BizTalk Server 2000 Business Process Orchestration

Bimal Mehta, Marc Levy, Greg Meredith Tony Andrews, Brian Beckman, Johannes Klein, Amit Mital
Microsoft Corporation
One Microsoft Way, Redmond, WA 98052

Abstract

BizTalk Server 2000 unites, in a single product, enterprise application integration (EAI), business-to-business integration, and the advanced BizTalk Orchestration technology to allow developers, IT professionals, and business analysts to easily build dynamic business processes that span applications, platforms, and businesses over the Internet. BizTalk Server 2000 is a high-performance platform for integrating message-based applications in the Internet providing support for transport normalization, message transformation and business process management. BizTalk Orchestration, BizTalk's business process management component, is deeply integrated with Microsoft SQL Server and COM+ for addressing the needs of automating mission critical business processes. Its unique compensation and exception handling features provide additional support to address difficult real life business problems. BizTalk Orchestration introduces a novel exception handling and compensation framework. The article will explore the different areas addressed by the BizTalk product, in particular its support for open-nested transactions and its unique compensation and exception framework for handling errors in long-lived business transactions.

1 Introduction

BizTalk Server 2000 unites, in a single product, enterprise application integration (EAI), business-to-business integration, and the advanced BizTalk Orchestration technology to allow developers, IT professionals, and business analysts to easily build dynamic business processes that span applications, platforms, and businesses over the Internet.

Business Process Management. The BizTalk Server infrastructure helps integrate, manage, and automate dynamic business processes by exchanging business documents among applications, and within or across organizational boundaries. With all the tools that companies need for business process orchestration, BizTalk Server helps building processes that span not only applications, but also businesses, over the Internet. Graphical tools make it easy for analysts and developers to model and implement these processes.

Integrate Applications and Business Partners. BizTalk Server 2000 makes it easy to integrate applications and businesses together with a host of rich graphical tools for building Extensible Markup Language (XML) schema, performing schema transformation, establishing trading partner relationships over the Internet, and tracking and analyzing data and documents that are exchanged. With XML and standard Internet transport and

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

security technologies support, BizTalk Server 2000 extends the features of traditional e-commerce and electronic data interchange (EDI) to entire e-commerce communities.

Interoperability Using Public Standards. With extensive support for public standards and specifications, such as XML, EDI, Hypertext Transfer Protocol (HTTP), and Simple Mail Transfer Protocol (SMTP), and security standards like public key encryption, digital signatures, and encryption, BizTalk Server 2000 ensures the highest level of interoperability and security for applications and business partners.

2 BizTalk Server Architecture

BizTalk server is built for performance, reliability and scalability. BizTalk server is targeted to process hundreds of business documents per second, even on small sized configurations. Microsoft SQL Server is used to store all business documents. COM+ application servers host and execute long-lived business processes. Clustering and fail-over support provided by Windows2000 and SQL Server ensure a very high level of reliability and availability [3].

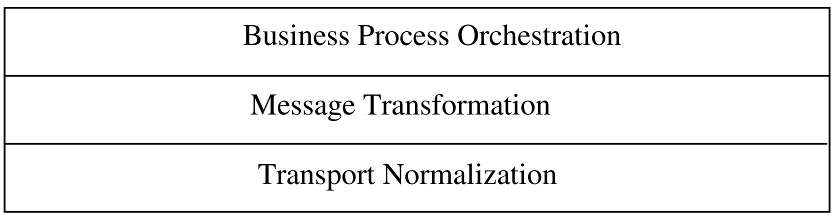

Figure 1: BizTalk Processing Layers.

BizTalk server manages trading relationships between business partners. We can distinguish three different layers within BizTalk Server: transport normalization, message transformation and business process management. At the messaging layer trading relationships are described by a set of transport protocols over which to exchange business documents and a set of rules for encrypting, parsing and validating documents exchanged. Once a document has been received and passed all processing steps it is forwarded by BizTalk Server to a subscribing application, business process or remote trading partner. Transmitting a document may require the document to be transformed into one understood by the receiving partner, signed, encrypted and sent over a transport with the correct set of delivery guarantees.

BizTalk Server supports the protocols and message formats defined in BizTalk Framework 2.0 [3] to reliably exchange business documents over the Internet.

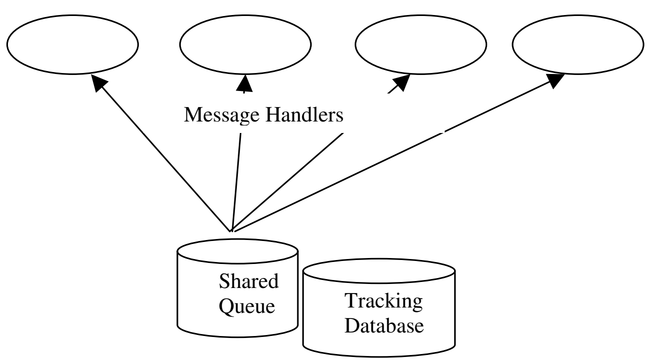

Figure 2: BizTalk Server Group.

BizTalk Server allows the configuration of groups targeted towards a particular application. Each group has its own set of resources. Each group has a database acting as shared queue. A second database is used for recording information on the progress of messages.

Message handlers access the shared queue for storing and retrieving messages. For scale-out message handlers are spread across nodes in a cluster. Processing nodes can be added or removed depending on the load of the overall system.

All groups share a common configuration database. Different groups may be used for different types of work. An application may use one group for low-latency message processing jobs while routing bulk processing jobs to a different group to ensure quality of service.

All message processing is done within a transaction. Messages received are stored in the shared queue before forwarding them to a trading partner, application or business process. Business may use the message store for auditing and optionally mining the exchange of documents between business partners.

BizTalk Orchestration layered on BizTalk messaging has been built to handle transactional, enterprise-scale business processes. BizTalk Orchestration processes individual steps in a business process within a transaction. It stores, i.e. "dehydrates", the state of a business process in the database. BizTalk Orchestration makes use of COM+ transaction services to ensure consistency and recoverability of the overall business pro cess.

3 BizTalk Orchestration Programming Model

BizTalk Orchestration is BizTalk Server's business process management component. BizTalk Orchestration uses the underlying transformation and messaging capabilities of BizTalk Server for exchanging documents with business partners and applications. BizTalk Orchestration supports the execution of long-lived business processes.

BizTalk Orchestration's programming model is based on the notion of exchanging business documents, i.e. messages between autonomous agents. All actions are expressed in terms of messages received and messages sent. Messages are constructed by specifying how to populate fields with data from already existing messages. Messages once created are immutable documents.

The message centric approach adopted by BizTalk Orchestration [1] ensures that business processes are compositional. It reflects the true nature of business processes in the Internet which do not allow for shared state but have to depend on all information be communicated. Individual participants publish their parts of a business contract. It describes what documents to exchange and the order in which they have to be submitted.

Figure 3 shows an interaction diagram, which describes interactions, i.e. the exchange of business documents, in a procurement process. BizTalk Orchestration defines an XML based language, XLANG, which captures those interactions in a document. XLANG programs capture the dynamic and static properties of messages exchanges between business partners. Dynamic properties define sequencing rules, guarding the exchange of messages. Static properties define ports, i.e. transport endpoints, and how they are connecting participants cooperating in a business process.

BizTalk supports all the usual control flow constructs such as sequence, branching and loops. For connecting the various participants BizTalk ports can be bound to a variety of transport protocols such as BizTalk messaging, COM, and MSMQ.

BizTalk is unique in its support for passing binding information at runtime. This is a very common technique used by Web services, which pass URIs as referrals, or redirects to partners. It allows business processes to dynamically connect and locate participants in a business process upon receipt of an e-mail address or upon receipt of an HTTP URL. Ariba's cXML protocol [5] is a good example for the importance of such a technique when connecting buyers and suppliers in a market place. Reference passing is key to dynamically establishing multi-party relationships.

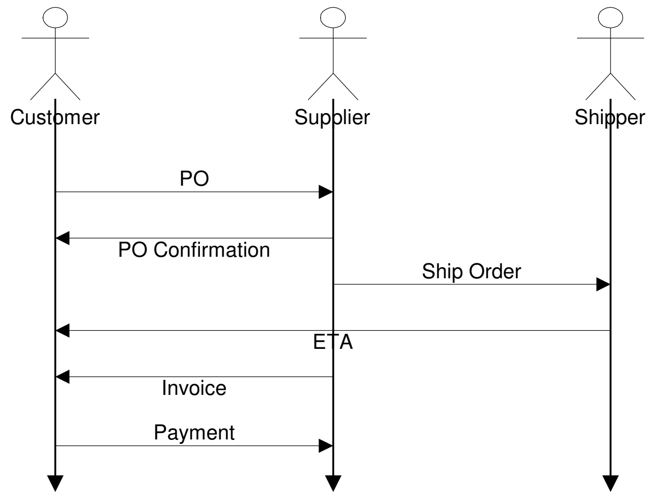

Figure 3: Procurement Scenario.

In the Internet service descriptions and binding information will be located using directory services as proposed by Microsoft, IBM and Ariba [4] and service descriptions as described in WSDL, a joint Microsoft and IBM proposal for publishing web services [4].

4 Transactions, Exceptions, and Compensations

BizTalk facilitates management of trading relationships in the Internet. In the future we expect businesses to register with directory services and publish service descriptions for customers to automatically locate services and start trading. Today, setting up and debugging a trading relationship is a very time consuming and costly task. Error handling and recovery are often manual adding additional cost to the execution of business processes and making it difficult to rely on automated processes. Support for long-lived transactions, exception handling and compensations is key to truly enabling business process automation in the Internet.

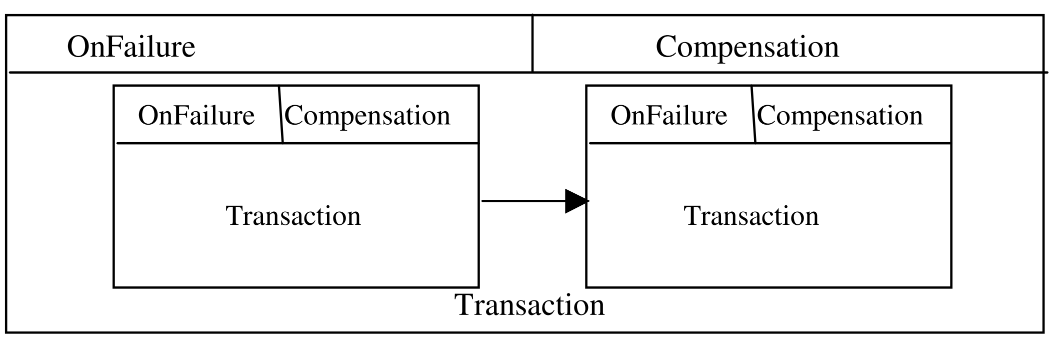

Figure 4: BizTalk Open Nested Transactions.

BizTalk Orchestration allows business process designers to define transaction boundaries. Transactions may be nested. Innermost transactions are atomic database transactions [2]. The other types of transactions include a variant of open nested transactions. BizTalks open nested transactions implement long-lived business processes. The state of a long-lived business process is recoverable up to the last committed ACID transaction.

In contrast to undo operations of ACID transactions compensations for long-lived open nested transactions are supplied by the application. BizTalk Orchestration allows applications to associate a complete compensating business process with a transaction. The process defines how to compensate for the actions of the original transaction. BizTalk invokes compensation processes on abort of the parent transaction. Compensations are invoked in reverse order. Compensations are executed within a new transaction context.

BizTalk Orchestration associates exception-handling processes with transactions. Exception handlers get invoked on abort. Exception handlers are executed within a new transaction context.

The BizTalk compensation and exception-handling framework controls the visibility of messages. Message visibility is defined as follows. A transaction has access to all messages associated with its parent transaction. Once a transaction commits its messages are inherited by the parent and visible to all siblings.

For an aborted transaction only the messages created by its exception handling process are visible to the parent transaction. Thus an exception handling process can control what results, i.e. messages become visible upon transaction abort.

While undo operations of ACID transactions are executed in isolation business level compensations often depend on the outcome of subsequent transactions and preceding compensations. Examples are purchases, which could not be returned, or money spent, which could not be recovered. The message visibility rules presented here enable compensations to get access to this information by examining messages generated by previously executed transactions and compensation.

5 Concluding Remarks

BizTalk Server is a powerful platform for defining trading relationships and integrating messages based application in the Internet. Built on a mature database system and a scalable high performance transaction-processing platform it is targeted towards the execution of mission critical business processes. With its unique support for compensation and exception handling it gives applications additional tools to automate costly error scenarios. BizTalk's commitment to a message based framework sets it apart from other business process frameworks and positions it as one of the best for composing services in the Internet for executing business contracts.

References

[1] Milner, R. Communicating and Mobile Systems: The pi-Calculus Cambridge University Press. Cambridge, UK (1999).

[2] Gray, J. and Reuter, A. Transaction Processing: Concepts and Techniques. Morgan Kaufman Publishers, San Mateo, CA (1993).

[3] http://www.microsoft.com/biztalk

[4] http://www.uddi.org

[5] http://www.cxml.org