INFORMATIONWEEK
October 19, 1998, Issue: 705
Section: Trends

Java's Future -- Sun's Language Is Working Its Way Into Mainstream Computing

Christy Hudgins-Bonafield

Java's promise of "write once, run anywhere" has too often turned into "write once, test and debug everywhere." The vow to extend Java via the browser without touching the client is seen as sales hype. Some even are poised to decry client-based Java as a dead movement.

But if Java hasn't delivered everything that was promised in its first three years, that's not to say it isn't a success. Java is many things, some of which make the enterprise grade, and some that don't. For example, Java is positioned as:

- A language that can cut development cycles for sophisticated applications by a factor of two while building in greater reliability than alternatives such as C++. But it's a language that can't yet be ubiquitous because of performance and compatibility issues. Also, clients based on different Java Development Kits, Microsoft's independent approach to Java, and disparate machine capabilities limit functionality to early JDK versions (see story, p. 80).

- A way to write portable client-server code that's widely perceived as better than any alternative, but is far from perfect.

- An important language for third-party enterprise application development, though Java draws low marks from businesses seeking such applications today.

- A server platform that's gaining momentum both as a way to knit together back-end applications and to create simpler and more functional user interfaces, but which still lacks a critical reference model.

- A distributed computing platform, along with Corba, that must compete

with Microsoft's Component Object Model framework and growing IS dependence on combination frameworks, including those that tap Corba/COM with proprietary APIs and platforms.

- A new product approach built around the still far-flung and nebulous features to be found in emerging application servers.

- An emerging client platform for niche vertical applications and embedded systems that figures to catch on, given Java's ease of development and reliability.

Indeed, Java isn't perfect, but its lightning-quick evolution in a 36-month span makes it a phenomenon in the history of development languages and platforms. Java development already is mainstream in the top 20% to 30% of IT organizations and is on the verge of becoming mainstream in the rest, says Gartner Group Inc. analyst David Smith. Similarly, Forrester Research Inc. finds that almost half of Fortune 1,000 companies already use Java-and that nearly 20% consider it important or critical. By 2000, Forrester predicts, nearly half of the Fortune 1,000 will consider Java important or critical, with 80% of those companies relying on it as their dominant application development language.

Java may even prove to be as strategic to Microsoft as it is to Sun Microsystems, its developer. While Microsoft's ultimate vision may be that of a singular, universal Windows platform, the company has been forced to pay attention to non-Windows platforms. Microsoft needs Java to move effectively into enterprise distributed networking. As long as a competing vision exists, Microsoft must retain some degree of Java one-upmanship.

So, when Microsoft adopted Java, it wrapped pieces of the language, its tools, and its virtual machine in a bear hug. But it bared its teeth when it came to the heart of its own territory-the Windows client interface and the distributed networking architecture mapped out in COM. Microsoft's decision to stamp Java with its own brand raises concerns about Java on the client, where it's most widely deployed in business today.

At the same time, Java is picking up momentum on the server, where it stands to make its most lasting imprint. Java is most noticeable on the server side in quick-to-build midtier applications that extend functionality, such as between clients and legacy databases and applications. Java is also showing up in less-visible ways on the server, such as in its role as intermediary between older, back-end systems. Nike Inc., for example, built a Java application that updates its enterprise resource planning software with procurement information in real time.

Java helps many businesses-especially those that still rely on terminal emulation-to jump several technology hurdles at once. Rather than fashion new applications and retrain users, some businesses choose

midtier servers that bring the mainframe to the desktop browser. As a result, emulation is rapidly approaching commodity status on these servers. The downside is getting servers to scale without spending a fortune, since they must handle emulation and pipe graphics to the client. Some companies also find great value in taking a once-complicated terminal emulation interface and simplifying it so that it can be used not only by sales reps, for example, but also by outside customers.

Java's simplicity goes a long way toward explaining its burgeoning popularity. Java makes it easy to extend sophisticated logic to the server while maintaining reliable performance characteristics. Good Java programmers say the language can increase their productivity by a factor of two over C or C++. Best of all, without the pointers and memory-management headaches of C, the resulting applications are more stable, reliable, and modular. That's reason enough for many companies to use Java, but managers also welcome Java's reputation for being much more portable than C or C++, the two main languages used for sophisticated enterprise applications that may one day need to be ported.

Although Microsoft positions Visual Basic as the functional peer of Java and C, few large users or analysts agree. Jim Turley, senior editor of the Microprocessor Report newsletter, says programmers typically view VB as something teenagers learn. If you're 35 years old and still writing in VB, he says, "it's like saying you fingerpaint for a living."

But fingerpainting must be a big business, judging by the huge number of programmers who value the ease of writing in VB and find Java too obtuse. A middle ground may lie with new Java-based development platforms like the one associated with an application server from SilverStream Software Inc. Programmers say SilverStream's big advantage is that its development environment resembles that of a fourth-generation language. It's sufficiently alluring that even Microsoft shops, such as Cellular Technologies Inc., have standardized on SilverStream's Java products.

Aberdeen Group analyst Tim Sloane says Java could go even further to attract the base of VB developers now writing for Windows. If Sun were willing to create an integrated development environment, Sun's pure Java would be accessible to the 80% or 90% of developers committed to the Microsoft platform, Sloane says. But Sun isn't doing that, he adds, because "it would appear to the industry that it's buying into Microsoft's corruption of Java."

Growth Patterns

As things stand, Java is unlikely to overshadow the massive amount of development being done in VB anytime soon-if ever. Gartner Group's Smith predicts that the growth in VB usage will continue unimpeded by Java.

C++ is another story. Java, combined with HTML and JavaScript on the client, appears to be displacing C++ as the language of choice for independent software vendors and businesses with high-volume/high-functionality platforms. However, since many of today's most savvy businesses and software vendors deploy apps based on multiple languages, and languages tend to be entrenched, the consensus is that total displacement of even C++ is likely to take five to eight years.

If this is the case, Java use is evolutionary, not revolutionary-but it's a rapid and clear evolution. Even on Windows NT, the languages of choice are trending toward toward Java and HTML, especially for Web applications. In a survey of 159 IT managers, Strategic Focus found that 35% of NT development is expected to be in Java within two years, with almost 29% in HTML. Alternatives such as VB, Dynamic HTML, JavaScript, and Perl all get lumped into the "other" category.

In the enterprise, Java gets used when development time or portability is an issue, while C++ gets the call when reliance on dynamic memory allocation spawns performance concerns. For intranet use of an application, Java may be deployed on the client. On the Web, that part of the application might be rebuilt using HTML, Dynamic HTML, or JavaScript. For many businesses, Java is best for building lightweight server processes, such as servlets. These servlets can in turn provide heavyweight and complicated logic. Tools to access back-end databases are also a good use of Java on the server.

Steve Gimnicher, VP of Computer Network Technology Corp., which provides tools for reengineering and building applications for mainframe and other legacy systems, thinks the best opportunities for Java lie with the server. CNT is focusing on applications such as one built for AT&T's Inbound Call Receive Center, which uses HTML for the thin client for 400 customer-service representatives and taps Java on the server to access customer records and other data. While CNT relies mainly on a non-Java midtier connectivity product, it expects this new Java solution to account for 25% of its sales in less than 12 months.

Businesses also use Java to develop apps that, for scalability reasons, may one day reside on servers other than those now in use. Java portability is far from perfect, but John Neffenger, chief technology officer and founder of Java chat software provider Volano LLC, insists that his company's software-which runs on 23 operating systems, 10 Java virtual machines and 17 hardware platforms-proves that "write once, run anywhere" works.

"People thought they could write once and not do any testing," Neffenger says. "They got carried away and thought they could write [code] on their notebooks with Windows 95 and never do any testing. Still, Java is a heck of a lot better than writing source code for every platform you'll

deploy on and hiring 12 people for the compilation-dependent platform switches."

The next significant challenge for businesses deploying Java is standardizing on tools and development environments. Many companies still hesitate to commit their entire enterprise to a single tool provider. They cite Java's still-nascent status and fears that tool providers will include proprietary extensions in their toolsets that harm Java's portability. Meanwhile, some are waiting to see if any of the innovative smaller tool providers weather the market.

Some leading development tools are Inprise's JBuilder, SilverStream's application server development environment, Symantec's Visual Cafe, and IBM's VisualAge for Java.

Microsoft's Push

Expect Microsoft to push forward with new Java tools and environments as well. It's already clear that Microsoft's new development environment, Visual Studio (slated to be launched this fall), is intended for Java, VB, Visual C++, Visual FoxPro, Microsoft Access, and Source Safe.

Visual J++ 6.0 is Microsoft's newest development tool for ActiveX controls, Windows executables, Java scripts, HTML, and Java applets on Microsoft platforms. It not only supports Microsoft-flavored Java, but also provides direct access to the Windows platform.

But Bill Dunlap, Microsoft's Visual J++ 6.0 product manager, runs hot and cold on Java. He expects Visual J++ to be used as extensively as C++ and VB as a Windows development tool by 2000, and he'd be happy if 25% of COM development was with Java and 50% with VB. But he puts his emphasis on Java with Microsoft overlays and strategies-COM for distributed computing and Dynamic HTML on the client. He also stresses that Microsoft research shows 3 million developers use VB compared with about 150,000 using Java.

Dunlap dismisses the notion of Java's portability, arguing that businesses buy hardware for a specific reason: "Java is a great language for accessing the great capabilities of a specific machine. It's a pipe dream to think people will move large-scale apps from machine to machine," he says. He argues that Java isn't the development cornerstone of the enterprise, but that its primary use will be as midtier software between the client and legacy app on the back end.

"For performance-critical systems, you want to use Visual C++," Dunlap says. "For the fastest bang-'em-out apps, the environment is Visual Basic, and the way to build Windows and Web apps with a single framework is Visual J++." Dunlap doubts Java performance will ever match that of C++, and says Microsoft is researching how to create a native Java

compiler for Windows to improve Java's performance.

What would boost Java in the market? Third-party applications wouldn't hurt. If the enterprise has yet to standardize on Java development tools, it comes as no surprise that enterprise Java applications remain works in progress. Last summer, Sun and IBM pegged the number of third-party Java applications at about 1,200. That's probably one reason that about half the businesses responding to an informal Network Computing magazine E-mail survey give Java a grade of D or F today when it comes to availability of enterprise applications. About 8% say Java will continue to be below average by 2000.

Lew Tucker, director of strategic relations for Sun, says Java development is much further along than the 1,200-app statistic suggests because that figure doesn't reflect the use of Java as an integration platform for packaged enterprise applications such as those from SAP and PeopleSoft, or the use of Java to build customized applications.

But the issue isn't that enterprise applications won't ever be available-it's that they can't come soon enough. John Rymer, president of Upstream Consulting, says he expects enterprise-ready products to begin arriving by early 1999.

Another factor that will determine Java's long-term success in the enterprise is the establishment of a single reference model for Enterprise JavaBeans. Without such a model, users say, there's no guarantee of EJB interoperability.

Users and analysts say an EJB reference model has yet to emerge simply because Sun is overwhelmed by the breadth of its Java undertaking, coupled with the company's recent reorganization. Analysts speculate that because Sun's internal product development is complemented by the control it exerts over a reference implementation, it will resist farming out so vital a task. That, in turn, will slow Java's acceptance in the enterprise.

So what's Java's future? Because Java is only three years old, Java veterans and proficiency won't pop up overnight, even if, as Gartner Group's Smith concludes, "Java is rapidly becoming the teaching language at universities."

Businesses with the largest Java programming teams have attacked the staffing problem by retraining C and C++ programmers and establishing mentorships for programmers with less experience.

But it isn't the cultivation of Java programmers, the creation of a reference model, or even technology investment that will push Java out across the enterprise. Gartner Group's Smith says it's something much simpler: changing the perception among old-line IT staffers that Java is

just for the Web.

Christy Hudgins-Bonafield is business and trends editor at Network Computing, a sister publication of InformationWeek