
Integrating Messaging and Databases to Implement Service Architectures

Michael Champion

Abstract

There has been much debate over two quite different approaches to implementing XML services. The "web services" approach leverages a rather large and not yet stabilized stack of formats and protocols built on top of SOAP that promise secure, reliable operations; the "REST" or "Plain old XML over HTTP" approach keeps the basic formats and operations quite simple, but puts the burden for any security or end-to-end reliability on the application developer rather than the computing infrastructure.

This presentation considers a third approach which complements many of the ideas in both WS and REST but uses an XML-capable DBMS as the messaging hub or service broker. This makes it feasible to support asynchronous, loosely coupled communications between service requesters and providers. Applications which offer or consume services can communicate with the DB with WS protocols, simple HTTP, or any number of proprietary mechanisms as needed. The key point is that the underlying DBMS implements the challenging security and reliability features that large scale applications require.

The recent addition of powerful XML features to modern DBMS environments allows the ready use of schema languages to define message format expectations, XQuery to locate messages of interest, XSLT to translate them, and so on. When XML messaging is integrated with XML database engines, users get simplicity: one programming model, one product to install and manage, and a minimal set of operations to apply to generic XML data. They also get enterprise class features such as high scalability, reliable transaction processing, and high-end security. By keeping things simple and using standard XML and database metaphors wherever possible, the difficulties inherent using asynchronous queues and pipelines for inter-application communication are minimized.

This architectural pattern is by no means a panacea, nor is it a competitor to either the web services or REST approaches, but offers a useful approach in a number of common situations. This presentation outlines an approach to implementing XML-powered service architectures that combines elements of each while maintaining basic compatibility with both. It uses the Service Broker feature of Microsoft SQL Server 2005 as a concrete example of one way in which this is being applied today.

Table of Contents

1. Paper Not Received 3

1. Paper Not Received

The author did not prepare a paper for the proceedings.

Biography

Michael **Champion**

Program Manager

[Microsoft Corporation](http://www.microsoft.com) [<http://www.microsoft.com>]

Woodinville

Washington

United States of America

Mike Champion has worked as a software developer since 1980, and have specialized in SGML/XML since 1996, working at Arbortext and Software AG. He is currently Program Manager for XML Standards at Microsoft. His blog is at blogs.msdn.com/mikechampion.