

Dot Net and J2EE for Web Services

Andrew Chen

Arizona State University, USA

INTRODUCTION

The capabilities of Internet technologies continue to evolve as businesses learn and implement more sophisticated e-business applications in order to adapt to dynamic environments. Web services are an industry-wide response to the need for a more flexible and efficient business collaboration environment. Supported by major institutions and industry leaders, Web services have become the promising method for making e-business information and applications programmatically available via the Internet. Web services are self-contained, modular business applications that have open, Internet-oriented, and standards-based interfaces. Chen, Chen, and Shao (2003) provide a good review of the implications and impacts of Web services to electronic-commerce research and practices.

In order for the Web-services idea to become a working reality, there must be a common agreement on how they will work. Web services rely on a set of standards to support interoperability among applications developed in different languages and running on different platforms or operating systems. Core Web-services standards include SOAP (simple object access protocol), WSDL (Web services description language), and UDDI (universal description, discovery, and integration). The basic idea of Web services is the use of the SOAP messaging protocol to invoke software methods in remote systems. A WSDL file contains service definitions for distributed systems to support the automatic creation of client-side links and the binding to the Web services. UDDI is a registry standard for Web-services providers to publish their Web services. It then can be used by a Web-services consumer to discover (search) Web services developed by Web-services providers.

The grand vision of Web-services-oriented architecture is that Web services can be composed and invoked dynamically to support business processes within and across enterprises. A number of new standards have been introduced to address this Web-services composition issue, including BPEL4WS (business process execution language for Web services), WSCI (Web services choreography interface), and BPML (business process modeling language; Arkin, 2002; Arkin et al., 2002).

Semantic matching and Semantic Web are other developments that enable greater access to services on the

Web (Berners-Lee, Hendler, & Lassila, 2001; McIlraith, Son, & Zeng, 2001). The purposes for the development of an ontology of services, DAML-S (DAML Services Coalition, 2003) under the DARPA agent markup language (DAML) program, are closely related to various Web-services standards. There are very limited interactions between Web-services standards and semantic-Web research. Adding ontology support to UDDI to facilitate Web-services search is a promising direction for future research.

BACKGROUND

Organizations have been motivated to start Web-services initiatives in order to compete in the future business environment. Two major companies, Microsoft and Sun Microsystems, have developed platforms (Microsoft .NET and Java 2 Enterprise Edition [J2EE], respectively) to support Web services. In general, Microsoft .NET is a product suite that enables organizations to build enterprise-class Web-services applications. On the other hand, J2EE is a standard that facilitates building Web services with other products (e.g., used with IBM's WebSphere, HP's Web Services Platform, and Sun's Sun ONE).

Microsoft .NET

Microsoft focuses on three goals as its vision for Web services. First, everything in the system must be a Web service. Second, once these services are created, one must be able to integrate and aggregate these services in simple and easy ways. Third, one needs to have a simple and compelling consumer or end-user experience (Microsoft Corporation, 2001b). According to the Microsoft Corporation, "The Microsoft .NET platform includes a comprehensive family of products...that provide for each aspect of developing, managing, using and experiencing XML Web Services." One unique idea that the .Net framework proposes is its multilanguage support. This idea is to allow developers to program Web services in any language they feel comfortable with (e.g., COBOL, C, Java, VB, etc.). For Web applications, active server page (ASP) technology has been updated to ASP.NET, which provides advanced server-side Web controls and easy-to-create user controls. At the same time, Windows

XP is introduced as the innovative operating system that is aimed at helping consumers interact more directly with Web services. An important part of the XP system is the integration of common tasks with services available on the Internet. With these fundamental innovations, Microsoft has made substantial efforts to advocate Web-services development and implementation.

J2EE

J2EE was designed to simplify complex problems with the development, deployment, and management of multitier enterprise solutions (Vawter & Roman, 2001). The goal of J2EE is to give the user free choice in the use of other products and tools, and to encourage best-of-breed products to emerge through competition. This is the idea of the market and community that J2EE envisions.

J2EE was not originally created for Web services per se; however, J2EE supports the server-side deployment of goods and services with the J2EE application server. Typically, a J2EE application server deploys, manages, and executes three types of standard components, namely, the Web component, Web service, and EJB (Enterprise JavaBean) component. Briefly described, a Web component (e.g., a servlet component, a Java server page [JSP] component) interacts with a Web-based client by connecting with Web-services components for functionalities, and with EJB components for business logic and live business data.

COMPARISON OF .NET AND J2EE

While both .NET and J2EE help business build Web-services applications, many similarities and differences exist between them. There are many studies that compare .NET and J2EE from different perspectives such as Web services in general (Lurie & Belanger, 2002; McGarvey, 2004; Vawter & Roman, 2001), benchmark performance (Microsoft Corporation, 2004; Middleware Company, 2002), architecture (Farley, 2000), and enterprise Web application (Sheil & Monteiro, 2002). While these studies provide many insightful and interesting comparisons, in this section, these platforms will be compared on the basis of factors that can be reasonably measured or ascertained. Furthermore, we also summarize findings of benchmark-performance comparisons between .NET and J2EE at the end of this section.

Portability

Vawter and Roman (2001) state that a key difference between J2EE and .NET is that J2EE is platform agnostic,

running on a variety of hardware and operating systems, such as Win32, UNIX, and Mainframe systems. This portability is real today because the Java run-time environment (JRE), on which J2EE is based, is available on any platform. On the other hand, Microsoft claims that .NET offers portability as well. Even most experts believe that J2EE possesses superior portability, but Microsoft has positioned .NET for portability by offering a two-step compilation process that allows .NET to provide run-time environments for different platforms.

Another aspect of portability is how .NET or J2EE can support a variety of implementations. Companies will likely choose between .NET and J2EE with considerations of implementations on not only intrafirm applications, but also interfirm-application integrations. As suggested by Vawter and Roman (2001), this aspect makes the future of J2EE very bright and is one of the critical differentiators between J2EE and .NET.

Platform Maturity

Platform maturity assesses the potential risk of system failure. Java has been developed and refined since the mid-'90s. The release of J2EE furthered the strength of Java by providing a stable and mature standard for Web applications. On the other hand, not until 2000 did Microsoft start to introduce its .NET strategy and subsequently offer the .NET platform. Further, many developers from different companies and technology communities have used and supported Java-oriented applications. This phenomenon results in better maturity for J2EE compared to .NET, which is young and supported by only one player: Microsoft. As suggested by Vawter and Roman (2001), J2EE is the more mature platform. They suggested that the very underlying fabric of .NET is an overhauled rewrite, and this represents enormous risk compared to the new J2EE features.

Language Support

Extended from Java, naturally all J2EE applications have to be written in Java. This single-language nature of J2EE contrasts fundamentally with .NET, which supports many different languages. While J2EE supporters can argue that a single language provides a more elegant solution, .NET offers an advantage to organizations that would like to leverage its knowledge workers' IT expertise (Lurie & Belanger, 2002). In its design, .NET's multilanguage-support concept is more appealing to a wider range of developers because it can encompass a wider variety of programmers, each knowing different programming languages. However, while a complicated application is written with multiple languages, it is very likely to cause a

3 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/dot-net-j2ee-web-services/12541

Related Content

Quality of Perception in M-Commerce

G. Ghinea and M. C. Angelides (2004). *Mobile Commerce Applications* (pp. 284-302).

www.irma-international.org/chapter/quality-perception-commerce/26463

Deploying Java Mobile Agents in a Project Management Environment

F. Xue and K. Y.R. Li (2003). *Managing E-Commerce and Mobile Computing Technologies* (pp. 245-259).

www.irma-international.org/chapter/deploying-java-mobile-agents-project/25787

Trust in E-Commerce: Consideration of Interface Design Factors

Ye Diana Wang and Henry H. Emurian (2005). *Journal of Electronic Commerce in Organizations* (pp. 42-60).

www.irma-international.org/article/trust-commerce-consideration-interface-design/3465

Modeling Customers' Intention to Use E-Wallet in a Developing Nation: Extending UTAUT2 With Security, Privacy and Savings

Vishal Soodan and Avinash Rana (2020). *Journal of Electronic Commerce in Organizations* (pp. 89-114).

www.irma-international.org/article/modeling-customers-intention-to-use-e-wallet-in-a-developing-nation/241249

A Classification Scheme for B2B Exchanges and Implications for Interorganizational eCommerce

Paul A. Pavlou and Omar A. El Sawy (2002). *Business to Business Electronic Commerce: Challenges and Solutions* (pp. 1-21).

www.irma-international.org/chapter/classification-scheme-b2b-exchanges-implications/6130