

The IBM Application Framework for e-business

by G. Flurry
W. Vicknair

As companies race to transform their businesses into e-businesses, they are discovering that the transformation process is not always straightforward. The IBM Application Framework for e-business (the Framework) is a means for achieving business transformation and a foundation framework for developing and extending e-business processes and applications. This paper describes the Framework, showing it is based on industry standards, including enterprise Java™ technologies, and focuses on enabling complete e-business solutions. It further describes how the system model and application programming model of the Framework and the IBM products that contribute to the platform of the Framework combine to enable a cohesive, prescriptive approach to creating e-business solutions. The paper also shows that the Framework will evolve to support increasingly sophisticated e-business applications, and that IBM is dedicated to working closely with the industry to contribute new technologies to open standards.

An e-business connects critical business systems directly to customers, employees, suppliers, and distributors via the Web to improve time to market, access a broader base of customers and suppliers, improve efficiency, and reduce costs. To achieve these benefits, existing businesses must transform their traditional business processes with e-business applications. New businesses, sometimes called “Net-Gens,” can adopt e-business applications from the beginning to achieve the same benefits. To allow e-businesses to reap the desired benefits, e-business applications must meet some fundamental requirements; they must be:

- Standards-based to ensure portability of e-business applications across multiple client and server platforms and improve flexibility and time to market
- Server-centric to allow e-business applications to be developed and deployed quickly, expand access to a broad range of client types, and offer improved management and deployment capabilities characteristic of modern e-business applications
- Scalable to allow e-business applications to handle the highly variable and unpredictable loads in today’s Web environment
- Available to address the global nature of the Web requiring that e-business applications run 24 hours a day, seven days a week, with guaranteed quality of service
- Secure to address customers’, suppliers’, and other constituents’ demands for secure Web interactions, in recognition of the potential risks of doing business on the Web
- Easy to develop and deploy to achieve lower costs and faster time to market, both critical success factors for any business
- Manageable to achieve lower maintenance costs and contribute to higher availability
- Able to leverage and extend existing assets to improve time to market and reduce cost of development and deployment, while improving security, reliability, and scalability

©Copyright 2001 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

Application frameworks¹ are becoming increasingly important for developing complex applications, because these application frameworks help application designers deal with complexity by allowing the designers to “capture the essence of successful patterns, architecture, components, policies, services, and programming mechanisms.”² As a result, applications are “correct, portable, efficient, and inexpensive.”³ An application framework describes a set of interacting components and services available to an application, describes the responsibilities of components and services, and describes the interactions between the components and services. A developer creates an application by composing and extending the components and services available in the application framework.

The IBM Application Framework for e-business⁴ (henceforth called the Framework) is a cross-IBM initiative providing a very powerful, comprehensive set of open standards, services, and products that address the requirements of e-business applications. The goal of the Framework is to enable businesses, large and small, to quickly and easily build and deploy a broad spectrum of robust, secure, scalable, manageable, interoperable, and portable e-business applications.

Application frameworks typically address specific business domains, such as manufacturing or finance. The intent of the Framework, however, is to address all of e-business and so span multiple business domains. Thus, the Framework may be considered a *foundation* application framework, with guidance for e-business application development at two levels: first, an application framework for e-business applications in general and, second, a set of application frameworks for domain-specific e-business applications.

To achieve the fundamental goal of a foundation application framework, the Framework defines:

- A *system model* that structures the fundamental topology of e-business applications for a broad spectrum of business domains
- An *application programming model*, based on the system model, for designing e-business applications, including an architecture that supports a broad spectrum of business domains, and architectures that support a number of business-specific domains

- A *platform* for developing, deploying, and managing e-business solutions for a number of business domains

The process of creating an e-business application with the Framework begins with the classification of the type of interaction necessary for the business domain. Within the Framework these classifications are called patterns for e-business. Each pattern suggests a set of application frameworks that can be used to create the application. Choosing an application framework involves decisions about the requirements listed above. The implementation uses the appropriate aspects of the general application framework.

This paper describes these aspects of the Framework, with a focus on technical aspects of the application programming model, showing how they contribute to meeting the requirements of e-business applications. The paper also briefly describes some of the ways in which the Framework is evolving under IBM's direction to keep pace with the rapidly evolving requirements for e-business. Finally, the paper discusses related work.

The Framework system model

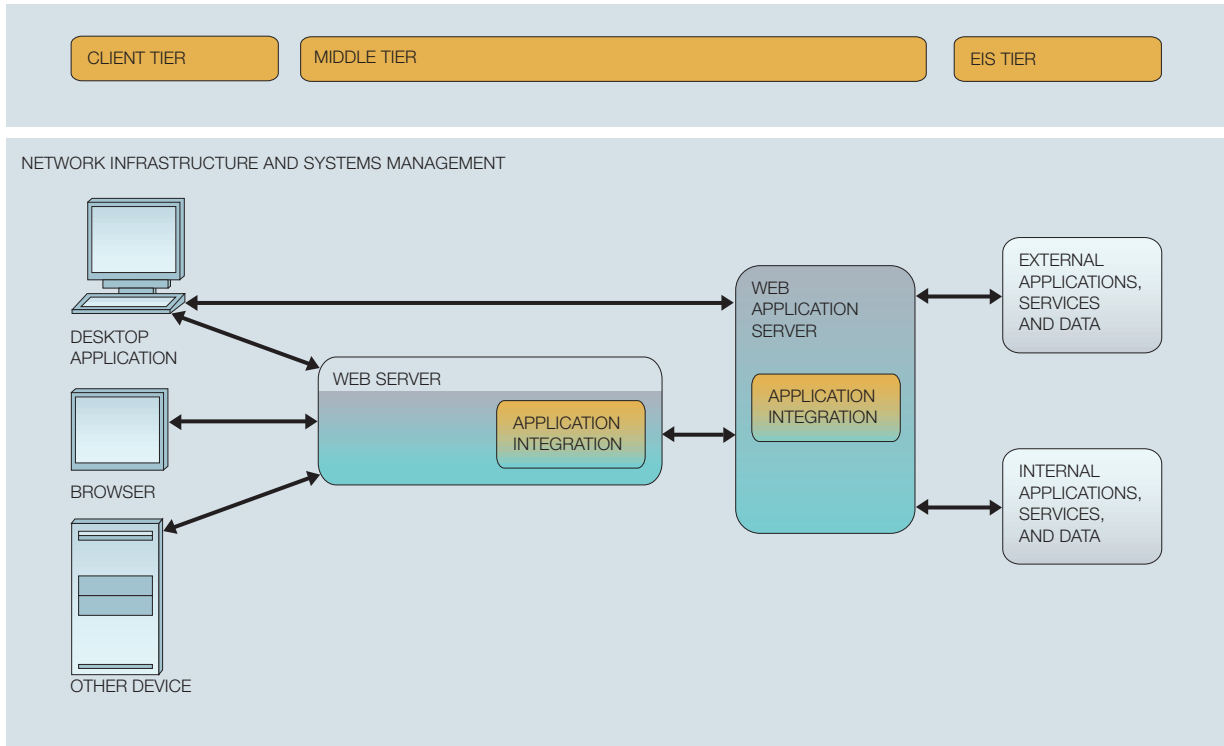
The Framework defines a system model⁵ that supports the design of a broad range of Web-oriented e-business applications that meet all the requirements mentioned previously. The multitier standards-based e-business system model of the Framework (see Figure 1) contains the following elements.

Clients are implemented with industry-standard technologies, interact with the user, and communicate with a middle tier via standard protocols, and send and receive standard data formats. The Framework features the “thin client” to improve manageability, flexibility, and time to market.

Middle tier servers include a standards-based Web server for interacting with the client tier and defining user interaction, and include a standards-based Web application server for executing business logic independently of the client type and user interface style. The middle tier servers incorporate several application integration technologies for communicating with applications, data, and services in other tiers.

The *enterprise information system (EIS) tier* includes new and existing internal applications, services, and data, and also external applications, data, and services from new and existing business partners.

Figure 1 The Framework system model for e-business solutions



The *network infrastructure* ties the tiers together and connects them to the Internet and extranets using standard protocols and data formats. The network infrastructure also provides functions such as service location, security-related capabilities such as user authentication, and technologies to improve scalability.

Systems management implies that the elements of the system model are manageable. System management provides functions such as capacity planning, change control, asset management, and security testing to enhance availability and lower maintenance costs, all based on industry standard techniques.

The characteristics imposed by the Framework system model help the Framework meet the requirements of e-business applications discussed above.

The Framework application programming model

Although the Framework system model offers broad suggestions on the structure of e-business applica-

tions, the Framework application programming model, or APM, describes the details of how to design e-business applications according to the structure suggested by the system model. The APM has multiple levels. At the highest level, the Framework supplies patterns for e-business, proven end-to-end patterns that are appropriate for solving problems in common business domains.

At a lower level, the Framework supplies an architecture that defines the services available to e-business applications and the relationships between the services. The services include application servers, network access, directory, and database. At the lowest level, the Framework architecture defines the set of parts that an application developer can use to compose the application and access the Framework services.

Patterns for e-business. The Framework patterns for e-business⁶ show how to take the Framework parts and tie them together in proven end-to-end patterns that are appropriate for solving classes of common business problems. The patterns for e-business in ef-

fect are domain-specific application frameworks¹ built on the more general application framework to be described below. The patterns provide common architectural frameworks used in e-business application design.⁷ The Framework is intended to support a broad spectrum of patterns for e-business, leveraging IBM's extensive work with customers.

The Framework patterns for e-business are a group of proven, reusable assets that drive the design, development, and deployment processes as follows. An application designer first chooses one of a number of business patterns that most closely matches the interaction between participants in the desired e-business solution. An individual business pattern offers the choice of several application topologies; the different application topologies associated with a business pattern account for differing customer requirements and lead to a particular configuration that focuses principally on the relationship of the application logic and associated data, but does not mandate where middleware runs or where data might be stored. A particular application topology offers multiple run-time topologies, each of which offers a different grouping of the functional and operational components within interconnected logical nodes; each node represents a set of components that provide either infrastructure or application components of the types to be described below. A run-time topology can then be implemented with many physical patterns, each of which maps specific products from the Framework platform, described below, to the run-time topology to instantiate the solution. The actual physical pattern chosen depends on factors such as existing systems, available skills, desired quality of service, scalability, and reliability. Additionally, the Framework provides pattern-specific guidelines for the design, development, deployment, and management of e-business applications based on the specific product mappings chosen.

IBM has completed some of the patterns for e-business solutions to be covered by the Framework. The *user-to-business pattern* covers the general case of users interacting with enterprise transactions and data. This pattern is most relevant to enterprises dealing with goods and services that cannot be listed and sold from a catalog; an example is customer relationship management. The *user-to-on-line-buying pattern* is really a subset of the user-to-business pattern. This pattern is most relevant where packaged goods are sold on line through a catalog via a shopping cart or similar tool; an example is a Web site using e-commerce to offer software to customers.

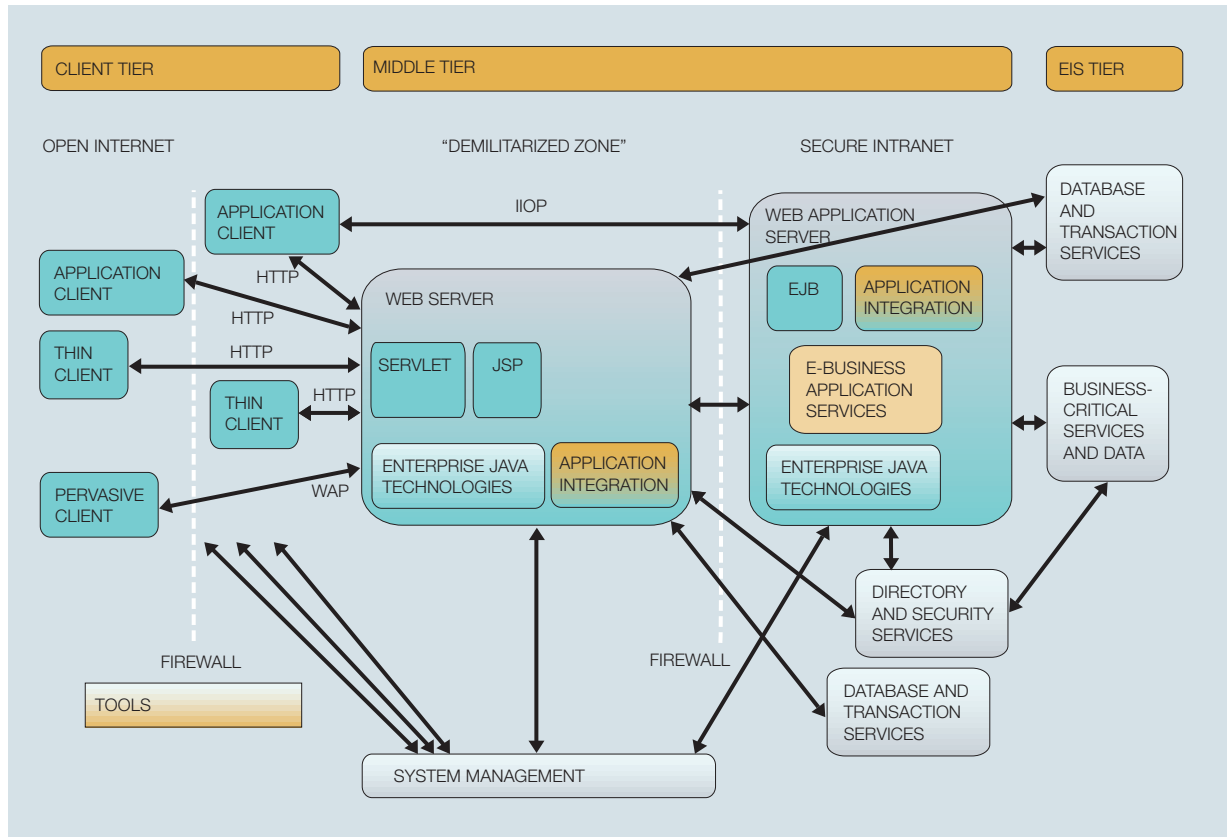
The *user-to-data pattern* covers use of tools to extract useful information from large volumes of disparate information; this pattern can be applied to business intelligence and knowledge management. The *business-to-business integration pattern* supports automated interaction between arm's-length business partners, where a trading partner agreement might be appropriate; supply chain management is an example of this pattern.

The Framework architecture. The Framework software architecture,⁸ shown in Figure 2, structures the services available to an e-business application, defines the parts used and extended to construct the application, and defines the relationship between the parts and services. The parts present a set of interfaces, protocols, and data formats through which the developer can leverage the services of the Framework. Interfaces refer to the application programming interfaces for the parts and services, e.g., the methods of a Java[®] class. Protocols refer to the underlying communications mechanism used to interact with other applications or services; an example is the Internet Protocol (IP) from the Internet Engineering Task Force⁹ (IETF). Data formats refer to the forms of data, e.g., HyperText Markup Language¹⁰ (HTML) from the World Wide Web Consortium¹¹ (W3C), transferred between applications and services and between clients and servers. The interfaces, protocols, and data formats in the Framework are based on formal or *de facto* industry standards from the W3C, Object Management Group¹² (OMG), IETF, International Organization for Standardization¹³ (ISO), the Java community,¹⁴ and other organizations. It is the set of interfaces, protocols, and data formats that form the low-level Framework application programming model.

The Framework architecture, derived from the Framework system model, is composed of the following key elements:

- Application servers for the middle tier of e-business applications, including the Web server and the Web application server, together provide the core presentation services and business logic for e-business applications, database and transaction services, mail and community services, collaboration services, messaging services, and others.
- Clients supported in the Framework can range from devices running standard Web browsers to pervasive devices, such as personal digital assistants, smart cards, and digital wireless telephones.

Figure 2 Framework architecture



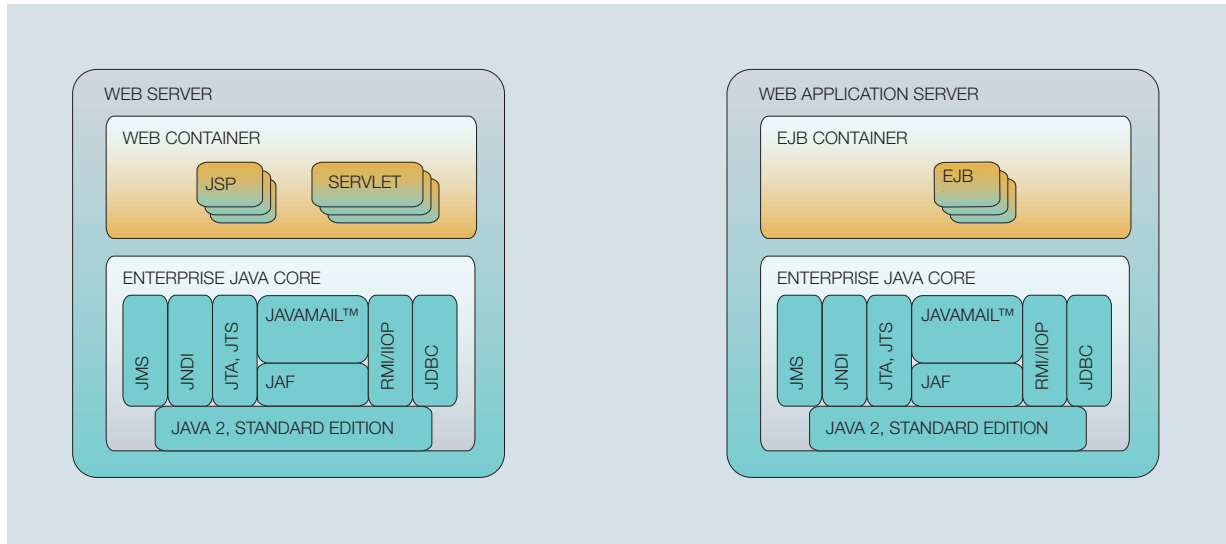
- Application integration software provides access to existing data and applications.
- A network infrastructure that provides services such as basic network connectivity, directory, and security is accessed via standard interfaces and protocols.
- Systems management functions accommodate the unique management requirements of e-business applications across all elements of the system.
- The e-business application services provide pre-built, tested building blocks to facilitate the creation of e-business solutions.
- Tools are used to help build, run, and manage e-business solutions.

The following subsections elaborate on these elements of the Framework and identify the various parts and the interfaces, protocols, and data formats that they introduce.

Application servers. The Framework application servers¹⁵ provide the core services required to develop and support the presentation and business logic of e-business applications. The application servers include the Web server, which offers the capabilities of a HyperText Transfer Protocol¹⁶ (HTTP) server, the Web application server, database services, transaction services, messaging services, mail and community services, and collaboration services.

The Web server coordinates, collects, and assembles Web pages composed from static and dynamic content and delivers them to clients. As such, it supports the mechanisms for locating and invoking the control and presentation logic of the application. The Web application server supports the execution of much of the business logic of an application. Together, these servers offer the application developer access to the other services in the middle tier.

Figure 3 Web server and Web application server detail



The Framework application servers adhere to an important principle of the Framework system model, namely the reliance on standards and, in particular, reliance on enterprise Java technologies. Reliance on enterprise Java technologies, especially in the middle tier, eliminates the dependence of the business logic on the underlying hardware, operating system, and networking infrastructure, thus increasing portability and decreasing development and maintenance costs.

Figure 3 illustrates the use of enterprise Java technologies in the Framework Web server and Web application server and introduces some of the most important parts in the Framework architecture: *components* and *containers*. Components are created by the application programmer using and extending the appropriate enterprise Java technologies. The Framework recommends that the Web server support “Web components,” i.e., Java servlets¹⁷ and JavaServer Pages^{**18} (JSP^{**}) components to provide most of the user interactivity aspects of a Framework application. Both JSP and servlets are more flexible and powerful alternatives to alternative technologies, since they use platform-independent Java technologies to efficiently leverage services in the middle tier. The Framework also recommends that the Web application server support Enterprise JavaBeans^{**19} (EJB) components to provide much of the business logic of a Framework application, especially access

to database and transaction services. EJB offer developers some important benefits. EJB isolate the component developer from the unique characteristics of the underlying database and transaction services, simplifying the development of platform-independent business logic.

Containers provide a federated view of the underlying Framework services to the application components. Containers transparently provide important mechanisms such as life-cycle management, transaction management, security, resource pooling, threading, and state management by providing well-defined run-time environments that are aware of the packaging and deployment characteristics of Framework applications. The separation of components and containers enables “late binding” of the application component to the Framework services at deployment rather than development. Late binding enhances the portability, manageability, and reliability of the application and facilitates team development of e-business applications by isolating the various development roles so that the respective domain experts can each focus on his or her own task. By depending on the mechanisms provided by the container, the various components of the application created independently can adapt to one another and the operational environment at run time, based on configuration information defined at deployment time.

The middle tier containers, shown in Figure 3, support the implementation of application components based on enterprise Java technologies. Java 2 Platform, Standard Edition²⁰ (J2SE^{**}) is the basis for enterprise Java and introduces a number of standard interfaces, protocols, and data formats into the Framework.

The middle tier containers include a number of extensions to support enterprise applications. These extensions include additional interfaces and protocols above and beyond J2SE, including extensions to Java Database Connectivity (JDBC^{**}) for more sophisticated access to database services, the base interfaces for the enterprise Java components, servlets, JSP, and EJB, Java Message Service²¹ (JMS) for access to reliable messaging services, Java Naming and Directory Interface^{**22} (JNDI) for access to directory services, Java Transaction API,²³ or application programming interface, (JTA) for access to transaction services, Java Transaction Service²⁴ (JTS) to define the interface to a transaction manager, JavaMail^{**25} and the JavaBeans Activation Framework²⁶ (JAF) for access to mail services, Remote Method Invocation over Internet Inter-Orb Protocol²⁷ (RMI-IIOP) for robust distributed computing, and deployment descriptors for each enterprise Java component type that defines external resource requirements, security requirements, transaction characteristics, etc.

The Extensible Markup Language (XML) is of increasing importance to e-business, especially in e-business to e-business applications. As part of the Web server and Web application server, the Framework includes support for creating, parsing, and translating XML documents required for working with modern e-business data. The Framework includes a Java implementation of the XML Document Object Model²⁸ and Simple API for XML²⁹ (SAX) interfaces. The validating parser supports parsing, generating, manipulating, and validating XML documents. The Framework also includes an Extensible Stylesheet Language (XSL)³⁰ processor that provides a complete implementation of the W3C Recommendations for XSL Transformations³¹ (XSLT) and the XML Path Language.³² The XSL processor allows a developer to take the abstract semantics of an XML document and transform it into a presentation language such as HTML or into another XML document type.

The database services of the Framework supply the features and functions of an object-relational data-

base to Framework applications. The database services provide query and update access to database information through EJB using Structured Query Language, Java SQL,³³ and JDBC interfaces; this includes Web applications updating the database us-

Containers provide a federated view of the underlying Framework services to the application components.

ing two-phase commit. The Framework also supports connection caching, which greatly improves application throughput and response time by maintaining a cache of open connections to a database.

The transaction services of the Framework support the middle tier servers by providing a highly available, robust, expandable, and secure transactional execution environment. With these capabilities, businesses can provide new services that can scale to service the demands of millions of users. Standards and capabilities supported for application use include EJB and JTS for transactional application support; OMG Object Transaction Services, Object Concurrency Control Service, and the full set of OMG object services; Open Group TxRPC for integrating transactions with remote procedure calls and the Transaction Authority reference model standards for two-phase commit; and Customer Information Control System³⁴ (CICS^{*}) APIs for general-purpose on-line transaction processing.

The messaging services of the Framework provide robust, asynchronous communication and message brokering facilities that support a publish-subscribe model of communication and message transformations. The messaging services provide reliable, asynchronous, one-time transfer of information between e-business applications, independent of platform and protocol considerations, and are especially valuable in e-business to e-business applications. The messaging services can be accessed via JMS and can be accessed through the EJB run-time environment.

The mail and community services of the Framework enable businesses to forge closer relationships with customers, partners, and internal teams. The services

include e-mail messaging, calendaring and group scheduling, chat, and newsgroup discussions. The mail and community services are based on standard Internet technologies and protocols. The standards supported include Simple Mail Transport Protocol,³⁵ Post Office Protocol Version 3,³⁶ Internet Message Access Protocol Version 4,³⁷ and the ISO X.400 for e-mail, Internet Relay Chat,³⁸ for chat, Network News Transfer Protocol,³⁹ for newsgroup discussions, the iCalendar⁴⁰ object model and protocols for calendaring and group scheduling, Lightweight Directory Access Protocol⁴¹ (LDAP) for access to directory services, and the ISO X.509⁴² certificates for authentication.

The Framework collaboration services provide a rich shared virtual workspace and support the coordination of business workflow processes. The collaboration services are also optimized for environments where the network is unreliable, network bandwidth is constrained, or large numbers of end users are mobile and disconnected from the network for significant periods of time. The collaboration services derive from the Lotus Domino** Application Server,⁴³ an open, secure platform optimized to support collaborative Web applications. The collaboration services are accessed using the Lotus Notes** API.

Application integration. An extremely important capability of the middle tier servers of the Framework is their ability to access existing computing assets as well as the need to intelligently integrate business processes that are built on disparate computing architectures. The Framework provides a number of techniques for application integration.⁴⁴ All of these techniques are particularly important for e-business to e-business scenarios.

*Connectors*⁴⁵ are small Java libraries, or “framelets”¹ that provide access to services reached through application-specific protocols. Connectors shield the developer from the typically proprietary network protocols needed to access these servers, minimizing development time and reducing cost. The Framework provides a comprehensive set of connectors to a wide range of existing application, transaction, and data servers. The *connector architecture* of the Framework provides a uniform model in Java for using all connectors and is the basis for a future Java standard extension for connectors. The business logic running in a servlet or JSP on the middle tier accepts an HTTP request from the client and invokes a connector to securely communicate with EIS-tier services on the client’s behalf. The connector architecture is

“pluggable,” allowing additional services and applications to be supported in the future.

Application messaging services provide message-based communication between applications with asynchronous, assured delivery of messages, using the message services described previously. The application messaging services are particularly important where the communication has a high intrinsic value, such as in financial transactions. In the Framework, these services are accessed via JMS.

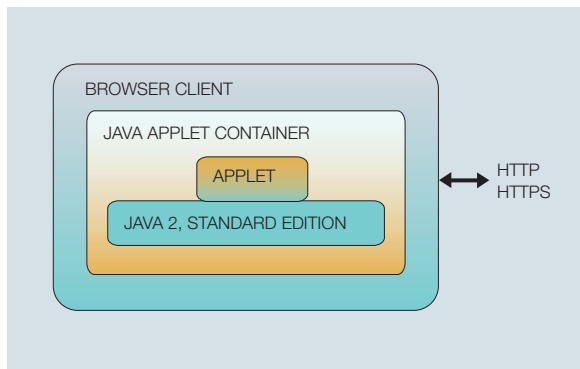
Business process integration and workflow services extend the base messaging services with message brokering, intelligent message routing, and message translation. The business process integration and workflow services provide a central hub to which each application has a single connection. The hub becomes the focus for the extra layer of business process integration capability. The hub maintains knowledge of the business, including processes and rules by which the business is run, and knowledge of the applications interacting. This allows the hub to take appropriate action depending on the type of messages received, for example, intelligent message routing and translation of message formats. XML plays a key role in the message translation process. In the Framework, these services are accessed via JMS.

Component integration services enable object “wrapping” of existing application logic written in any language, extending this logic to object-oriented environments. The component integration services of the Framework allow new applications to be built by integrating applications from all over the enterprise and combining them with transactional support.

Clients. Clients⁴⁶ supported by the Framework span a diverse range of devices from information appliances, such as digital telephones and personal digital assistants, to personal computers. Although the capabilities of each of these devices vary significantly, they are unified by their reliance upon a set of Web-based technologies. The Framework client model exhibits two key characteristics. First, the client model promotes the thin client but, as indicated in Figure 2, supports other clients. Second, in keeping with the Framework system model, the client model is based on standards and technologies being widely adopted in the computing industry.

To the Framework, thin client means that little or no application logic runs on the client and therefore relatively little of the application is required to be

Figure 4 Applet client architecture



installed on the client, even dynamically. Basically, only user interaction and sometimes input validation functions run on the client. The definition of the user interface is supplied by the middle tier, and the application business logic and data storage are generally deployed on middle and EIS-tier servers.

The Framework today focuses on two basic styles of thin client. Both styles interact with a Web server using HTML and HTTP or secure sockets layer (SSL) for improved security. The browser thin client requires only a standard browser to render HTML and interact with the user when the interaction is defined totally via HTML. To provide better interactivity and higher performance, a browser client may exploit the European Computer Manufacturers Association JavaScript^{*,*},⁴⁷ Dynamic HTML, and Cascading Style Sheets⁴⁸ when supported by the target browsers.

The applet thin client supports Java applets running in the browser to provide a richer user experience. The component-container relationship described above extends to the applet client. As shown in Figure 4, an applet container in a Web browser supplies the J2SE required by applets; this container handles security and deployment issues for applets on the client.

Thin clients provide the following advantages for Framework-based e-business applications. A broader range of client devices can be supported since the dependency on device capabilities is reduced. The middle tier servers integrate access to business-critical resources, which simplifies application design, deployment, and management, improves scalability, and provides greater security. Finally, users' appli-

cation environments can be centrally managed and distributed to client devices of different types and in different locations to provide support for mobile users.

There are limitations to the thin client model recommended by the Framework. For example, thin clients do not easily support highly interactive applications. The Framework supports other models as well. For example, rather than going through a Web server, applets can connect directly to back-end services for database access via JDBC. The Java application client, which might be appropriate in intranet deployments, can interact with the user and middle tier services in various ways to provide an experience similar to typical desktop applications.

Network infrastructure. The Framework network infrastructure⁴⁹ provides all tiers in the Framework architecture with a secure, reliable, and scalable distributed networking platform. The network infrastructure, based on open standards, includes network services that support e-business applications in Internet, intranet, extranet, and combined environments; security services that prevent unauthorized access across the network environment; and directory services that locate users, services, and resources in the network.

The Framework network services support various IP-based technologies, including Transmission Control Protocol⁵⁰ and User Datagram Protocol⁵¹ from the IETF to provide the basic session-level protocol throughout IP-based networks; Dynamic Domain Name System and Dynamic Host Configuration Protocol from the IETF to provide dynamic name and IP address assignment and resolution; IP security (IP-Sec) from the IETF to provide network-level security that enables deployment of Virtual Private Networks (VPNs); and SSL to provide session-level security that enables information to be transferred securely between a client and an application server. The Framework network services include a proxy server that acts as a conduit between the client and Web server, which addresses security issues and can cache Web server responses and serve subsequent client requests, improving scalability. The network services also include a network dispatcher, sometimes referred to as an IP sprayer, to link many individual servers as a single, logical server with a single IP address; the dispatcher determines which server responds to a request by rules-based load balancing to achieve optimal scalability and availability and contributes to improved performance.

The Framework security services⁵² include services, protocols, and interfaces that help Framework-based solutions address the security requirements for e-business, such as authentication, access control, asset protection, accountability, administration, assurance, and availability.

The extensible Framework security architecture accommodates a range of authentication mechanisms, including user identifier and password, one-time pass tokens, secret-key cryptography (Kerberos Version 5), and digital certificates based on public key infrastructure⁵³ standards such as X.509. The Framework includes single sign-on technology that enables users to access Web resources and back-end systems using existing user names, making it much easier to integrate with existing applications.

The Framework supports access control to implement businesses' authorization and accountability policies. Security policy information is stored and managed via the directory service. Developers can use the Java Authentication and Authorization Service⁵⁴ to authenticate users and assign and assess access control privileges.

The Framework supports a variety of asset protection mechanisms. It provides secure communications protocols to protect user and application data, such as SSL for session-level security and VPNs via IPSec and related standards for network-level security. The Framework provides data protection and nonrepudiation services to protect stored data and store-and-forward traffic with the Java Cryptography Extension,⁵⁵ the (Domino-based) IETF Secure Multipurpose Internet Mail Extensions (S/MIME)⁵⁶; and the Secure Electronic Transaction⁵⁷ (SET) protocol.

To address accountability, the Framework provides logging services that log all attempts to access corporate resources to ensure that the system is secure and to facilitate analysis of use patterns. Though many IBM products provide extensive logging and audit, standards do not yet exist. IBM is working with the appropriate standards bodies to create standards.

The security services defined in the Framework allow centralized administration of information such as users, groups, and access control. The security administration is integrated with the system management of the Framework.

An e-business must ensure that application components, networks, and data are protected with regard

to confidentiality and integrity. The Framework offers firewalls to control access across network boundaries and to support VPNs, intrusion detection to detect compromised networks, and virus protection to identify and isolate viruses and repair any damage done by the viruses.

To enhance availability, the Framework offers the network dispatcher to balance the load between systems, and offers various technologies for data replication and data backup.

The highly scalable, shareable, and secure directory services of the Framework⁵⁸ provide central storage for user and resource information and enable control of users and resources in the network. These services are accessed via the industry standards LDAP and JNDI. The directory support of the Framework includes a standards-based LDAP client and server that supports basic client authentication, mutual authentication between the client and server via SSL, and Simple Authentication and Security Layer,⁵⁹ a framework for adding additional authentication mechanisms. The directory support uses a common directory schema that defines the rules by which objects are stored and retrieved in the directory; the common directory schema allows different applications to share the same set of objects such as users, groups, roles, and network policies so that the information is not duplicated, and enables directory-enabled applications to be developed independent of a specific directory implementation. IBM is working with industry standards bodies such as the IETF and Distributed Management Task Force (DMTF) to help define standard schema definitions.

Systems management. The Framework includes network-centric systems management⁶⁰ services and tools to manage the complete e-business application stack, i.e., the applications themselves, the services infrastructure and the systems upon which they run, and the network infrastructure that connects them. This can greatly reduce the cost of ownership for e-business applications, and at the same time enhance reliability and availability. The systems management architecture of the Framework is composed of the following key elements. *Management agents* monitor and control resources used by an e-business application; managed resources and the associated agents exist in all layers of the application stack. *Management applications* analyze data collected from agents and initiate commands to agents for problem resolution. A policy-based *management framework* supports communication between agents and man-

agement applications, provides storage for collected information, and supports a mechanism to apply policies and procedures to the management environment. Finally, a *management console* provides a consolidated, central view of resources and management applications.

The Framework systems management services support management of the complete life cycle of an application and cover deployment, availability, op-

Monitoring operational characteristics such as availability and performance is crucial to meeting e-business requirements.

erations and administration, and security. Deployment includes installation and configuration of application parts. The Framework simplifies deployment because of its thin client model. The Framework addresses its deployment requirements with the Tivoli-led, industry-supported Application Management Specification⁶¹ (AMS). AMS provides a nonintrusive method to define management characteristics of an application using the Common Information Model defined by the DMTF. Support includes tools to build AMS files that describe an e-business application.

Monitoring operational characteristics such as availability and performance is crucial to meeting e-business requirements. The management agents of the Framework monitor the resources used by an e-business application and send events, as defined by the appropriate policies and procedures, to management applications that deal with operations and administration. These rules-based applications act appropriately for the situation, e.g., autonomously controlling resources through their agents or notifying an administrator. The Open Group standard Application Response Measurement⁶² APIs are used by management agents to measure e-business application response time. The Framework supports a number of agent-reporting protocols including Simple Network Management Protocol from the IETF.

In conjunction with the security services of the Framework, the system management services provide the tools and services that allow administrators

to manage different security schemes in an integrated fashion, enabling single sign-on and role-based access control for application resources.

e-business application services. The Framework e-business application services⁶³ build on and extend the underlying Framework infrastructure to support new business models. The e-business application services enable a faster, lower-cost method of assembling e-business applications that facilitates rapid development and supports enterprise application integration. These prebuilt, pretested components improve time to market and can improve performance, reliability, availability, and scalability. In particular, the Framework focuses on some key business processes common to a number of industries. The Framework includes business-to-business integration services that extend the core application integration services by enabling businesses to integrate their own computing systems, linking operations, such as supply chain management and logistics, and integrating them with their trading partners' systems; services for customer relationship management, supply chain management, and electronic commerce, including e-marketplaces, all for transforming key business processes into an e-business environment; services for knowledge management and business intelligence intended to help leverage e-business data; and components for core business processes such as general ledger, order processing, inventory management, product distribution, and accounts payable and receivable.

IBM is working across the industry to incorporate or develop standard interfaces, data formats, and protocols for these services and components. Examples include Electronic Data Interchange, Open Buying on the Internet^{**}, and trading partner agreement Markup Language. IBM is also working to provide a more complete set of components for various business processes that can be used as the building blocks of e-business solutions.

Tools. An extremely important aspect of the Framework architecture is the ability to assemble parts to build an e-business application, deploy the application, and manage the application. The Framework provides a broad range of tools for these purposes.

To facilitate introducing the tools, the Framework defines a set of roles⁶⁴ involved in the life cycle of an e-business solution. Each role describes the tasks and the skills required to perform the tasks in various stages in the process of e-business application

design, development, and deployment. Further, each role describes the relationships of that role to other roles in the process. There are three broad roles within the process, each of which has several specializations: An *application developer* creates or extends the parts that can be used to assemble an application; in this context the parts include HTML, images, or other types of content associated with an application, as well as JSP or EJB. An *application assembler* combines the parts into an application that can be installed and managed. Finally, an *application administrator* deploys the application and deals with systems management issues such as performance and security. The Framework supports the different roles involved in developing Web applications by providing tools that target specific roles and facilitate collaboration among members of the development team.

The Framework platform

The Framework system model offers broad suggestions on the structure of e-business applications, and the Framework APM defines the details for designing e-business applications according to the system model. The Framework system model and APM, however, are primarily specifications, not the physical manifestation of the technologies and services, i.e., real products, needed to develop and deploy e-business applications. The Framework platform, although not a single product, encompasses real products that implement a set of product components, tools, and services that together support the development, deployment, and management of e-business solutions designed according to the Framework system model and the Framework APM. The platform helps a customer or independent software vendor (ISV) focus on developing the e-business application rather than assembling and testing a number of individual technologies. The platform enforces the responsibilities of the services with regard to the semantics implied by the APM, and the behavior of the services in aspects such as interoperability, persistence, transactions, and security. For example, a Web application server product would supply a default container that deals with the persistence and transactional characteristics of EJB so that the ISV does not have to write the container.

The use of industry standards by the Framework allows many vendors to supply components, tools, and services for the platform. For example, Web servers produced by IBM or Netscape Communications Corporation can be inserted into the platform. This stan-

dardization enables customers to choose from a variety of vendors who support these open standards. Likewise, solution providers can build their solutions using a variety of differently sourced products.

Of course, IBM supplies products⁶⁵ that implement a comprehensive, integrated, tested, scalable, reliable platform for running Framework-compliant solutions, and supplies tools for building and managing those solutions on a number of hardware platforms. See Table 1.

IBM also publishes a volume of material to help developers use these products to create robust, high-performance, scalable, manageable solutions. For example, IBM Redbooks⁸² offer on-line or printed books with practical development advice; Web sites such as developerWorks⁸³ also offer on-line assistance. IBM also offers courses⁸⁴ that allow architects and developers to learn the Framework technologies and products and acquire the skills required to use them to build e-business applications.

Many IBM customers have used the Framework platform to successfully build and deploy e-business solutions for a number of industries. IBM publishes case studies⁸⁵ for some of these customers, with solutions ranging from commerce to business-to-business transactions. These case studies demonstrate the viability of the Framework and range of e-business solutions supported by the Framework.

The Framework evolution

Because the world of e-business is rapidly evolving, the Framework must evolve to provide better support for existing business models and to support new business models. IBM will enhance the existing Framework and extend it in order to support increasingly sophisticated e-business requirements. Ongoing efforts to enhance the Framework cover higher scalability and better manageability. In keeping with the characteristics of the Framework system model, the enhancements and extensions will be via industry standards where possible. Efforts to extend the Framework include support for emerging e-business models such as portals, application hosting, and pervasive devices.

Portals⁸⁶ provide a single integrated point of comprehensive and ubiquitous access to information, applications, and people. Portals represent the next generation of integrated services and business processes. Portal-specific enhancements include infor-

Table 1 Framework-compliant IBM products

Framework Capability	IBM Product
Web server, Web application server	WebSphere* Application Server ⁶⁶ in three editions
Database and data management services	DATABASE 2* (DB2*) family ⁶⁷
Transaction services	CICS, TXSeries* ⁶⁸
Collaboration	Domino family ⁶⁹
Messaging services, including message brokering and workflow	MQSeries* family ⁷⁰
Application integration	Component Broker and connector support in WebSphere Application Server, Enterprise Edition; messaging and workflow are part of the MQSeries family
Network services	WebSphere Edge Server ⁷¹ provides proxy, load-based dispatching and caching; Tivoli SecureWay* ⁷² provides firewall and other network services
Directory and security services	Tivoli SecureWay family provides a highly scalable, robust directory server, a PKI infrastructure, policy management tool, and single sign-on; IBM Payment products ⁷³ support SET
System management	Tivoli ⁷⁴ family provides a number of system, network, and application management technologies and tools
e-business application services	WebSphere BtoB Integrator ⁷⁵ for linking business processes between trading partners; WebSphere Business Components Mail Analyzer ⁷⁶ for categorizing mail and SanFrancisco* ⁷⁷ for customizable business process components for certain industries; WebSphere Commerce Suite ⁷⁸ for e-commerce; a number of products for knowledge management ⁷⁹
Tools	VisualAge* ⁸⁰ family, Domino family, and WebSphere Studio ⁸¹ for application development; WebSphere, SecureWay, and other product families provide associated tools for deployment and management

mation services for aggregation, document management, search, and data mining; collaboration services for expertise location; information access and integration services for categorization, content integration and personalization, workflow, messaging, and application integration, and support of meta-data; presentation services for customization of the user experience and task management; system management services for notification, logging, user registration, subscription, administration, and user workspace management.

Portal support will require the Framework to help define and implement additional standards such as XML Query Language,⁸⁷ XML Schema,⁸⁸ and Common Warehouse Metadata Interchange⁸⁹ for data analysis and management. IBM products such as the WebSphere Portal Server,⁹⁰ Enterprise Information Portal,⁹¹ and Lotus's Raven⁹² will incorporate the

relevant Framework extensions as they become available.

Application hosting⁹³ refers to providing "rentable" applications over the Web. The application hosting services of the Framework are targeted at application service providers (ASPs). ASPs represent an increasingly important business model, since application hosting benefits both customers, who can concentrate on their core business processes while the ASP provides infrastructure support, and independent software vendors because it creates new channels for their applications and reduces the complexity of developing the applications. The application hosting services of the Framework will enable an ASP to obtain a lower total cost of ownership while providing a higher quality of service to the customer. The Framework will promote an open, shared hosting model with extensions to support the customer

with services for data import and export, storefront, registration, and customer service. The Framework extensions for the application provider include services for application management, resource management, licensing, activity logging, usage tracking and billing, work space, real-time collaboration, and workflow. The Framework extensions for the service provider include services for warehousing, publishing, administration, and service-level monitoring and management. Work is under way to define and implement appropriate standards for the ASP business model, and the Framework will adapt them as available. IBM products such as the Lotus ASP Solution Pack⁹⁴ and services such as ASP Prime⁹⁵ will incorporate the Framework extensions when possible.

Pervasive computing⁹⁶ enables enterprises, telephone service providers, Internet service providers (ISPs), and ASPs to deliver their services effectively, efficiently, and economically to any device, anywhere. Pervasive devices such as handheld computers and smart telephones exhibit characteristics very different from the “normal” e-business client, for example, limited input/output capabilities, processing power, and memory; high latency, low bandwidth, and unreliable connections; and disconnected operation. The diversity of devices, networks, and network connections makes it difficult to provide a universal solution. Pervasive-specific extensions to the Framework include enhanced network services to provide efficient, cost-effective, secure, and reliable communication support for mobile users; transcoding services to filter, enhance, convert, or reformat content to enable access to data by different devices, networks, and I/O mechanisms; disconnected services such as caching, data synchronization, and asynchronous message queuing to support disconnected operation; pervasive device services to enable connection to a wide variety of unique devices; systems management services to address software distribution, configuration management, availability, and inventory management.

Pervasive device support will require the Framework to incorporate mobile IP, being defined by the IETF, the Wireless Application Protocol⁹⁷ (WAP) and the Wireless Markup Language,⁹⁸ the *de facto* standard for the presentation and delivery of information on wireless devices, Bluetooth⁹⁹ for wireless connectivity, and other standards as they emerge. The Framework extensions will be incorporated into products such as the IBM Mobile Net Connect,¹⁰⁰ MQSeries Everyplace,¹⁰¹ DB2 Everyplace¹⁰² that already provides services for disconnected operation,

Table 2 Examples of future patterns for e-business

e-business Solution Area	Business Pattern
Collaboration	User-to-user pattern
Business intelligence; knowledge management	User-to-data pattern
Business application integration	Application integration pattern

Mobile Services for Domino and EasySync for Notes¹⁰³ for general support of wireless devices, WebSphere Transcoding Publisher¹⁰⁴ that provides transcoding services, and WebSphere Everyplace Suite¹⁰⁵ that bundles a number of services to create a secure, reliable, flexible infrastructure supporting pervasive computing.

IBM will continue to extend the Framework to support a broader spectrum of patterns for e-business, including the new business models above. Table 2 offers some examples of the additional e-business solutions to be covered by the Framework patterns for e-business in the near future.

Open standards will remain of key importance to the Framework as it evolves. In some cases, standards may exist. In other cases, standards do not yet exist, and IBM will work with the industry to develop the standards. New technologies¹⁰⁶ will be contributed to industry standards bodies or to the Java community, as appropriate, on a case-by-case basis.

Related work

One of the key principles of the Framework is its basis on open standards. As a part of its application programming model and platform, the Framework leverages many industry or *de facto* standards, including enterprise Java technologies. IBM, Sun Microsystems, Novell, Oracle, and nearly 50 other companies cooperated to create the specification¹⁰⁷ of an open industry standard Enterprise Java Platform,¹⁰⁸ known as the Java 2 Platform Enterprise Edition (J2EE**), which integrates a multitude of enterprise Java technologies. Like the Framework, J2EE is intended to simplify the development and deployment of multitier e-business applications. Although the Framework and the J2EE share a common goal, the scope of the Framework is much broader. The Framework includes a more comprehensive system model, a more extensive application programming

model that supports the more extensive system model, and a robust platform implementation populated with services and the tools required to design, build, run, and manage complete e-business solutions.

IBM was a major contributor to the J2EE specification, and much of its definition derived from earlier versions of the Framework. As the Framework evolves, enterprise Java technologies will continue to be an important component of the Framework application programming model and platform, and IBM plans to continue its contributions to the open industry standard J2EE.

Conclusion

The goal of the Framework is to enable businesses, large and small, to quickly and easily build and deploy robust, secure, scalable, manageable, interoperable, and portable e-business applications. To do so, it defines a system model, an application programming model, and a platform.

We first described the Framework system model that bounds the problem and guides us in the direction of a flexible three-tiered architecture. The system model is standards-based, server-centric and Web-oriented, allowing for broad user and device access, as well as good availability, manageability, and scalability.

We then described the Framework application programming model that fits into the three-tiered architecture and provides the means to implement solutions within that architecture. The Framework APM prescribes interfaces, protocols, and data formats, using industry standards wherever they exist. The APM also provides useful high-level design patterns and business components that aid in rapid application development and deployment.

Finally, we described the Framework platform which, through a suite of IBM products and technologies, collectively known as the WebSphere software platform,¹⁰⁹ provides the environment for supporting e-business application development and deployment, and integrates e-business applications to make complete e-business solutions. The Framework platform is modular and enables customers to start with the services that meet immediate needs, then to add to that as the e-business solution grows.

The Framework is evolving to support new business models and to provide better support for existing bus-

iness models. IBM will continue to work with the industry to embrace or create new standards in the technology areas encompassed by the Framework.

Acknowledgments

We thank Jack Boyce for the inspiration for the paper. We thank Jack and John Meegan for help with identifying and digesting information about the Framework and IBM products implementing the Framework. We thank John and others for assistance in reviewing and improving this paper.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Sun Microsystems, Inc., Lotus Development Corporation, Open Buying on the Internet, or Telefonaktiebolaget LM Ericsson.

Cited references and notes

1. M. E. Fayad, D. C. Schmidt, and R. E. Johnson, *Building Application Frameworks: Object-Oriented Foundations of Framework Design*, John Wiley & Sons, Inc., New York (1999).
2. *Ibid.*, p. xv.
3. *Ibid.*, p. 3.
4. IBM's Application Framework for e-business, IBM Corporation, <http://www.software.ibm.com/ebusiness/>.
5. IBM Application Framework for e-business: The System Model, IBM Corporation, <http://www.ibm.com/software/ebusiness/sysmodel.html>.
6. Patterns for e-business, IBM Corporation, <http://www.ibm.com/software/developer/web/patterns/>.
7. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-Oriented Modeling and Design*, Prentice-Hall, Upper Saddle River, NJ (1991).
8. Application Framework for e-business: Architecture Overview, IBM Corporation, http://www.ibm.com/software/ebusiness/arch_overview.html.
9. The Internet Engineering Task Force, <http://www.ietf.org/>.
10. HyperText Markup Language Home Page, World Wide Web Consortium, <http://www.w3.org/MarkUp/>.
11. World Wide Web Consortium, <http://www.w3.org/>.
12. Object Management Group, <http://www.omg.org/>.
13. International Organization for Standardization, <http://www.iso.ch/>.
14. The Source for Java Technology, Sun Microsystems, Inc., <http://www.javasoft.com/>.
15. IBM Application Framework for e-business: Application Server Software, IBM Corporation, <http://www.ibm.com/software/ebusiness/appsvsw.html>.
16. HyperText Transfer Protocol, World Wide Web Consortium, <http://www.w3.org/Protocols/>.
17. Java Servlet Technology, Sun Microsystems, Inc., <http://www.javasoft.com/servlet/index.html>.
18. JavaServer Pages, Sun Microsystems, Inc., <http://www.javasoft.com/products/jsp/index.html>.
19. Enterprise JavaBeans Technology, Sun Microsystems, Inc., <http://java.sun.com/products/ejb/>.
20. Java 2 SDK, Standard Edition, Documentation, Sun Microsystems, Inc., <http://java.sun.com/jdk/1.2/docs/>.

21. Java Message Service API, Sun Microsystems, Inc., <http://www.javasoft.com/products/jms/index.html>.
22. Java Naming and Directory Interface (JNDI), Sun Microsystems, Inc., <http://www.javasoft.com/products/jndi/index.html>.
23. Java Transaction API, Sun Microsystems, Inc., <http://www.javasoft.com/products/jta/index.html>.
24. Java Transaction Service, Sun Microsystems, Inc., <http://java.sun.com/products/jts/index.html>.
25. JavaMail API, Sun Microsystems, Inc., <http://www.javasoft.com/products/javamail/index.html>.
26. JavaBeans Activation Framework, Sun Microsystems, Inc., <http://www.javasoft.com/beans/glasgow/jaf.html>.
27. RMI over IIOP, Sun Microsystems, Inc., <http://www.javasoft.com/products/rmi-iiop/index.html>.
28. Document Object Model (DOM), World Wide Web Consortium, <http://www.w3.org/DOM/>.
29. SAX 2.0: the Simple API for XML, Megginson Technologies, Ltd., <http://www.megginson.com/SAX/index.html>.
30. Extensible Stylesheet Language (XSL), World Wide Web Consortium, <http://www.w3.org/Style/XSL/>.
31. XSL Transformations (XSLT), World Wide Web Consortium, <http://www.w3.org/TR/1999/PR-xslt-19991008>.
32. XML Path Language (XPath), World Wide Web Consortium, <http://www.w3.org/TR/1999/PR-xpath-19991008>.
33. SQLJ technology, SQLJ.org, <http://www.sqlj.org/>.
34. Customer Information Control System, IBM Corporation, <http://www.ibm.com/software/ts/cics/>.
35. Simple Mail Transfer Protocol (SMTP), Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc0821.txt?number=821>.
36. Post Office Protocol (POP), Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc1939.txt?number=1939>.
37. Internet Message Access Protocol (IMAP), Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc2060.txt?number=2060>.
38. Internet Relay Chat (IRC) Protocol, Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc1459.txt?number=1459>.
39. Network News Transfer Protocol (NNTP), Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc0977.txt?number=977>.
40. Internet Calendaring and Scheduling Core Object Specification (iCalendar), Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc2445.txt?number=2445>.
41. Lightweight Directory Access Protocol, Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc1777.txt?number=1777>.
42. Internet X.509 Public Key Infrastructure Certificate and CRL Profile, Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc2459.txt>.
43. Domino Family of Servers, Lotus Development Corporation, <http://www.lotus.com/home.nsf/tabs/domino>.
44. IBM Application Framework for e-business: Web Application Programming Model, IBM Corporation, <http://www.ibm.com/developer/features/framework/framework.html>.
45. IBM Application Framework for e-business: Connectors, IBM Corporation, <http://www.ibm.com/software/ebusiness/connectors.html>.
46. IBM Application for e-business: The Client Model, IBM Corporation, <http://www.ibm.com/software/ebusiness/client.html>.
47. JavaScript Documentation, Netscape Communications Corporation, <http://developer.netscape.com/docs/manuals/javascript.html>.
48. Cascading Style Sheets, World Wide Web Consortium, <http://www.w3.org/Style/CSS/>.
49. IBM Application Framework for e-business: The Network Infrastructure, IBM Corporation, <http://www.ibm.com/software/ebusiness/netwkinf.html>.
50. Transmission Control Protocol (TCP), RFC-0793, Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc0793.txt?number=0793>.
51. User Datagram Protocol (UDP), RFC-0768, Internet Engineering Task Force, <http://www.ietf.org/rfc/rfc0768.txt?number=0768>.
52. IBM Application Framework for e-business: Security, IBM Corporation, <http://www.ibm.com/software/developer/library/security/index.html>.
53. For background information on public key infrastructure, see Enterprise Security, Netscape Communications Corporation, <http://verisign.netscape.com/security/pki/>.
54. Java Authentication and Authorization Service (JAAS), Sun Microsystems, Inc., <http://java.sun.com/products/jaas/>.
55. Java Cryptography Extension (JCE), Sun Microsystems, Inc., <http://java.sun.com/products/jce/index.html>.
56. See S/MIME Mail Security, Internet Engineering Task Force, <http://www.ietf.org/html.charters/smime-charter.html>.
57. Secure Electronic Transaction, SET Secure Electronic Transaction LLC, <http://www.setco.org/>.
58. IBM Application Framework for e-business: Directory Services, IBM Corporation, <http://www.ibm.com/software/ebusiness/directory.html>.
59. Simple Authentication and Security Layer (SASL), Internet Engineering Task Force, <ftp://ftp.isi.edu/in-notes/rfc2444.txt>.
60. IBM Application Framework for e-business: Systems Management, IBM Corporation, <http://www.ibm.com/software/ebusiness/sysmgwp.html>.
61. Application Management Specification, Tivoli Systems, Inc., http://www.tivoli.com/products/index/module_designer/resources/ams_v20.pdf.
62. Systems Management: Application Response Measurement (ARM), The Open Group, <http://www.opengroup.org/public/pubs/catalog/c807.htm>.
63. IBM Application Framework for e-business: e-business Application Services, IBM Corporation, <http://www-4.ibm.com/software/ebusiness/AppServices.html>.
64. G. Booch, *Object-Oriented Analysis and Design with Applications*, Addison-Wesley Longman, Reading, MA (1993).
65. IBM Software, IBM Corporation, <http://www.ibm.com/software/>.
66. Web Application Servers, IBM Corporation, <http://www.ibm.com/software/webserver/>.
67. Database and Data Management, IBM Corporation, <http://www.ibm.com/software/data/>.
68. Transaction Systems, IBM Corporation, <http://www.ibm.com/software/ts/>.
69. Domino Family of Servers, Lotus Development Corporation, <http://www.lotus.com/home.nsf/welcome/domino>.
70. MQSeries Family, IBM Corporation, <http://www.ibm.com/software/ts/mqseries/>.
71. WebSphere Edge Server, IBM Corporation, <http://www-4.ibm.com/software/webserver/edgeserver/>.
72. Security, IBM Corporation, <http://www.ibm.com/software/security/>.
73. E-commerce, IBM Corporation, <http://www-4.ibm.com/software/webserver/commerce/>.
74. Tivoli products, Tivoli Systems, Inc., <http://www.tivoli.com/products/>.

75. WebSphere Business-to-Business (BtoB) Integrator, IBM Corporation, <http://www-4.ibm.com/software/webservers/tbointegrator/>.
76. WebSphere Business Components Mail Analyzer, IBM Corporation, <http://www.ibm.com/software/webservers/components/mailanalyzer/>.
77. IBM SanFrancisco, IBM Corporation, <http://www.ibm.com/software/ad/sanfrancisco/>.
78. WebSphere Commerce Suite, IBM Corporation, <http://www-4.ibm.com/software/webservers/commerce>.
79. Knowledge Management, IBM Corporation, <http://www-4.ibm.com/software/data/knowledge/products.html>.
80. Application Development, IBM Corporation, <http://www.ibm.com/software/ad/>.
81. WebSphere Studio, IBM Corporation, <http://www.ibm.com/software/webservers/studio/>.
82. IBM Redbooks, IBM Corporation, <http://www.redbooks.ibm.com>.
83. developerWorks, IBM Corporation, <http://www.ibm.com/developer/index.html>.
84. IBM Learning Services, e-business and Application Development, IBM Corporation, <http://www-3.ibm.com/services/learning/spotlight/ebusiness.html>.
85. Software Case Studies by solution area, IBM Corporation, <http://www2.software.ibm.com/casestudies/swcs.nsf/solution>.
86. IBM Application Framework for e-business: Portals, IBM Corporation, <http://www.ibm.com/software/developer/library/portals/index.html>.
87. XML Query Language (XQL), World Wide Web Consortium, <http://www.w3.org/TandS/QL/QL98/pp/xql.html>.
88. XML Schema, World Wide Web Consortium, <http://w3c.bilkent.edu.tr/TR/xmlschema-1/>.
89. R. Cover, The XML Cover Pages: OMG Common Warehouse Metadata Interchange (CWM) Specification, Organization for the Advancement of Structured Information Standards (September 30, 2000), <http://www.oasis-open.org/cover/omg-cwmi.html>.
90. WebSphere Portal, IBM Corporation, <http://www-4.ibm.com/software/webservers/portal>.
91. Enterprise Information Portal, IBM Corporation, <http://www.ibm.com/software/data/eip/>.
92. Project Raven Overview, Lotus Development Corporation, search for Project Raven Overview at <http://www2.lotus.com/>.
93. IBM Application Framework for e-business: Application hosting services, IBM Corporation, <http://www.ibm.com/software/developer/library/ahs/>.
94. Lotus ASP Solution Pack, Lotus Development Corporation, <http://www.lotus.com/aspsolutionpack>.
95. ASP Prime for Developers, IBM Corporation, <http://www.developer.ibm.com/aspprime/index.html>.
96. IBM Application Framework for e-business: Pervasive computing, IBM Corporation, <http://www.ibm.com/software/developer/library/pvc/index.html>.
97. WAP Forum, <http://www.wapforum.org/>.
98. WAP Forum Specifications, WAP Forum, <http://www.wapforum.org/what/technical.htm>.
99. The Official Bluetooth Web Site, The Bluetooth Special Interest Group, <http://www.bluetooth.com/>.
100. Mobile e-business, IBM Corporation, <http://wwwtest.pc.ibm.com/ww/workpad/srg/mobnetconnect.html>.
101. MQSeries Everyplace, IBM Corporation, <http://www-4.ibm.com/software/ts/mqseries/everypace/>.
102. DB2 Everyplace, IBM Corporation, <http://www.ibm.com/software/data/db2/everypace/>.
103. Lotus Mobile and Wireless Solutions, Lotus Development Corporation, <http://www.lotus.com/home.nsf/welcome/wireless>.
104. WebSphere Transcoding Publisher, IBM Corporation, <http://www.ibm.com/software/webservers/transcoding/>.
105. WebSphere Everyplace Suite Overview, IBM Corporation, <http://www-3.ibm.com/pvc/products/wes/index.shtml>.
106. IBM previews implementations of many of the emerging technologies on its alphaWorks Web site, located at <http://www.alphaworks.ibm.com/>.
107. Java 2 Platform, Enterprise Edition, Sun Microsystems, Inc., <http://java.sun.com/j2ee>.
108. G. Flurry, "The Java 2 Platform, Enterprise Edition and e-business," *Java Developer's Journal* (December 1999); note that some of the description may be incorrect since it was written before the J2EE specifications were finalized.
109. IBM WebSphere software platform, IBM Corporation, http://www-4.ibm.com/cgi_bin/software/track3.cgi?file=/software/nfo/websphere/index.html&S_TACT=100AWW10&S_CMP=campaign.

Accepted for publication November 6, 2000.

Greg Flurry IBM Software Group Technology Center, 11400 Burnet Road, Austin, Texas 78758 (electronic mail: flurry@us.ibm.com). Mr. Flurry is a Senior Technical Staff Member with a background that includes interactive graphics hardware and software development, interactive multimedia hardware and software development, "thin client" computing, and Java development for IBM's Advanced Interactive Executive™ (AIX™) operating system. He has 20 patents in these areas. His current responsibilities include definition and prototyping of technologies that support highly scalable e-business applications, and include integrating these and other technologies into the IBM Application Framework for e-business and the Software Group products that implement the Framework.

Wayne Vicknair IBM Software Group Technology Center, 11400 Burnet Road, Austin, Texas 78758 (electronic mail: waynev@us.ibm.com). Mr. Vicknair is a senior software engineer with a background in network protocols, operating system internals, and object-oriented systems. His current responsibilities include definition and prototyping of technologies that support pluggable, service-oriented architectures and APIs for application service providers with a focus on user registration and billing services.