The n-Tier Hub Technology

Rainer A. Sommer, Thomas R. Gulledge, and David Bailey

Enterprise Engineering Laboratory

George Mason University, Mail Stop 2E4

Fairfax, VA 22030-4444 USA

rsommer@gmu.edu, gulledge@gmu.edu, dbailey3@gmu.edu

Abstract

During 2001, the Enterprise Engineering Laboratory at George Mason University was contracted by the Boeing Company to develop an eHub capability for aerospace suppliers in Taiwan. In a laboratory environment, the core technology was designed, developed, and tested, and now a large first-tier aerospace supplier in Taiwan is commercializing the technology. The project objective was to provide layered network and application services for transporting XML-based business transaction flows across multi-tier, heterogeneous data processing environments. This paper documents the business scenario, the eHub application, and the network transport mechanisms that were used to build the n-tier hub. Contrary to most eHubs, this solution takes the point of view of suppliers, pushing data in accordance with supplier requirements; hence, enhancing the probability of supplier adoption. The unique contribution of this project is the development of an eHub that meets the needs of Small and Medium Enterprises (SMEs) and first-tier suppliers.

Key Words: eHubs, Extensible Markup Language (XML), Service Provider, Small and Medium Enterprises (SMEs).

1 Introduction

The project was sponsored under the Industrial Automation and eBusiness Development plan for the Republic of China (ROC) [http://www.iaeb.gov.tw]. The project objective was to enable and accelerate Taiwan's transition into a hub for managing an extended enterprise that spans the world. In accordance with internal eBusiness objectives, the ROC government sponsored the Boeing Company to build an eBusiness Laboratory for technology transfer). Boeing assembled a Team[1] to help with the design and implementation of the laboratory, and to develop and test an eHub solution with a large first-tier supplier and three ROC mid-range ERP solution providers. The details of the technology solution are presented in this paper[2]. General information is located at http://boeingicp.eep.gmu.edu/.

The laboratory was designed to facilitate the integration and testing of service provider business applications in the lower tiers of large supply chains. Since many of Taiwan's manufacturers are SMEs that are suppliers to large international verticals, their interaction with trading hubs has become an absolute requirement. In general, large-scale supply chains are very well integrated with their 1st tier suppliers. However, lower-tier suppliers serve many vertical industries and usually do not generate enough transaction volume in each vertical tier to justify the costs associated with multi-trading hub application integration. This situation is depicted in Figure 1.

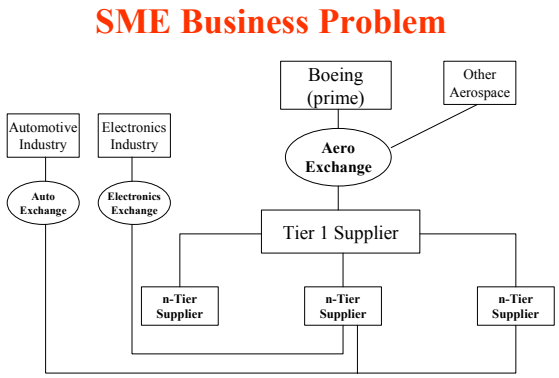

Figure 1: n-Tier Supplier Business Relationships

Our solution to this common problem is to develop lower-tier business models and hub technologies that are specifically tailored to the needs

[1] The eBusiness Cooperation Project Team is comprised of the Boeing Company (USA), Corporate Synergy Development Corporation (ROC), Devco, Inc. (USA), National Tsing Hua University (ROC), and George Mason University (USA).

[2] George Mason University was the technical lead on this project.

of SMEs, providing lower-tier SMEs access to cost-effective trading exchange and data processing technologies. A second way to address the integration problem that faces n-tier suppliers is to develop a cost-effective n-tier hub software solution that allows lower-tier suppliers to aggregate their B2B resources. In essence, such a solution empowers groups of SMEs to combine their resources and provide B2B services to higher-tier exchanges[3]. To transact business in this manner, SMEs only need to interact with the hub using their standardized XML business document convention, as indicated in Figure 2. In contrast to Figure 1, this model makes sense because n-tier suppliers usually provide goods and services to many vertical industries and thus a common hub model allows direct B2B interactions among many exchange partners without the complexity and expense of implementing multiple interface technologies.

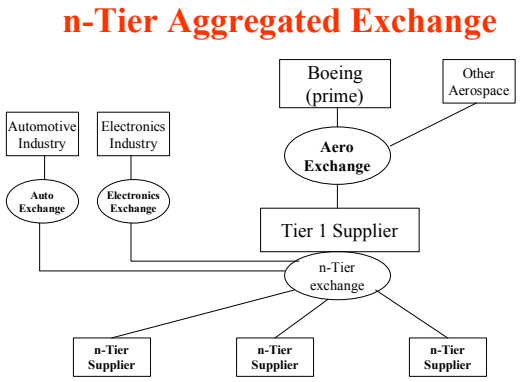

Figure 2: n-Tier Aggregated Exchange

An n-tier exchange, as depicted in Figure 2, was developed and tested in the Boeing demonstration laboratory at National Tsing Hua University (NTHU) in Hsinchu, Taiwan. The hub was developed to help the domestic service provider community, who are driven by market forces to design, test, and deploy SME-oriented B2B applications in an n-tier supply chain environment.

The technology of the n-tier hub is explained with a simple XML-based transaction model for routine procurement transactions. The choreography for this model, as in the example presented in Figure 3, minimizes service provider requirements for participating in an n-tier hub.

[3] In fact, it usually makes sense for this aggregation to occur at a large first-tier supplier, since these companies have incentives to broker transactions.

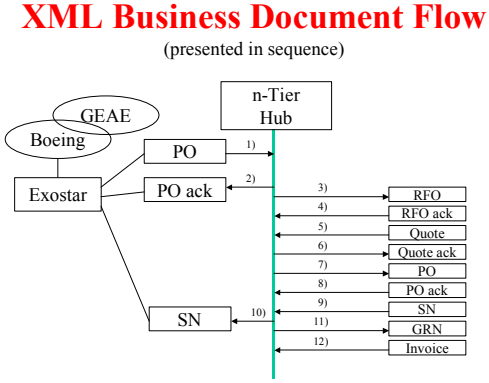

Figure 3: XML-Based Choreography

The figure depicts a sequenced transaction flow that utilizes the n-tier exchange application to translate an Exostar[4] XML standard into a local service provider supported XML standard. In this scenario the difficult B2B XML translations are handled by the respective exchanges (Exostar and the n-tier hub), while the SME service provider application supports a single native XML standard for transacting business with a large number of customers. Hence, suppliers can participate using the data exchange format of their choice, as opposed to meeting the mandate of large prime contractors[5]. For this application, we used various flavors of XML, but the hub technology developed in this project supports true any-to-any document swap.

2 The n-Tier Application Architecture

In order to operationalize the n-tier-enabled B2B transaction process, the hub application must be simple to deploy and update. Hence, the laboratory team used commercial "off-the-shelf" technologies to build a componentized exchange application based on open standards. An extensive review of existing technologies revealed that the Oracle 8i database environment, along with mature Oracle development tools and web service applications, provided the basic functionality for designing an n-tier hub.

The most critical technology requirement of an exchange is the design of an XML translation engine that provides the necessary parsing, query, and formatting functions to transform one XML standard

[4] Exostar is the aerospace and defense trading exchange. See http://www.exostar.com/.

[5] The supplier participation problem is well known in the literature. The issues surrounding the problem are discussed in detail in [1].

document into another. In general, the database is used as a library of Document Type Definition (DTD) elements. Each XML standard[6] implements the basic business document naming convention in a relatively standardized format (i.e., document names "Purchase Order" and "Request for Quote" are just two of the standardized naming conventions implemented by the various XML business document development initiatives). However, the specific data elements (tags) that define the content of the XML business document can vary to a great degree[7]. These differences can be especially varied among diverse vertical industries [2]. For example a "Purchase Order" document that was developed by the automotive industry will contain data elements (tags) that are superfluous, irrelevant, or incompatible with the "Purchase Order" requirements used by the food and grocery industry.

XML standardization is also a problem within the same industry verticals, especially if the supply chain business partners use ERP systems to support B2B integration. Most all ERP and transaction processing systems implement a native form of XML. These forms are mostly based on "open" or proprietary standards and thus may not fit with the XML requirements of their customer and supplier systems. The problem of supporting multiple flavors is well known in the trade literature [2, 3, 4].

The problem for lower-tier suppliers is even more critical. In most cases supplier systems must accommodate several XML standards. Since each prime customer may mandate a different XML standard, it becomes increasingly difficult to support a growing customer base in an integrated B2B model [5]. This form of integration is costly because of the transaction processing complexity required to support multiple XML standards and the associated customer-specific application business rules. Instead of using complex EAI solutions that are often too costly for suppliers, we provide the business documents in accordance with supplier requirements.

To support such a complex business model in an ERP-based B2B environment, we build a database that holds all XML data elements and describes the most common business documents exchanged with all customers and suppliers. In essence the database holds every variation of every XML-based business document that is used to transact business with every customer/supplier. This hub-based "any-to-any" translation defines our XML translation engine.

To build the translation engine the laboratory team used the Oracle 8i database XML parsing engine, Oracle Workflow, Oracle XPATH search utilities and the PL/SQL transaction processing language to parse and identify every data element of an incoming XML document [6]. Once the document is parsed, the XPATH utilities populate a master table in the database with all the identified tags.

Next, a transaction logic routine written in PL/SQL searches for the ID tag of the intended receiver and calls an XSLT (XML Style Sheet Translation) table. This transformation is accomplished via PL/SQL logic, defining the syntax and semantics of the transformation. XSL is a language for transforming XML documents into other XML documents, and XSLT is designed for use as part of XSL as well as providing a style sheet language for XML. In addition to XSLT, XSL includes an XML vocabulary for specifying formatting. XSL specifies the styling of an XML document by using XSLT to describe how the document is transformed into another XML document that uses the formatting vocabulary [6].

Figure 4 depicts an n-tier hub transaction scenario that is modeled with Oracle Workflow. The transaction sequence is initiated through a Process Pilot graphical user interface mechanism designed by the Promatis Corporation [http://www.promatis.com]. Process Pilot initiates a PL/SQL procedure that loads the initial attribute values for the Oracle Workflow, along with the name and location of the local file(s) to be transformed. These values are then entered into the ICP_FILENAME_T transformation table. Oracle Workflow then reads the contents of a message queue to determine if an XML-based business document has arrived. If a document exists in the queue, its contents and that of the initial ICP_FILENAME_T table are stored in the ICP_XMLDOCS_T table as CLOBs[8].

[6] XML standards are numerous. Currently there are over 200 "flavors" of XML, some tailored to a particular vertical industry, some are proprietary, while others are general "open" standards sponsored by industry or governmental groups.

[7] The EDI community defines this inconsistency as "Implementation Convention" differences.

[8] A CLOB is an Oracle Large Object whose value is comprised of character data that corresponds to the database character set defined for the Oracle8 database

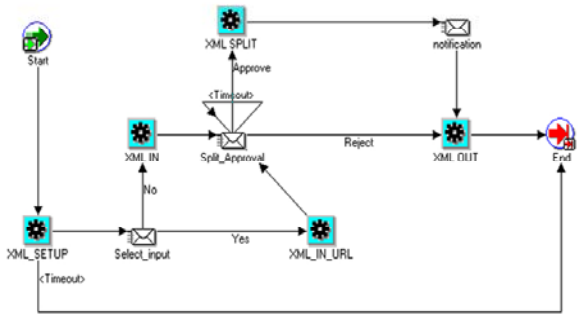

Figure 4: n-Tier Trading Exchange Application Architecture

Next, the workflow engine will ask the administrator for approval to initiate an XPATH search of the stored document. If approval is given, the workflow proceeds to the XML_SPLIT process. If approval is not given, the workflow proceeds to the XSLT Transform procedure. Following this workflow-based table manipulation, a PL/SQL procedure is initiated. This procedure uses XPATH to separate the table-based XML tag set(s) into individual entities and adds the information to the ICP_HEADLINES_T table. Searches and PL/SQL procedures can now be executed against individual data items stored in the database. In this manner, the data are transformed via style sheets (XSLT) into other XML documents. Once the XML document has been transformed to the new "flavor" of XML, the workflow system writes its output to the local file system where it can then be picked up by the appropriate transport mechanism.

Currently this n-tier hub application manipulates database tables through PL/SQL initiated procedures. In its current form it does not support complex business logic to preprocess XML documents in preparation for company specific ERP-based business document exchanges. All XML document related business logic must be executed by the respective customer/supplier ERP systems.

3 The Uniqueness of the Contribution

The uniqueness of this contribution is not in the technology of the solution. Most EAI vendors (e.g., IBM, Oracle, Iona, CrossWorlds, Enterworks, etc.) can support any-to-any document swap. However, these solutions require a client-side implementation at the supplier location. For example, Oracle Message Broker works perfect if the customer and the supplier have OMB implemented at both locations. Unfortunately, most suppliers have many customers, so they are forced to accommodate many client-side solutions, and this is not practical.

The uniqueness of our solution is that no special requirements are required for a supplier-side implementation. Since we "push" the business documents in any format that the supplier requires, there is no need to implement a proprietary client. This may seem trivial from a technology point of view, but it was the key to the success of the implementation.

4 The n-Tier Transport Mechanism

Although the n-tier hub application is capable of manipulating the database XML document translation, the transport mechanism requirement used to receive and transmit XML data from customers and suppliers is complex. Large businesses possess the necessary capital and expertise to develop very elegant and complex messaging systems that handle the XML document transport quite easily. Unfortunately, these resources are usually not present in SMEs. SMEs are often characterized by older legacy information technology and under-trained personnel. To that end the n-tier hub application must be layered on top of a cost-effective and easy to configure network-messaging standard.

4.1 Oracle 8i Message Broker

To provide a true universal network transport mechanism, a robust and sophisticated messaging architecture was required. Since the laboratory team was already familiar with the Oracle 8i database, the advanced features of the product were incorporated into the n-tier application architecture. The 8i architecture includes advanced queuing features found in the Oracle Integration Server add-on. Advanced Queuing (AQ) is the database-integrated message queuing component of the Oracle8*i* Enterprise Edition database management system. It provides an infrastructure that simplifies the passing messages within and among applications.

Its functional strong points include:

- Store and forward capability
- Simple message management
- Recovery to point-of-failure
- Transactional integrity
- Guaranteed message delivery between databases
- Message mining
- Message tracking
- A choice of programming interfaces (APIs) to place messages in and remove them from queues

The AQ functionality allows the n-tier hub to communicate with other message brokers (IBM MQ Series, Microsoft Message Broker, etc.) that reside on other systems. This situation is depicted in Figure 6. Depending on a SME's business needs (i.e. simple or complex business system integration needs), the message broker architecture can be used to customize XML document transmission and receiving options to meet a wide variety of message processing options. These options include very basic "hands-on" browser-based XML processing and can range to fully integrated "hands-off" workflow enabled ERP system processing.

Given the flexibility of message broker technologies, the benefits are numerous, but the technical complexity and associated maintenance costs can be quite high and thus out of reach for most SME implementations. To date, integrated message broker technologies are mostly utilized in more sophisticated data processing environments where high volume transaction flows can justify the purchase and maintenance costs of the software.

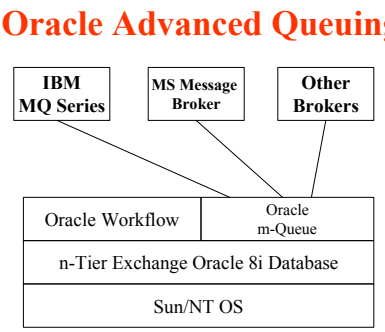

Figure 5: Oracle Advanced Queuing

Although message broker technology can be scaled to fit most any data processing model, the technological sophistication required to configure and maintain the system is cost prohibitive to traditional SMEs.

4.2 Oracle Internet File System (iFS)

To make the n-tier exchange cost-effective for a wide range of SME needs, a centralized messaging system was required. The advanced queuing message broker paradigm required the SME to implement and maintain complex software technologies on their local systems. A significant reduction in cost and complexity is achieved if the n-tier trading exchange application supports the messaging and transport software at the hub level. This model allows an SME to link to a variety of messaging options (from simple to complex) without having to install, configure, and maintain complex localized client software. A solution to this problem was developed using the Oracle Internet File System (iFS) technology.

The Oracle iFS is a departure from localized and distributed file storage mechanisms. iFS only relies on local/distributed file systems (i.e., NTFS, FAT, UFS, NFS, CIFS, etc.) for managing database files and providing database integrity. Under iFS, all document and data files are stored within the database; hence, file access, security, and manipulation are performed within a relational database environment rather that a local/distributed file system. In this environment files can be accessed via a traditional Windows interface (a.k.a. Windows share), over the web, via e-mail, ftp, and through traditional client software.

Given that files are stored in a database file system rather than a traditional file system, the power and security of relational database processing can be focused on a single file, or any number of files. These features include file manipulation and transport, which make iFS appropriate for transferring business documents to SMEs. By integrating the iFS with the Oracle 8i database and advanced queuing capabilities, the n-tier exchange eliminates the need for SMEs to implement client software. SMEs are only required to have an Internet-enabled computer that can access the n-tier hub. The iFS provides a single shared file system, advanced queuing, and messaging to all entities (customers, suppliers). Under this model, customers and suppliers are assigned individual database shares on the n-tier hub application, with these shares offering bi-directional file access in a variety of ways:

- Windows based drive mapping, where the drive actually represents an iFS database share on the hub, through which XML documents are up- and down-loaded.
- e-Mail enabled access, which allows an SMTP e-mail client to access the iFS database share to send/receive XML documents.
- File Transfer Protocol, which allows an FTP client to access the iFS database share in order to send/receive XML documents.
- Web Folders, which allow an HTTP client (Explorer or Netscape) to access the iFS database share in order to send/receive XML documents.
- Web Folders with WebDAV (Web-based Distributed Authoring and Versioning), an Oracle-specific extension to HTTP that allows

for remote cooperative editing and file management in an iFS database share.

The details of an n-tier hub using the iFS technology are depicted in Figure 6.

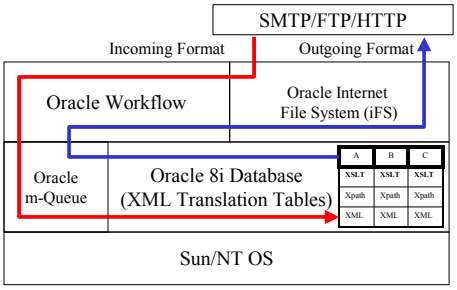

Figure 6: n-Tier Exchange iFS Details

The technologies depicted in Figure 6 provide a powerful array of transport options that are easy to implement and maintain from an SME perspective. That is, if the hub is maintained at a first-tier supplier, lower-tier suppliers only have to access a file over the Internet.

Although the n-tier iFS concept is meant to bridge the SME cost/technology complexity gap, traditional message broker (m-Queue) technologies still remain a critical component of the n-tier hub application. The message broker is required in the n-tier architecture because it offers more complex transaction processing capabilities to higher-tier customers and suppliers that have integrated ERP and business system transport requirements built around the broker communication paradigm. However, the participating SME will most certainly find a cost-effective alternative in the localized iFS transport services offered by the n-tier hub application.

5 Conclusions

The n-tier exchange is in an on-going development mode. Work is being completed on implementing a large library of XSLT-based style sheets for performing XML translations among many differing standards and to implement the complete array of iFS transport options. Additionally, work is underway to implement basic business logic within the n-tier application itself. The addition of Oracle Workflow based business logic allows SMEs to preprocess XML-based business documents in the exchange application without having to download the document. Many SME's don't have ERP or modern accounting systems that can execute routine business process logic (i.e., routing a purchase order to another location, qualifying a customer's credit worthiness, etc.), thus these common business processes may add benefit to the n-tier community as a whole. This situation is depicted in Figure 7.

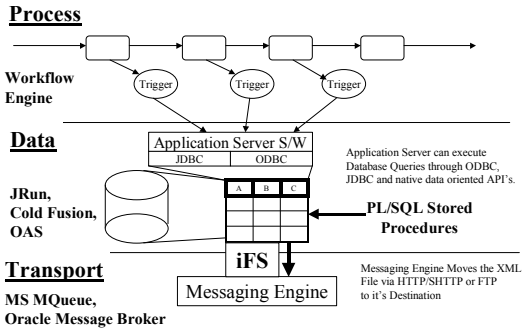

Figure 7: Completing the n-Tier Model

Although the hub provides SMEs with needed functionality, we are aggressively trying to extend the functionality to better meet SME needs

6 References

[1] Gulledge, Thomas R., eMarketplaces and Small- and Medium-Sized Enterprises, Accepted for Publication in *Computers in Industry*, 2001.

[2] Gibbons-Paul, Lauren, XML Standards: Too Much of a Good Thing? PC Week, April 12, 1999. http://www.zdnet.com/devhead/stories/articles/0,4413,398134,00.html

[3] Messmer, Ellen and John Fontana, XML Struggling to Gain Customer Acceptance, *Network World*, September 11, 2000.

[4] Liebman, Lenny, XML's Tower of Babel, *InternetWeek*, April 24, 2001.

[5] Cox, John, Survey shows XML use growing fast in Enterprises, Network World, 02/26/01. http://www.nwfusion.com/news/2001/0226xml.html

[6] Muench, Steven., Building Oracle XML Applications, O'Reilly & Associates, 2000.