. (ii) the history or pedigree of a work of art, manuscript, rare book, etc.; concr., a record of the ultimate derivation and passage of an item through its various owners*.

Provenance is already well understood in the study of fine art where it refers to the trusted, documented history of some art object. Given that documented history, the object attains an authority that allows scholars to understand and appreciate its importance and context relative to other works. Art objects that do not have a trusted, proven history may be treated with some scepticism by those that study and view them. This same concept of provenance may also be applied to data and information generated within computer systems. This being so, one of our primary objectives is to define a representation of provenance that is suitable for computer systems, and the necessary architecture to make use of such a representation. Hence, in this context, we define the *provenance of a piece of data as the process that led to that piece of data*.

Computational Provenance

Generally, in computer systems, applications produce data. Our vision is to transform applications into so called provenance-aware applications, so that when they run, they produce a description of their execution. Such descriptions, which we refer to as *process documentation*, are stored in a *provenance store*, which is a repository for the storage and management of process documentation. Additionally, the provenance store also provides querying facilities to enable services to retrieve the provenance of data items. In support of this vision we have designed a *provenance architecture*, including suitable *data models* and the necessary underpinning *functionality*, with concerns for *scalability* and *security*.

The development of the architecture has been strongly influenced by the service-oriented architectural style, according to which services or *actors* interact with each other by exchanging messages. By enabling actors to make execution-related assertions, or *p-assertions*, we ensure that necessary and sufficient forms of process documentation are captured to be able to give a complete account of any data item's provenance. For example, the p-assertion model allows us to document various aspects of execution, and thus provide descriptions of those parts of an execution that relate to, or impact upon, a given data item. This allows a user to determine the data item's relationships to other data items and processes, such as its dependencies or causal effects and, at the same time, provides a description of the data flow through an application.

The p-assertions within a provenance store are organised in a conceptual structure, called the *p-structure*, based around *interaction records*, each of which is a collection of p-assertions that relate to a single interaction (i.e. an individual message exchange).

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

The p-structure provides a hierarchical view of process documentation that facilitates the retrieval of p-assertions, independently of the actual technology used in a given application.

Provenance Functionality

From a functional perspective, the provenance store supports two operations: *recording* p-assertions and *queries* over p-assertions.

In order to record p-assertions, the architecture offers a *recording interface* based on the p-assertion recording protocol (PReP). PReP is designed to be *stateless* to allow for asynchronous and out-of-order recording by actors. Furthermore, the provenance store's behaviour is specified to ensure that p-assertions do not become modified or deleted, preserving documentation in its original form, thus reflecting execution as it was originally documented.

Once recorded, documentation is then available for third parties to obtain the provenance of data items, which is achieved via a *process documentation query interface* for the retrieval of p-assertions and their contents, and a *provenance query interface* for the retrieval of a data item's provenance. Querying the provenance of a given data item involves: identification of the data item at a specific point during execution, and scoping of the process of interest to filter causal and functional relationships. The output of queries comes in the form of a collection of p-assertions representing a portion of the data flow graph, which allows a user to understand the provenance of the data item in question up to the specified point in execution.

Non-Functional Considerations

In terms of non-functional requirements, a provenance architecture must address three important considerations: *scalability, security* and *management*.

For many applications, extremely large amounts of process documentation can potentially be captured. This presents problems for recording, querying, management and storage of such information. Consequently, there is a need to deal explicitly with such scalability issues and, since the applications that record provenance may be distributed and large scale, the sheer quantity of recorded p-assertions requires a scalable means of storing them. To achieve this, the architecture enables several *recording patterns* that provide flexible ways for recording actors to record p-assertions. For example, one pattern allows different actors to record p-assertions in different stores, even if they refer to the same interaction. Because the documentation of a single process may end up being recorded in several provenance stores, in order to collect all the p-assertions about a process, it is necessary to provide directional *view links* to these provenance stores, where other parts of the documentation may be found.

For some applications, p-assertions may relate to large data sets, such as an actor's state, for example. In such cases, storage capacity problems can arise that are dealt with by allowing p-assertions to reference data that may be stored externally. The re-

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

placement of a data item with a reference can be seen as the result of a transformation, and constitutes just one of the possible ways that messages can be transformed using several *documentation styles*, which provide for more flexible ways to make assertions about data, and enable requirements on scalability and security to be met.

Security represents a central concern in many application domains, and it is standard software engineering methodology to integrate security features at the earliest time possible in the development life-cycle. Security concerns, both in relation to the interactions of the internal components of provenance systems and the actors using such systems are addressed, to ensure that appropriate access control for provenance stores is maintained. In addition, it is important that p-assertions can be attributed to the actor responsible for creating them, which is achieved by the inclusion of *assertion signatures*.

Management is not specific to provenance, but should contain functionality that is common to most data management systems, such as notification to users of changes to a provenance store (e.g. the addition or removal of p-assertions) and indexing of a provenance store's contents.

By developing an industrial strength provenance architecture, the EU Provenance project has made possible the capture and exploitation of provenance, and thus greatly facilitates the growth and utility of Grid-based applications by explicitly tackling the problems of trust, accountability, compliance and validation in such open, distributed systems.

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 Tudomanyos Akademia Szamitastechnikai es Automatizalasi Kutato Intezet	Hungary

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