SoftWired iBus: An Expert Opinion

Philip Eskelin
Chief Technology Officer
Electronic Commerce
Ernst & Young LLP

750 Seventh Avenue
19th Floor
New York, NY 10019
Phone: +1 212 773 7638
E-mail: Philip.Eskelin@acm.org

January 3rd, 2000

Overview

The Internet is profoundly affecting the way people and businesses interact today. It is largely based on web browsers requesting and receiving HTML content through the HTTP protocol. This mainstream behavior has generated a vast proliferation of clever applications and innovations. However, behavior centered on the notion of events is largely unapplied, and represents the next major step in the evolution of our connected economy.

Fundamental to a next step in the piecemeal evolution of the Internet is real-time, event-driven collaboration between businesses and their customers, partners, suppliers, and employees. When its messaging language and transport protocol are just as pervasive as HTML and HTTP, our connected economy and humans around the planet will naturally find new ways to make things easier, faster, and more "live".

On-line gaming, content syndication, more efficient value chains, global trading webs, and innovative financial applications will rapidly gain momentum, making today's hot concepts look like crude attempts at solving primitive problems with antiquated tools. This new wave will be accessed by billions of devices including desktop computers, devices, and smart appliances.

My thesis is that the iBus product suite has the potential for being *the underlying catalyst* that could make this scenario a reality. As society starts to "get it", we will experience an even more explosive growth than associated with the worldwide web. Performance, flexibility, and quality of service (QoS) requirements will become incredibly demanding, and iBus is uniquely poised to facilitate it. The following sections will provide my analysis of the competitive climate, future demand potential, and position regarding applicable products, standards, and technologies.

Competitive Climate

My position in assessing the iBus products is based on the following directly relevant experience:

- Evaluation of iBus//MessageBus, iBus//MessageServer, and iBus//ANSI-C. Products were tested with a session Java bean run from a Java Server Pages (JSPs) on JavaServer Web Development Kit 2.0, and with a COM object run from Active Server Pages (ASPs) on Internet Information Server 4.0.
- Specific experience at a global investment bank in architecting and implementing a highly reliable and scalable component framework that provided trading floors worldwide with access to real-time data.
- Extensive experience in building open distributed enterprise applications.
- Chief Technology Officer of a group of electronic commerce thought leaders at a large consulting organization, responsible for Fortune 50 accounts, technology vision, and establishing key alliances in emerging technologies.

The iBus family of products is unique in that it provides all of the following features, extensions, and capabilities:

- It is the only server-less JMS-compliant messaging product
- Fully implemented in Java
- Provides quality of service (QoS)
- Allows for multi-protocol subscriptions
- Supports HTTP tunneling (through iBus//Web)
- Supports non-Java integration through a proxy (iBus//ANSI-C)

- Supports IP Multicast (iBus//Real-Time)
- Supports embedded systems (iBus//Embedded)

There are approximately twelve vendors who supply JMS-compliant implementations in various shapes and forms. We will look at three of the major offerings. The key theme you will notice is that each product's JMS support is nothing more than an afterthought for existing application server or enterprise integration and messaging products. They are merely reactions to technology advancements *du jour*.

BEA WebLogic[1]

The BEA WebLogic Server is the leading Java server today. In being J2EE compliant, they support the JMS standard. It provides a point-to-point messaging solution that integrates with WebLogic Server and integrates with their support for distributed transactions.

Its weakness in relation to iBus is that its implementation is centered upon an application server and not an independent implementation. And while BEA says that their JMS implementation provides reliable delivery, it doesn't provide the QoS and real-time via IP Multicast functionality that iBus provides.

Also, since the WebLogic Server JMS implementation based on an existing application server product that integrates with popular web servers such as IIS and Apache, it's not poised to provide the flexibility that will be required for non-browser and non-request-response applications.

WebLogic Server's JMS support doesn't provide tunneling of events through other protocols such as HTTP (which will become increasingly important as the Internet becomes more and more collaborative and real-time). It doesn't have any kind of support for smart appliances or devices that will need an embedded systems version of the various J2EE standards.

IBM MQSeries[2]

IBM's MQSeries classes for Java allows Java programs to connect to MQSeries as an MQSeries client using TCP/IP or Java Native Interface. MQSeries for Java Messaging Service implementations provides support for publish and subscribe and asynchronous message delivery via message selectors and structured message classes.

While MQSeries offers a great solution for enterprise application integration and process automation, its support for JMS merely allows existing MQSeries implementations to integrate with other JMS-compliant messaging clients. It does not provide the same kind of guaranteed delivery, QoS, and real-time capabilities that iBus has, nor does it provide any kind of HTTP tunneling or embedded systems support for smart appliance or device support.

Sun Java Message Queue[3]

While Java Message Queue (JMQ) offers a JMS-compliant application providing property-based routing (intelligent path selection and bandwidth utilization for messaging), it suffers from the same issue when compared with iBus: no support for QoS, web tunneling, or embedded systems. My opinion is that JMQ is more of a reaction and has been included in Sun's "dot com" marketing campaign, but in no way is ready for the kind of support and functionality requirements for serious enterprise applications.

Related Products

In addition to these twelve vendors, related technologies along shared data spaces, also known as tuple spaces[4], exist today. They offer very similar functionality, but are more focused upon collaboration in a distributed environment. Sun JavaSpaces, IBM TSpaces, and Sun Java Shared Data Toolkit (JSDT) are three products/toolkits in this area.

JavaSpaces is built specifically for shared data spaces in the Sun Jini initiative. While Jini is promising and has lots of exciting possibilities, mission-critical enterprise applications of Jini are several years off (assuming the technology is still relevant). IBM TSpaces is a cool product, but it is free, and not supported at a level that is acceptable for companies doing serious development of collaborative applications. JSDT has support for sockets, RMI, and lightweight reliable multicast protocol (LRMP), but suffers from being non-standard and is being left behind as Sun moves forward with J2EE.

---

[1] http://www.beasys.com/products/weblogic/server/datasheet.html
[2] http://www.ibm.com/software/ts/mqseries/api/mqjava.html
[3] http://www.sun.com/workshop/jmq/index.html
[4] The tuple space idea comes from a project at Yale University called "Linda" in the late 80s that established the concept of collaborative applications using a shared data space. Both IBM TSpaces and Sun's JavaSpaces are based on Linda.

Future Demand Potential

Today, Java developers will be looking to a solid Java Messaging Service (JMS) product to provide a J2EE-compliant one-way and two-way messaging and event delivery. For more industrial strength applications, secure and guaranteed delivery of messaging for Java applications will be required by companies who have made significant investment in establishing Java competencies for solution development.

In the future, we can look forward to increased demand for these developers due to growth in Java-based solutions and increased enterprise integration requirements with legacy, hosted, client-server, enterprise resource planning (ERP), and external partner and supplier systems. But another, and perhaps the most exciting dimension, is that of the next major trend in the way people and businesses benefit from the Internet.

The first major wave of the Internet was based on web browsers and web sites. It thrust our world into new applications and innovations and had an effect comparable to that which the Gutenberg press had on society. This next major wave of the Internet will thrust our world into even more exciting new applications that are hard to imagine today.

When the connected economy society as a whole realizes the possibilities and benefits of collaborative applications based on real-time dissemination of events, a whole host of technologies, products, and standards will be developed. They will facilitate the adoption of collaborative applications on web browsers, smart appliances, and other devices such as printers, cellular phones, personal digital assistants, etc.

Here are some of the possibilities we can imagine today:

- **Collaborative, real-time trading exchanges** – providing platforms for exchanges, auctioning, buy-side and sell-side aggregation, bid and proposal, and private trading extranet. This will lead to new, non-traditional derivatives that will in turn be facilitated by them.
- **"Internet on wheels"** – cars will be tied to individuals, and cars will join and discover information and commerce networks during transportation. General Motors is already attempting to break ground with their OnStar and "Web Car" initiatives. Most of the cost of manufacturing will be tied to software development rather than assembly line production.
- **Massively multi-user online gaming** – imagine participating in a Civil War with 500,000 other soldiers and officers in a collaborative, real-time, heterogeneous, and distributed environment.

SoftWired iBus is uniquely poised as a simple, innovative, and flexible offering that provides a comprehensive foundation for this envisioned future. At "Internet Speed", it's not hard to imagine it becoming a reality in 2-3 years.

Applicable Products, Standards, and Technologies

Guaranteed Message Delivery

There is no such thing as "partially reliable Federal Express" – all of it is guaranteed, but there are different options in terms of the delivery's arrival urgency. Therefore, in the virtual world we can expect guaranteed delivery of globally available business messaging solutions to become important, and QoS will be more focused on variable privacy and bandwidth allocation rather than variable reliability.

IP Multicast

IP Multicast is being used today by serious, mission-critical applications in the financial services, telecommunication, and manufacturing industries. This is the *de facto* standard for high-bandwidth event dissemination, and as requirements for collaborative, real-time applications becomes more and more extreme, IP Multicast will play a more important role.

Real-Time Applications for Web Clients

Today we like to go to broadcast.com to listen to radio broadcasts, or go to espn.com to watch a football or baseball game in "LiveCast", or go to on-line trading sites such as datek.com to receive online streaming stock quotes. All of these applications use "streams" to constantly push this "real-time" data to web clients. Many times Java applets are used to render this information on the browser as it's being made available.

The problem is that it's not scalable enough to support the kind of bandwidth requirements and load that will be required for massively collaborative applications. In addition, it's not suitable for any of the smart appliance or device applications we envision becoming available. Therefore, streaming will be seen as a stopgap solution to an inadequate technology. Products like iBus are poised as the provider of real-time dissemination of events and information that will be required to support the next wave of collaborative applications.