XAML

Transaction Authority Markup Language

White Paper

Executive Summary

Transaction Authority Markup Language (XAML) is a vendor-neutral standard that enables the coordination and processing of online transactions in the rapidly emerging world of XML web services – the revolutionary new model of Internet-based computing that is now being adopted by all major systems and software vendors. XAML is intended to be a completely open standard for web-based business transactions.

The standard defines a set of XML message formats and interaction models that web services can use in order to provide business-level transactions that span multiple parties across the Internet.

XAML is not owned by any one vendor. Instead, the standards proposal will be submitted to an appropriate open standards body to ensure that it remains an open industry standard in which all interested companies and organizations can participate.

XAML is intended to enable a whole new class of dynamic, highly distributed computer interactions known as "business web transaction processing" or BWTP, as distinguished from OLTP, or classical online transaction processing. Business web transactions involve web services from multiple organizations on the web and must coordinate the low-level operations of commit, cancel, retry, and compensate (undo or reverse), in order to ensure business-level transaction integrity.

The following scenario demonstrates a business-level transaction involving a set of web services that would utilize XAML. Consider an industrial company that purchases benzene from a chemical manufacturer on the web. In order for the buyer to purchase the benzene, she requires additional value-added services provided by third parties, such as shipping with specific delivery terms, payment financing, casualty insurance, and government compliance for safe transport. The buyer will not agree to the purchase of benzene until all of these services are available, and all meet her requirements. She will purchase all of them or none of them. In other words, all of these inter-related requirements need to be satisfied in order for the business transaction to be completed.

This scenario requires that the industrial company initiate a set of calls to web services that are owned by the various product and service providers mentioned above. Today, web services provide the low-level means of supporting basic requests to distributed systems, using protocols such as HTTP, XML, SOAP, and other industry-specific data formats. However, in this scenario, the industrial company needs to be able to engage these web services to form a single business transaction.

In order to do this, the requestor must be able to coordinate the calling of the individual web services, and must have a means of asking for commitment from all of them, prior to actually committing. The process involves multiple interactions with each of the web services, and multiple stages of progress towards each web service's transactional completion, with the ability to commit and cancel operations, and, in some cases, initiate compensating web services that undo or reverse the work of previous transactions.

The software providing the top-level business transaction needs to coordinate with each of the participating web services. These include: (1) the chemical provider's inventory system; (2) an insurance policy service to insure the product being shipped; (3) a financing service to ensure payment according to vendor terms; (4) a transportation service to guarantee timely shipment/delivery of product; and (5) a regulatory service to ensure compliance with government safety requirements.

Each of the underlying web services provides one part of the overall business transaction context. Should one of these services refuse or fail to perform its operation, XAML provides the means by which the industrial company can issue requests to a combination of the same or different web services asking them to cancel or provide compensating operations to reverse the benzene order. This level of multi-staged coordination and processing of web services is required for complex B2B commerce operations.

People have been implementing distributed transaction management within the firewall for years. Using XAML, companies will be able to push the envelope beyond the firewall – they will be able to build advanced business-level web service interactions that take advantage of distributed web services, and provide essential management for business transactions spanning multiple parties.

The ability to create transaction-enabled, dynamic business webs on the fly changes the world of business forever.

* * *

Introduction

Bowstreet, Hewlett-Packard Company, IBM, Oracle, and Sun Microsystems are leading an initiative to enable distributed e-business transactions across the Internet. The companies are developing an open, vendor-neutral standard that will enable the coordination and processing of online, multi-party or multi-vendor transactions in the rapidly emerging world of web services. A web service is any software process that is exposed on the web and that uses XML or related message formats.

The new standards initiative is called XAML, for Transaction Authority Markup Language. The XAML initiative is so named because it is an application of XML (the common language of e-commerce) that defines transactional interaction among web services, based on interfaces as defined by the widely adopted XA (Transaction Authority) standard and by JTA (Java Transaction API).

XAML will provide the standard mechanism to enable XML web services to participate in business transactions spanning multiple parties across the Web. Web services provide unprecedented business interoperability by enabling businesses to share processes and competencies on the web, creating a new era of business connectivity and dynamic, “plug-and-play” e-commerce.

As plug-and-play e-commerce emerges, businesses are mixing and matching web services from multiple partners to create sophisticated e-business applications. Because these “business webs” consist of loosely coupled web services distributed across the web from multiple businesses, coordination among these web services is imperative, in order to carry out business-level transactions. There needs to be the notion of a transaction at the web service level, as well as a means by which software systems can coordinate the processing of calls to multiple web services to provide higher-level business transactions.

XAML supports transactional semantics at the web service level, in addition to enabling businesses to coordinate the calling and processing of interdependent web services among multiple partners operating in dynamic e-commerce environments. This vendor-neutral standard will precipitate a tsunami of e-commerce activity.

XAML defines a set of XML message formats and interaction models for web services that enables individual web services to provide transaction processing capabilities, and collections of web services to participate collectively as business transactions. The goal is to enable transaction processing, as we know it behind the firewall, to take on a whole new meaning in the distributed world of web services on the Internet.

Bowstreet, Hewlett-Packard Company, IBM, Oracle, and Sun Microsystems plan to submit the XAML proposal to an appropriate open standards body to ensure that it remains an open industry standard.

Business Webs

Today, almost any company has the potential to use the web to improve the way it does business. During the next few years, millions of companies of all sizes will take advantage of the web, by automating their routine business processes and making them available as web services.

Businesses are finding new ways to work with web-based partners, including the use of trading hubs, whose major purpose is automating the process of buying and selling. XML-based web services standards have been a major enabler in this evolution. For example, UDDI focuses on enabling web service providers to supply information used in the discovery of web services and the integration of web service partners. WSDL focuses on enabling descriptions of web services, while protocols such as SOAP define the transport of data and the interoperability of web services.

The leading companies are now pursuing a new model – they are building business webs. Instead of just automating their internal business processes and moving them to the web, companies are constructing sites that integrate calls to both their own and their partners' web services. These "business web" companies want to leverage their brands and existing channels, not by automating captive supply chains but by aggressively collaborating with partners and distributors. This collaboration is called "business web automation."

Business web automation is fostering a new kind of competition and creating new sources of wealth. In the future, most companies will become aggregations of constantly evolving services offered under a brand. Business consultant and best-selling author Don Tapscott, co-author of *Digital Capital: Harnessing the Power of Business Webs*, says, "The b-web is emerging as the generic, universal platform for creating value and wealth…. If the corporation embodied capital in the industrial age, then the b-web does the same for the digital economy."

Plug-and-Play e-Commerce

Prior to the web and the introduction of web services, when companies wanted to partner with other companies, they normally had to agree in advance on which tools and technologies to use to interface their business processes. This process of agreement usually involved the coordination and sharing of the companies' programming staffs. The process was almost always lengthy, tedious, and costly. Even today's digital trading hubs must drag their integration customers through this process.

But now, the business web economy enables companies to rapidly snap together calls to distributed partner companies' web services, to form business webs. Web services have self-describing XML interfaces, and can be called using standard web protocols. IT groups can now focus on defining the desired *business-level* interactions, instead of dwelling on low-level programming APIs.

The next logical step is "plug-and-play e-commerce" – the execution of these business-level interactions across business webs.

The Challenge: Web Service Transactions

Plug-and-play e-commerce requires a standard for the coordination of interactions within an environment of business webs. In this context, a business-level transaction can require the operations of one or more web services. The required standard has not existed until now.

With the XAML standard, companies can create a business web that offers desirable, real-world business transactions.

For example, consider a purchasing system that interacts with a manufacturer's web site to place an order for a quantity of a product. As the purchasing system places the order, the manufacturer's business transaction software negotiates terms for product shipment and insurance coverage of the product with third-party web service providers. For the purchaser, this high-level business transaction is a great convenience because the manufacturer is able to provide the coordination of the shipping and insurance terms based on the specifications of the product. In addition, the manufacturer can provide a high-level business transaction that ensures that the insurance and shipping web services are "booked" only in the event that the product is actually purchased.

This business transaction requires coordination of the web services of three companies. In order for the purchaser to get the convenience of this single point of interaction, the three participating companies must coordinate their services to behave as a single business transaction. If one of the web services cannot be performed, compensating actions must be taken to ensure the integrity of the purchaser's complete business transaction. The mechanism for coordinating and processing each of the web services to achieve this end is the missing element in plug-and-play e-commerce.

The Solution: XAML

XAML is a software-independent transaction standard that provides the missing element. XAML enables web services to participate in business-level transactions, while protecting current investments in back-office systems.

Bowstreet, Hewlett-Packard Company, IBM, Oracle, and Sun Microsystems have stated the following objectives for XAML:

- Provide a specification for the XML message interfaces and interaction models of web services to support the coordination and processing of multi-stage transactions on the Internet.
- Specify interfaces and protocols that preserve investment and strengths in transaction monitors and resources.
- Specify interfaces and protocols that can be "added on" to existing and emerging web service interfaces and protocols, including SOAP.
- Specify interaction models for software systems that provide business-level transactions that coordinate the processing of multiple distributed web services.
- Build on existing and emerging industry standards.

Transaction Management before the Web

Before we discuss how XAML works, let's take a brief look at the history of transaction management in the years *before* the web.

Transaction management is the process of making simultaneous changes in several places "atomically" – that is, all the changes related to a transaction are made or none of the changes are made.

For example, within a single database connection, the DBMS provides some means of demarcating the beginning and end of a transaction. This demarcation ensures that changes to the database are made atomically.

Sometimes, changes must be made atomically across *multiple* databases. For example, an insurance company might have to change both its claims information and its audit information at the same time, even though the audit information is in a separate database from the claims information. This multiple-database change would ensure that, during a later audit, the company would know which agent took the first report of the loss.

In this case, the existing XA (Transaction Authority) protocol is useful. XA provides a standard mechanism for coordinating changes to multiple databases (called resource managers or RMs) as an atomic unit of work. Basically, the XA protocol asks each RM to vote on whether a commit will be successful. Once an RM has voted “yes,” it must be able to commit the open unit of work without failure. The commit occurs only if all RMs vote “yes.” This process of obtaining a vote, and then performing a commit, is called a “two-phase commit.”

Resource managers are most frequently databases, but they can also be message-oriented middleware. XA allows completely heterogeneous collections of RMs within a single transaction; for example, a transaction can commit across DB2 and Oracle at the same time. All major database vendors support XA.

Relationship of XAML and Web Services to Existing Transaction Processing Systems

XAML will enable web services to expose transactional semantics of the resources providing the services. Given that TP monitors commonly provide some of the management and coordination functions of these resources 'behind the firewall" today, one of the goals of XAML is to enable TP monitors to participate and support the transactional semantics offered by web services. This includes passing of transaction ID's through web service messages, and supporting the XAML web service operations of commit and cancel.

At the level above individual web services, there is a new layer of software providing business-level transactions. This software makes calls to multiple web services, often spanning business boundaries. Given that XAML enables individual web services to support transactional semantics, there is also an opportunity for XAML to specify standard means for coordinating business-level transactions across collections of web services. To this end, one of the goals of XAML is to define message interfaces and interaction models that help software systems providing the business-level transactions to coordinate the interactions among web services.

There is an opportunity to define XML interfaces and interaction models for a new breed of web services that would help software systems at the business transaction level. These services would provide brokering capabilities for managing the interactions among web services, for both web services supporting XAML, as well as web services that do not support XAML. This new breed of web services requires XML interfaces and interaction models that define how software systems at the business transaction level would interact, to request assistance in shepherding a set of web services towards completion.

How XAML Works

XAML is an XML markup language used in web services, along with a set of interaction models, that establishes a means by which web services can perform transactional units of work, and participate in business-level transactions that span multiple web services. XAML enables a web service to provide transactional capabilities, including commit, cancel, and compensatory actions. XAML also enables software systems to coordinate the calling of multiple web services, so as to ensure that all of the work, or none of the work, is completed.

Consider a transaction that has to span two distributed web services at the sites One.com and Two.com. Through the use of XAML-encoded messages, the calling system issues a web service request to One.com. One.com replies, indicating that it can perform the service. Next, the calling system obtains the same level of promise from the web service at Two.com. At this point, the calling system can request activity from both One.com and Two.com. When the activity is complete, the calling system can obtain status on the activity from each web service, and can request to have that work committed, canceled, or compensated for, at each site. Depending on the success or failure of the work provided by each web service, the software handling the business transaction could coordinate the desired overall outcome.

XAML extends transactional capabilities to web services, enabling businesses to offer fine-grained interaction with their web services. This is critical to the software that provides business transactions and must ensure integrity of the high-level process.

Summary

XAML is a set of XML interfaces that allows web service providers to participate in business transactions on the web. These business transactions coordinate the interactions among distributed web services. In addition, XAML provides a standard means for individual web services to offer transactional capabilities.

There are emerging standards for web services (such as SOAP, ebXML, XP, UDDI, e-Speak and WSDL), and there are existing transaction management standards (such as XA and JTA). XAML ties the two groups of standards together.

XAML is an XML-based language that will be a common ground for transaction systems and web services. It will help companies meet the need to interact with the tsunami of web services that will redefine the computing landscape.

As companies build business webs, and produce remarkable new business models for buying and selling based on web services, XAML will enable these companies to integrate and leverage their existing transaction systems, and participate in business transactions that were never possible before.

For More Information

For more information on XAML, please visit www.xaml.org.