

Preface

Success in the design and deployment of e-business on demand* systems depends on the development of dynamic applications that adapt to the changing business environment and servers that can run those applications with high performance and scalability. Supplying a highly efficient infrastructure to meet these needs is an essential element in the on demand vision. The WebSphere* Application Server is the platform for IBM's distributed operating system supporting on demand computing, and it has evolved to meet the challenges of this dynamic environment.

This issue of the *IBM Systems Journal* is dedicated to the WebSphere Application Server—its architecture, applications, tools, and performance. Twelve papers discuss diverse aspects of the application server, from its role in developing on demand applications to its role as the transaction engine for those applications. We are indebted to N. Halim of the IBM Research Division for his enthusiastic efforts in guiding this issue in its design and in the coordination of the papers.

The WebSphere Application Server is introduced in the paper by Herness et al., which describes elements of its architecture and its fundamental role in the on demand infrastructure. Extensions to WebSphere Application Server to support the grid computing infrastructure, rich Web-based interaction models, service-oriented architectures, and autonomic capability have brought about a transformation from a distributed operating system to a distributed on demand operating system.

Bhaskaran and Schmidt describe WebSphere Business Integration (WBI), an IBM platform offering capabilities for solving challenging end-to-end integration problems. WBI can model, monitor, manage,

integrate, and connect business-operation systems, enterprise information assets, business partners, and collaborative networks of decision makers to address specific business problems. In "Enabling distributed enterprise integration with WebSphere and DB2* Information Integrator," Saracco et al. present another approach to enterprise integration solutions, one which allows system designers to cope with the typical diversity of information sources in a manner which reduces system cost and boosts performance.

Kloppmann et al. describe the use of business process choreography, a powerful tool provided for the WebSphere Application Server, enabling the rule-based specification of business processes in many environments including that of Web services (using the Business Process Execution Language for Web services) in order to build heterogeneous and distributed workflow-based J2EE** applications.

The model-view-controller (MVC) paradigm is a popular means of organizing and designing the functionality of server systems. In "On demand Web-client technologies," Ponzo et al. describe a set of technologies designed to improve the richness of user interfaces and the responsiveness of user interactions for Web clients by projecting the server-side MVC paradigm onto the client system. Enhancing the client in this way enables the exploitation of its inherent native power, resulting in richer user interfaces and local operations, while preserving back-end data integration. Fontes et al. describe the evolution of connector architectures for efficient connection to resources in WebSphere Application Server, from the JDBC** application programming interface to the common connector framework, and the J2EE Connector Architecture. A preview of an important new architecture, the WebSphere Channel Framework

Architecture (a logical extension to the J2EE Connector Architecture) is also presented.

Caching of dynamic content, such as servlets and Web page fragments, in J2EE applications is a key element in improving application-server performance. Willenborg et al. discuss the evolution of WebSphere Application Server performance and scalability features, including caching, and the improvements that have been achieved in this context. An extensive presentation of the dynamic-caching capability of the application server and its effects on application performance is given by Bakalova et al. in “WebSphere Dynamic Cache: Improving J2EE application performance.”

In “Eclipse: A platform for integrating development tools,” des Rivières and Wiegand focus on the area of Integrated Development Environments (IDEs) for application design. The Eclipse Platform, developed by an open-source organization and consortium, serves as the common basis for diverse IDE-based products, and facilitates the programming of applications using components created in a diverse range of technologies, including HTML, Java**, EJB** (Enterprise JavaBeans**), and relational database schemas. Budinsky et al. present an overview of the WebSphere Studio, an IDE based upon the Eclipse Modeling Framework, which offers tools for the development of distributed applications for J2EE servers, including integrated application testing, Web page design, and performance optimization.

WebSphere Portal provides a middleware framework and tools for building and managing portals, which provide end users with unified and personalized access to content, applications, and collaboration services. Will et al. provide an overview of WebSphere Portal, its architecture, advanced features, and some of the concepts being explored for possible inclusion in future releases.

Finally, Johnson and Reimer, in “Issues in the development of transactional Web applications,” discuss issues that are important for the successful deployment and execution of enterprise Web applications in production environments, based upon experience with deployments in various customer environments. These issues are either related to transaction processing or have a bearing on the design of transaction-processing applications. They include failure management, persistent data management, and memory leaks. Understanding these issues is important in the evolution and definition of transaction-

processing runtimes and in building effective tools to support them.

The next issue of the *Journal* contains papers on Unstructured Information Management.

David I. Seidman, Associate Editor
John J. Ritsko, Editor-in-Chief

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Sun Microsystems, Inc.