XL: A Platform for Web Services

Daniela Florescu
XQRL, Inc.
dana@xqrl.com

Andreas Grünhagen
TU München
gruenhag@in.tum.de

Donald Kossmann
TU München, XQRL, Inc.
kossmann@in.tum.de

Steffen Rost
TU München
rost@in.tum.de

1. OVERVIEW

We present a platform for Web services. Web services are implemented in a special XML programming language[1] called XL [1, 2]. A Web service receives an XML message as input and returns an XML message as output. The platform supports a number of features that are particularly useful to implement Web services; e.g., logging, timetables, conversations, workflow management, automatic transactions, security. Our platform is going to be compliant with all W3C standards and emerging proposals. The programming language is very abstract and can be optimized automatically (like SQL). Furthermore, the platform allows to integrate Web services that are written in XL and other programming languages.

2. FEATURES OF XL

A detailed description of XL is given in [1]. Some of the features of our platform are:

Message-based programming: Web services communicate using messages; that is, services are invoked using messages and results are returned using messages. This way, services implemented on top of our platform can communicate with services not written in XL.

Web conversations: There are special constructs in order to implement business processes or conversations. In other words, it is possible to call other services as part of the same conversation and maintain the state of such a conversation. This feature is important, for instance, in order to implement sessions in which users login first and then make a sequence of other actions.

Web service composition: It is very easy to construct high-level services out of the composition of more basic services. Such basic services can be written in XL or in Java or some other programming language. The only requirement is that it understands SOAP messages.

High-level programming: Application logic is expressed in a declarative way. For this purpose, the platform provides special declarative constructs for common programming patterns; e.g., logging, timetables, event triggers, invariants, exception handling, parallel and pipelined execution.

Transactions and Security: Transactions and security is implemented along the lines of Java 2 Enterprise Edition [3].

3. DEMONSTRATION

We would like to present an auctioning system. This system consists of *bidders* which can be installed at any site in the Internet. Furthermore, there is a central *auctioning site* that keeps track of the bids and of the products that are to be auctioned. Bidders and the auctioning site are implemented on top of our platform. The bidders communicate with the auctioning site as part of a conversation; for each product that is auctioned, a separate conversation is started between the auctioning site and the bidders (i.e., those bidders who are interested in the product). Bidders and the auctioning site act according to certain rules. These rules are implemented in XL. The auction for a product ends after a certain timeout; i.e., if no bidder has made a new bid for a certain amount of time. (Of course, other auctioning styles are also conceivable.)

This example demonstrates how easy it is to implement such Web services using our platform and that programmers can concentrate on the application logic (i.e., the rules of the auction). Furthermore, this example shows how the platform implements its basic services in order to support such applications.

---

[1] By an XML programming language we mean a language whose only type system is the XML type system and *not* a programming language whose syntax is described using XML vocabulary.

4. REFERENCES

[1] D. Florescu, A. Grünhagen, and D. Kossmann. An XML Programming Language for Web Service Specification and Composition. Technical report, Technische Universität München, 2001. http://xl.in.tum.de/publ.

[2] D. Florescu and D. Kossmann. An XML Programming Language for Web Service Specification and Composition. *IEEE Data Engineering Bulletin*, June 2001.

[3] Java 2 Enterprise Edition. http://java.sun.com/j2ee/tutorial.