IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4134

Review of Middleware Components in Multi-tier Structures

Qiyang Chen, Vinai Sharma and John Wang
Information and Decision Sciences Department, School of Business, Montclair State University, Upper Montclair, NJ
Tel: (973) 655-7270, chenq@mail.montclair.edu

ABSTRACT

Embracing inapt infrastructure technology is a major threat in developing extensive and efficient Web-based systems. The architectural strength of all business models demands an effective integration of various technological components. And middleware, the center of all applications, becomes the driver—everything works if middleware does. In the recent times, the client/server environment has experienced sweeping transformation and led to the notion of the "Object Web". Web browser is viewed as a universal client that is capable of shifting flawlessly and effortlessly between various applications over the Net. This paper attempts to investigate middleware and the facilitating technologies and point toward the latest developments, taking into account the functional potential of the on-market middleware solutions as well as their technical strengths and weaknesses. The paper would describe various types of middleware including database middleware, Remote Procedure Call (RPC), application server middleware, message-oriented middleware (MOM), Object Request Broker (ORB), transaction-processing monitors, and Web middleware etc., with on-market technologies.

INTRODUCTION

Evolution of Internet-based computing from local area networks (LANs), after transitioning from unconnected computers to networks, is the hallmark of all business models today. The technological backbone of this evolution is the middleware. First connecting, then communicating, and finally seamlessly integrating the distributed systems to external sites, customers, suppliers, and trading partners across the world is the real challenge for the business world. It doesn't stop there. Also required is the talking between client and server over heterogeneous networks, systems architectures, databases, and other operating environments. All this is facilitated by the middleware technologies that offer undercover functions to seamlessly integrate various applications with information instantly to make it accessible across diverse architectures, protocols, and networks. Automation of back-end and front-end operations of business is also effected by the middleware. Middleware binds discrete applications, such as Web-based applications and older mainframe-based systems, to allow companies to hook up with latest systems and developments that drive new applications without making their investments in legacy systems unyielding.

The chances of huge returns expected due to enabling middleware technology are, however, controlled—and often diminished—by the fact that the consequence of unpredictability or improper configuration of the middleware technology is extremely severe. Web browser war has given way to the middleware war. Numerous vendors offer various middleware product families—"the operating system of the Web"—with an estimated growth of about 65% for Object Request Broker, 50% for Messaging, and 15% for Transaction Processing Monitor for the year 2001. [Radding of Standish Group, 1998]

FUNCTIONS OF MIDDLEWARE

Middleware functions are generally classified into:

- Application-specific functions to deliver services for different classes of applications such as distributed-database services, distributed-data/object-transaction processing, and specialized services for mobile computing and multimedia,
- Information-exchange functions to manage the flow of information across a network—for tasks like transferring data, issuing commands, receiving responses, checking status, and resolving standoffs, and
- Management and support functions to locate resources, communicate with servers, handle security and failures, and monitor performance.

Database, Web, and legacy application middleware are three basic middleware application types, with database middleware as the major application in most systems and Web middleware is spreading fast and extensively. SQL mainly provides communications between clients and servers in database middleware. Vender-dependent standard such as ODBC is developed to keep apart the client application from database server implementations and can be easily reached from most application programming environments. Non-relational data now can be accessed through Universal Data Access. Remote procedure calls, messaging middleware, transaction process monitors, and object-oriented middleware are four basic middleware communication categories.

Since middleware is a relatively immature technology, the competing standards and fast changing technologies factor into the selection process making it a very complex decision-making area that requires a meticulous insight and analysis of the correlation and interaction between the middleware and applications. The two major issues that concern and influence this process include:

- The types of applications to be carried out by the middleware technology based on the current application environment such as distributed relational database project or non-relational file servers dealing with VSAM and ISAM data and
- The application development environment that is based on if a client application will be written in a traditional procedural language such as COBOL, an event-driven language like Visual BASIC, or an object-oriented environment such as C++ or Java.

MAJOR TYPES OF MIDDLEWARE

The selection of middleware technology is determined by what information is required to be communicated, for example, database middleware will be the choice if database is the main requirement. However, following are the major categories of middleware:

- Database Middleware,
- Remote Procedure Calls (RPC),
- Object Request Broker (ORB),
- Application Server Middleware,
- Message Oriented Middleware (MOM),
- Transaction Processing Monitor (TP),
- Object Transaction Monitor, and
- Web Application Servers

The most widely used, easy to install, and relatively economical middleware, *Database middleware*, is usually chosen to complement other types of middleware and facilitates communication among applications and local or remote databases but cannot transfer calls or objects. However, database middleware does not allow the two-way communications between servers and clients. SQL type command is generally subjected to the middleware gateway, which would convey the command to the end database to collect and send the reply of the

This paper appears in *Issues and Trends of Information Technology Management in Contemporary Organizations,* the proceedings of the Information Resources Management Association International Conference.

SQL query back. Synchronous point-to-point type of communications is the characteristic of database middleware and can pose problems when multiple demands from multiple users produce huge traffic and congestion. Database middleware is the most mature middleware technology.

Remote Procedure Calls (RPC) permits a client program to call procedures located on a remote server program. Remote procedure calls is not isolated as distinct middleware level and is entrenched into the application with calls embedded into the client portion of the client/server application program. Stubs are developed for both the client and the server to call up synchronously when the client makes a call to the server. The intricacies of distributed processing are reduced by remote procedure calls by maintaining the semantics of a remote call no matter the client and server are located on the same system or not. The synchronous nature of the remote procedure calls makes it most appropriate for smaller applications where all communications are one-to-one and not asynchronous.

Object Request Brokers (ORB) are language-independent, object-oriented, synchronous remote procedure calls in which an affiliate function of an object can be brought into play remotely by means of the same essential notation. Asynchronous communication suitable to large applications can be made possible by extending the main standards as in CORBA and DCOM, the main competing standards. Java's Remote Method Invocation (RMI) is also like an ORB, but is not language independent. Interfaces between objects can be defined by an IDL (interface definition language). The codes can be recompiled to avoid troubles associated with changing theses interfaces dynamically. ORB technologies are based on the reliability of the transport layer, which is required for the functioning. The application programmer is secured from the details of the client/server approach by using IDL interfaces that allows the application code to call a remote object, as if it were locally supported. Thus, the maintainability is improved as the object communication details are concealed from the application and isolated on the ORB. Hence, ORB-based middleware applications are becoming standard for the multi-tier model. CORBA, enterprise Java Beans, and Active X/COM are among the major technologies. The main limitation of ORB technology is that it goes well only with CORBA, JAVA, or COM oriented applications in the network. However, it can be used with TP Monitor, which supports the client/server communication management; CORBA with TP Monitor tender a good communication middleware in running Web-based business transactions.

Message Oriented Middleware (MOM) or enterprise message technology (EMT), provides asynchronous message delivery. The messages are lined up, just as objects, permitting the application that sends messages, to carry out other tasks without getting blocked till it receives the response. Generally located at a higher level than that of remote procedure calls, MOM assembly provides more than simply passing information. MOM also offers provisions for translating data, security, broadcasting data to multiple program, error recovery, and prioritization of messages and requests. MOM enhances flexibility by allowing applications to switch messages without the requirement of knowing on which platform or processor the other application located. MOM facilitates communications across a range of messaging systems, such as request-response, prolonged conversation, application queues, publishing and subscribing, and broadcasting. Messages are processed asynchronously with appropriate priority levels. Examples of messaging middleware are IBM's MQSeries, BEA's Message Q, and Microsoft's MSMQ.

Transaction Processing Monitor (TP) is over 25 years old technology that controls interactions between a requesting client and databases. It is a database independent technology. TP provides a three-tier client/server model and ensures an appropriate updating of the databases. This technology facilitates and controls the transport of data between numerous terminals and the application programs serving them. It can provide services to thousands of clients in distributed client/server environment by multiplexing client transaction requests by type on to a controlled number of processing routines that support particular services. TP monitor technology can also capture the application transitions logic from the clients. TP monitors are not employed for general-purpose program-to-program communications, but offer an environment for transaction type applications that utilize a database. On initiation of transactions by clients, the TP monitor transports the transaction relating to the database as required and the response is sent back to the client. By the use of TP monitors, the developers can define segments of the application as transactions with a clear start and stop points. TP monitors also offer failure protection and ensure atomic transactions by turning around a transaction back if not completed successfully. TP monitors are often used synchronously to ensure that failures are detected before further transactions take place. This is accomplished by breaking down complex applications into pieces of code called services. Three types of jobs are performed through TP monitors: process management, transaction management, and client/server communication management. Examples of TP monitors include IBM's ClCS and BEA's Tuxedo. BEA's Tuxedo is used by some companies as the middle-tier to allow an application developer to write applications as a set of services, each executing a single business function, such as add customer, process orders, and bill customers. Tuxedo supports LAN and Internet-based clients through its Tuxedo IWS and BEA Jolt. C, Cobol, and Java can do programming in Tuxedo applications. Standard Tuxedo communications integrate the business logic proficiently with legacy applications. Tuxedo also supports many security levels, such as the application password, end-user authentication, access control, and mandatory access control.

Object Transaction Monitor (OTM) is a coherent form of middleware consolidated and integrated from discrete technologies available such as TP Monitors, Message Q, and ORB. For instance, the Object Transaction Monitor (OTM) combines the functionality of MOM, TPM and ORB into one product (Boucher and Katz, 1999). BEA's Tuxedo integrates with ORB in one single OTM product known as M3. In addition, Microsoft's MTS, Iona's OrbixOTM, and Inprise's Visibroker ITS also offer OTMs. MTS, COM, and MSMQ are integrated in COM+ and IBM integrates these technologies into the family of WebSphere products. The latest trends in the middleware solutions support the integration of various technologies to increase returns by reducing costs of employing experts for different technologies and applications in the design and functioning of e-business models.

MIDDLEWARE AND THE WEB APPLICATION SERVERS

With the domination of the Internet over all business models, middleware technology has grown remarkably in the recent years and is expected to grow further. For instance, MOM that is the most potential candidate in middleware technologies is expected to grow at a compound annual growth rate of about 20%. The mission-critical obligation of systems engages far-reaching use of distributed computing to share information across expanding heterogeneous networks. That requires significant diffusion of information consistently across numerous applications on a wide range of machines—embedded in the range from the reproduction of a database to extraordinary push technologies to push required technology over the Internet. By and large, the information traverses throughout the network in the form of messages and the infrastructure entails software development.

HTML pages are served up through the application servers' interaction with back-end databases and applications. However, application servers differ by operation speed, support for component models, programming standards, and database interfaces. Application server software permits originating applications as a set of software components, such as Enterprise JavaBeans or ActiveX controls, and loading these applications on the application server. The components use the features, or the services, of the application server, such as accessing back-end databases, controlling transactions to those databases, inter-

acting with front-end Web server, and even balancing the workload over several copies of the application server. (Cox, 2000)

Sophisticated and easy-to-use Web development tools are being developed by the Web application server vendors. Simplification of coding and integration of distinct tasks allows the next-generation application servers to be developed with much reduced development loads for a successful Web operation, particularly in terms of time. However, one must be careful about increased dependence on the vendor's technology due to several factors: proprietary application components, sizable investments in software and product-specific training, and extensive use of the vendor's own development and consulting services. (Liebmann, 2000)

High vendor-specificity of the application server architecture, infrastructure, and connectivity, as well as increasing business logics being stored in a vendor's application server, lead to increased dependence of a company on vendor's product line. There are two ways to deal with the problem: one way is to create and design applications in a manner that each application function is separated into its own discrete components, so that if any particular feature of the application server becomes outdated, it can be substituted with another technology without breaking down the whole system. The second solution to the problem can be aimed at by integrating standard programming environments such as C++ and Java.

With pervading Web-technology over all business models, the use of application servers as critical components of the infrastructure has become more fundamental than even operating systems and databases. A fast-paced growth of the Net also requires the Web-based applications to be much more scalable, reliable, and transaction supporting. The standard application deployment platform of these applications with the Java 2 Enterprise Edition (J2EE) supports their use as application integrators with XML capabilities. BEA, IBM, and iPlanet/Sun/Netscape/AOL are among the leading vendors of the Web-application servers on-market. These Web-application servers are based on Java and Enterprise Java Beans (EJB) component model and J2EE standards. Each claims leadership in some or the other dimension. BEA, for example, with its WebLogic Server, is the unit volume leader. IBM leads in revenue, a claim supported by its associated implementations of AS/400 and S/390 in traditional enterprises, whereas iPlanet is the leader in the high-growth/high-value Internet sector. (D.H. Brown Associates, Inc., 2001)

BEA's WebLogic Server offers an easy-to-deploy J2EE platform that marks the mainstream market. WebLogic Enterprise builds on this capability, but requires two programming environments to accomplish all of its functions. iPlanet is consistent in almost all areas. IBM's strength in high-end transaction services, application-development tool integration, as well as its leadership in XML and UDDI make a powerful product. IBM's product also requires two programming environments to accomplish all of its capability, bridging CORBA and legacy application systems. BEA WebLogic Enterprise offers strong J2EE conformity, web-presentation services, object services, security, reliability, and scalability. WebLogic is also as good as IBM's offer in management functions and as good as iPlanet Application Server 6.0 in Java Messaging Services (JMS). WebLogic's execution of J2EE is considered cleaner or easier to install. On the other hand, iPlanet and IBM offer integrated implementations of XML. However, WebLogic 6 integrates XML into the package.

The Sun-Netscape-AOL alliance's iPlanet Application Server offers strong capability in J2EE. The first web-application server to be certified J2EE compliant. iPlanet 6.0 has a unique strength in simplified application development at the enterprise level based on its Unified Integration Framework that offers single programming environment to build enterprise applications. By contrast, BEA WebLogic requires the user to dub a separate server, called E-link. IBM's WebSphere EE offers both the Advanced Edition container and CORBA-based Component Broker. However, iPlanet Application Server 6.0 linger in management and transaction functions. IBM's WebSphere EE 3.5 leads the pack in transaction services and development-tool integration. It also is a leader in driving and propagating the XML and UDDI application-integration standards. WebSphere EE includes WebSphere AE as well as IBM's Component Broker technology. However, WebSphere lags in J2EE certification (Moorehead, 2000).

The factors that influence and drive this high-growth market segment characterized by furious competition come from the ever-expanding demand for application integration, growth of services, scalability and reliability, ease of implementation, and security control.

MAJOR PRODUCT FAMILIES OF MIDDLEWARE TECHNOLOGY IN WEB APPLICATION SERVERS

BEA's WebLogic integrates EJB with other Java-standard services required for business-critical applications and provides database integration, event management, remote object access, and directory service integration. Additional technologies sustaining the latest version include WML, XML and SNMP, with the add-on software kit offering support for the latest draft of the EJB 2.0 specifications. WebLogic provides see-through replication for automatic load balancing and fail over, strong RSA-based security; rich, graphical management of Enterprise JavaBeans, comprehensive support for the Java Enterprise Standards, and full Java types as arguments to Enterprise JavaBeans.

The WebLogic EJB Deployer tool provides the controls for managing multiple EJB.jar files and for configuring WebLogic Server deployment properties and resources. Deployer tool supports two levels of EJB deployment validation by automatically checking properties and references to make sure they contain the appropriate values and by verifying that key EJB required classes are compliant with the EJB 1.1 specification. WebLogic Zero Administration Client (ZAC) Publish Wizard, a graphical utility, allows creating, publishing, and managing packages containing an application, applet, or library of Java code. ZAC permits building the client-side Java applications and package them for distribution.

IBM's WebSphere can be used for e-business applications in multitier client/server environments. WebSphere software platform consists of three layers: WebSphere Foundation—consisting of Application Server and MQ series—and integrates the business processes and applications and transfer them to the Web; Foundation Extension to help the business develop, present, and deploy Web applications and extend the back-end systems to the web; and Application Accelerators that helps the business develop customized solution for collaboration and B2B integration. WebSphere supports every phase of e-business development from the start to the end. The WebSphere provides standards-based Java environment, XML, and XSL support, and support for applications built in Java Servlet specifications, transaction monitor, and message queuing software. It also provides site analysis tools and native language support. (Ling, Rihao, David , and Chou , 2001)

WebSphere Application server includes a HTTP server, user interface logic, the data to construct user interface of the e-business application, the business logic, and the connectors (Conner, 1999). The connectors support the communications with external applications, data, and services. The user interface logic is separated from the business logic and follows the user interface style and mechanisms. The client browser communicates with WebSphere Application server through HTTP, IIOP, and MQ Series. The Application Server integrates e-business applications with data server, existing enterprise resources, and business partners. WebSphere is capable of providing connectivity to many different types of databases and offers high-speed database access using JDBC.

WebSphere Application family includes many products that maintain the existing enterprise systems like the enterprise resource planning (ERP) and aid the integration with business partners. It also makes possible the access of the existing enterprise applications available to the customers, employees, and suppliers through Intranets and Extranets by making these applications securely available to the Web.

WebSphere Host Integration Solution makes possible the integration of legacy applications and critical enterprise data with the Web seamlessly. The Transcoding Publisher and WebSphere B2B Integrator enhance B2B data transaction. IBM's MQ Series can be used for flexible and rapid integration of core business process, data and applications, and B2B transactions. IBM also presents MQ Series Integrator that integrates applications originally developed with different systems for supply chain management and customer relationship management. WebSphere's VisualAge for Java offers patterns for expanding ERP systems (SAP R/3) to e-business (Rosencrance, 2000). IBM offers products ranging from simple Web applications to the assorted Transcoding Publisher, a technology that automatically translates and transforms Web-content (HTML) to XML or WML, the markup languages that are currently used in creating content and applications for pervasive devices such as wireless application protocol (WAP) devices, personal digital assistants (PDAs), and so on.

Sun-Netscape's iPlanet Application Server is the result of efforts to remake Netscape's Application Server, combined with NetDynamics technology, to take its place as a foundational piece of the iPlanet Internet Service Deployment Platform. iPlanet is expected to provide an extra level of support. iAS 6.0 meets the requirements of the latest J2EE specification and passed J2EE Certification Test Suite. Basic core services of iAS include a transaction monitor, multiple load-balancing options, full clustering and failover support, an integrated XML parser and Extensible Stylesheet Language Transformations engine, and full internationalization support. iPlanet products including Directory Server, Web Server, and other add-ons for Enterprise Application Integration are well integrated. The iPlanet Application Deployment tool is a Java-based program that leads through the process of deploying an application. Several well-documented sample applications present an overview of creating and organizing real-world applications. iAS offers a separate product, iPlanet Application Builder, that integrates with such third-party tools as WebGain Studio, Inspire JBuilder, IBM Visual Age, and Sun Forte for Java Enterprise Edition. iPlanet offers a good e-commerce platform with applications and performance required to successfully deploy today's cutting-edge Web applications. (Astley, Sturman, Daniel and Agha , 2001)

iPlanet provides the enterprises with a scalable, high performance platform on which to develop and deploy their large-scale, transactional Web applications. iPlanet provides advanced availability features such as failure recovery and session management and scalability features including dynamic load balancing. Developers can tool their applications in Java or C++, in addition to the offered support to the Client Independent Programming Model (CIPM) for the rapid development of applications for differing client types. iPlanet accomplishes high performance through its fully multi-threaded, multi-process architecture and advanced connection caching and pooling. Scalability is endowed with through dynamic load balancing and point-and-click application partitioning, allowing applications to scale animatedly to support numerous users. New enhancements include a standards based application model supporting Enterprise JavaBeans, JSP, JDBC, and the Java Servlet API. In addition, distributed transaction support enables high performance and centralized management, while "fail over" functionality provides enhanced application availability.

The Table 1 shows a comparison among several products of Web application server.

CONCLUSION

The traditional two-tier client server system can no more manage very large amounts of data and transactions due to a wide range of expanding and diverse operating systems, applications, and environments. What is required now is a translating middle layer that can make these systems, applications, and environments communicate seamlessly at a time. There are many middleware technologies developed to streamline these efforts. An imminent solution is that businesses adopt a multi-tier client/server architecture and infrastructure supported by several middleware technologies.

The choice of technology will be based on the specific needs of the business situation. It has been established now that middleware solutions offered by IBM's WebSphere, BEA's WebLogic, and Sun's iPlanet lent a hand to the businesses by integrating legacy systems to various applications on diverse platforms. These middleware solutions proved to be proficient and cost-effective to construct and support complete e-business models. No organization will be able to compete and survive today's fast-changing, Net-ized, e-business environment characterized by huge data flow unless it adapts and adopts a multi-tier client/server architecture. The middleware solutions, such as WebSphere, WebLogic, or iPlanet not only integrates the Web, client/server, and legacy systems but also supports integration of various middleware technologies to secure investment in existing technologies and skills and incorporate with ERP and business partners.

REFERENCES

Astley, Mark, Sturman, Daniel C., Agha, Gul A., (2001) "Customizable Middleware For Modular Distributed Software—Simplifying the Development and Maintenance of Complex Distributed Software," Communications of The ACM, May 2001, Vol.44, No. 5.

Boucher, Karen and Katz, Fima, (1999) Essential Guide to Object Monitors, John Wiley & Sons, Inc. New York.

Conner, Michael H., (1999) "Structuring e-business applications," IBM, Library Papers. www.IBM.Com/Software/Developer/Library

Cox, John (1999) "Web app servers bulk up on new features," www.nwfusion.com

CSIRO (2000), "Helping to Choose the Right Middleware," http://www.cmis.csiro.au/adsat/publications.htm

Edwards, Jeri (1999), 3-Tier Client Server At Work, John Wiley & Sons, Inc. NY.

Liebmann, Lenny (2000), "Web Application Servers-Are You Ready For The Commitment?" Internet Week online, 06/28/2000.

Rosencrance, L. (2000) "Middleware," Computerworld, Vol. 34 No. 41; p. 84.

Ling, Raymond Rihao, Yen, David C., and Chou, David C., (2001) "From Database to Web Browser: The Solutions to Data Access," J. of Computer Information Systems, Winter 2000-2001.

Moorehead, Kathy (2000), "Database Middleware Technologies," 9/4/2000, http://scis.acast.nova.edu

Rawles, Phillip T., and Goldman, James E. (2001) "Local Area Networks—A Business Oriented Approach,", John Wiley & Sons, Inc., 2000

Table 1: Comparison among several products of Web application server

Product	Operating systems	Support and Integration.	Legacy Integration	DB Integration	Format Integration	Content Management.
BEA Systems WebLogic Server 5.1	Windows NT/2000, Solaris, HP-UX, Compaq Tru64, SCO Unixware, Siemens MIPS with Reliant Unix, SGI Irix, Dynx, AIX, Linux, AS 400, OS/390	J2EE implementation with certified APIs, native HTML support, integrated HTML, JSP, EJB, O/R mapping development tools, integrates with Macromedia Dreamweaver, Visual Age for Java, Inprise Jbuilder, supports XML, COM, CORBA	More than 30 packaged plug-ins including SAP, Clarify, other mainframe, XML and MQSeries, adaptor developers kit for custom application integration	Oracle, MS-SQL Server, Informix, Sybase, any via JDBC-compliant driver	JMS (API+ SMTP), FTP, TCP/IP, POP, IMAP, SNMP	Workflow and personalization product plus native integration with Documentum, Interwoven and other content managers
IBM WebSphere Application Server 3.5	Windows NT/2000, AIX/6000, Solaris, Red Hat Linux, HP-UX, IBM OS/400, IBM OS/390, IBM OS/2	Supports third-party IDEs, integration with IBM VisualAge for Java and IBM WebSphere Studio IDEs, support for Java and non-Java applications, XML, JSP, EJB, COM, CORBA	IBM DB2, Oracle, Sybase, Informix, SQL Server, Lotus Domino, others, via JDBC, Versant EnJin, eXcelon, IMS, MQ-enabled Systems, MQSeries, CICS, TXSeries/Encina, SAP R/3	Oracle, DB2, SQL Server, Informix, Sybase,Versant, eXcelon, Lotus Domino	SMTP, FTP, LDAP, Telnet, TN3270, TN5250, TCP/IP, JMS, JDBC, JTS, JTA, JNI, XML/XLS, EJB, JNDI, J-IDL, CORBA, RMI, IIOP, RMI/IIOP	Workflow, personalization, rules-based engines, proxy, caching, serving, portal serving
iPlanet E-Commerce Solutions iPlanet Application Server 6.0	Windows NT, Solaris, IBM AIX, HP-UX	Includes an integrated J2EE development tool, O/R mapping, integrates with Sun Forte for Java tools and third-party IDEs including WebGain VisualCafe, supports XML, JSP, EJB, Servlets, COM, CORBA	CICS, SAP R/3, PeopleSoft, Tuxedo, MQSeries, Sun JMQ, includes an SDK for developing Enterprise Connectors for other Enterprise Information Systems	Oracle, DB2, Informix, Sybase, SQL Server	JMS, JNDI, XML, RMI, IIOP, RMI/IIOP, JTA, JTS, JDBC, LDAP, NSAPI, ISAPI, SMTP, SNMP, HTTP, XML, JNDI, FTP, TCP/IP, EDI	Native J2EE Workflow development tool and runtime, personalization, content management