Chapter 16

DiALog:
A Distributed Model for Capturing Provenance and Auditing Information

Christoph Ringelstein
University of Koblenz-Landau, Germany

Steffen Staab
University of Koblenz-Landau, Germany

ABSTRACT

Service-oriented systems facilitate business workflows to span multiple organizations (e.g., by means of Web services). As a side effect, data may be more easily transferred over organizational boundaries. Thus, privacy issues arise. At the same time, there are personal, business and legal requirements for protecting privacy and IPR and allowing customers to request information about how and by whom their data was handled. Managing these requirements constitutes an unsolved technical and organizational problem. The authors propose to solve the information request problem by attaching meta-knowledge about how data was handled to the data itself. The authors present their solution, in form of an architecture, a formalization and an implemented prototype for logging and collecting logs in service-oriented and cross-organizational systems.

INTRODUCTION

Service-oriented systems facilitate organizations to offer business capabilities as independent services. As standardized interfaces are used for the communication a loose coupling is supported. The loose coupling eases the integration of external services into internal workflows as well as the provisioning of services to consumers. The resulting flexibility facilitates the combination of services from different organizations into one comprehensive, integrated workflow leading to an agile virtual organization that is able to adapt more quickly to new organizational and business needs.
However, the new flexibility also shows disadvantages. An integrated workflow forwards and distributes data between organizations. The data may be confidential (e.g., personal data or intellectual property) and it may be involved in further processes (e.g., advertising). Thus, the distribution has the potential to violate concerns of privacy and IPR. Under such circumstances of flexible interworking between organizations, accounting for actions performed on data may be legally and/or contractually required.

To control the compliance with laws (e.g., Directive 95/46/EC in the EU), contracts, or policies, a data provider may request information about the processing and whereabouts of his data. The answer must contain details defined by the contract or law (e.g., who processed the data as well as why and how the data has been processed). The answer can be generated in different ways, e.g., by modeling and observing the distributed data processing. However, the answer can only be generated, if the model and the observation facilitate a detailed overview of the processing. Most frequently such an overview is lacking, even for internal workflows and data storage.

Hence, we require a model of the distributed processing of data in service-oriented systems in combination with a distributed mechanism for logging in service-oriented systems to collect the needed information and answer the request. Existing logging mechanisms, like the Extended Log File Format (Hallam-Baker et al., 1996) or syslog (Lonvick, 2001), are not sufficient to gain a full overview of a workflow that is distributed among multiple organizations, because they perform logging only in one execution environment. Because of the diversity of execution environments and because of a lack of standardized interfaces for exchanging logs, aggregating distributed logs remains a challenge.

In the following we present DIALOG (Distributed Auditing LOGs) and sticky logging. DIALOG is a method for auditing the distributed processing of data in service-oriented systems. Sticky logging monitors the processing of data items (independent of the actual business process) attaching the logs directly to the processed data as metadata. Furthermore, sticky logging allows for the reconstruction of how the data was processed by whom and why following the specification of DIALOG. Thus, sticky logging is a generic middleware for distributed logging. The paper is organized as follows: First, we present a scenario and analyze requirements for collecting information about the processing of private data in service-oriented systems. Following the requirements, we discuss various models for distributed processing of data. Then we introduce DIALOG and define notions of soundness and completeness relevant for the auditing in distributed systems. Based on DIALOG we present the architecture and a prototype implementation of sticky logging. Before we eventually discuss our approach and conclude, we compare it with related work.

**BUSINESS CASE**

We now present a business case that we use as a running example throughout the paper. The business case comprises a system that includes several services and is distributed over different organizations. To prove the functionality of the sticky logging mechanism this business case has been realized using the sticky logging prototype (see below).

**Small-Books Inc.**

The Small-Books Inc. is a book-selling company. Main parts of the logistics like storing the books, packaging, and shipping are outsourced to a logistics company named Fast-Shipping Inc. Analogously, the payments are processed by a credit card company. Assume that a customer, Mr. Smith, orders books via the Web site of Small-Books. To place his order, he has to insert his address and credit card number. Having received the order