

## Java: The inside story

We interview Java's creators to find what they had in mind.

By Michael O'Connell  
SunWorld Online

July 1995

Abstract

Poised to fill World Wide Web browsers everywhere with animation, audio, and real-time interactivity, Sun's Java language has survived an odyssey through consumer electronics, PDAs, set-top boxes, and CD-ROMs. While some of these areas may yet be exploited by the language formerly known as Oak, the Internet is Java's launch pad. How'd it get there, and what's its destiny? Will it successfully cross over into the (gasp!) non-Unix marketplace?

One would think that a key component in any business's current strategy for success would have been deliberately created based upon a clearly defined mission. But in the case of Sun Microsystems Inc.'s Java programming language -- the environment that turns static Web pages into interactive, dynamic, animated documents bolstered by distributed, platform-independent applications -- it seems the solution preceded the problem.

As Java creator James Gosling explained in a recent interview with SunWorld Online, the genesis of Sun's Web-enhancing technology can be traced to early 1991, when a small group of Sun engineers formed to explore opportunities in the consumer electronics market. At the time, the World Wide Web was still in the drawing rooms.

### Consumers vs. engineers

"We were trying to build a distributed system that would make sense as a business [product] ... to sell modern software technology to consumer electronics" manufacturers, Gosling says.

Gosling, 40, joined Sun in 1984 (coming from IBM's research division) and soon afterward began work on the technically impressive but commercially unsuccessful NeWS windowing system. He also wrote GOSMACS, the first EMACS text editor implementation in C.

During this consumer electronics effort, eventually referred to as the "Green" project, Gosling and fellow project engineers learned a great deal about the value of qualities such as reliability, cost, standards, and simplicity -- top priorities in the consumer marketplace. (See the timeline sidebar for additional details.) In contrast to workstation users, who typically want lots of power and will tolerate (and sometimes seemingly demand) high prices, steep learning curves and various bugs in exchange, consumers demand low-cost, bug-free and relatively simple, easy-to-use products.

"Consumers don't care which CPU is inside," says Gosling. They don't appreciate big or powerful RISC-based processors, which are "expensive and proprietary." To compete in the consumer electronics market, companies "treat CPUs as commodities" that can be swapped for lower-cost alternatives nearly instantaneously, and invest resources into backward compatibility and adherence to established standards in long-lived machines, such as toasters and televisions.

Gosling notes that just as modern toasters with embedded electronics employ the same basic user interface as his mother's 42 year-old toaster (which "still works just fine"), so must other consumer electronics products. Television followed a similar path of backward-compatibility when color broadcasts began (today's TV signals can be viewed on 1950s-era black-and-white sets), and faces its next challenge in making the move from analog to digital signals.

### Starting with C++

To make development a more platform-neutral process (and thus accommodate the consumer market's demand for CPU flexibility), Gosling began by extending the C++ compiler. Eventually, however, he realized that even with lots of extras, C++ would not suffice. Thus, Oak was conceived in mid-1991. (The name came

to Gosling when, while creating a directory for the new language, he glanced out his window, and spotted a tree. But the name didn't survive a trademark search, and was dropped in favor of Java.)

"All along, the language was a tool, not the end," Gosling says. "This was nice in a number of ways. The goal was never 'Let's take on C++,' [but] to build a system that would let us do a large, distributed, heterogeneous network of consumer electronic devices all talking to each other."

In the fall of 1992, after what then-project engineer Patrick Naughton characterizes as "massive amounts of hacking on Oak, the Green OS, the UI, [and the] hardware," among other things, the Green project delivered "\*7" (as in the star on a telephone keypad), the PDA-like device that Gosling calls a "handheld remote control."

"In 18 months, we did the equivalent of what 75-people organizations at Sun took three years to do," boasts Naughton -- "an operating system, a language, a toolkit, an interface, a new hardware platform, three custom chips...using new risky technology at every turn. We pulled out all our teeth and put them in each others' mouths."

Naughton, 30, was the project lead on Sun's OpenWindows user environment before joining the secret Green team.

The \*7's small form factor helped emphasize the small size and efficiency of the code, which was the core technology. The product was demonstrated around Sun and impressed important people like Scott McNealy and Bill Joy, but the next step was uncertain.

### **Request for proposal ("Instead of writing a few papers and moving on with life...")**

While the team was working on Oak and \*7, team members Ed Frank (hardware/technology) and Mike Sheridan (business planning) wrote business and technology road maps for a company in the mold of Dolby Labs that would create and license technology and get its logo alongside Dolby's on consumer electronics products. They had finished several versions of the plans by the time of the \*7 demo. But in early 1993, as Sun weighed Java's options, the Green team (now incorporated as FirstPerson Inc.) got wind of a request for proposal from Time-Warner for a set-top box operating system and video-on-demand technology. "A perfect fit," recalls Gosling.

FirstPerson quickly zeroed in on the set-top box OS market, and placed a bid with Time-Warner. But despite having been told that they had the best technology, Sun did not win the bid, due to what Gosling and Naughton characterize as wholly non-technical reasons, such as business politics. "[SGI's Jim] Clark sold his sword to get the deal," Naughton says.

FirstPerson kept trying to pursue set-top boxes until early 1994, when it concluded that "the market wasn't real," Gosling says. "A lot of people hyped things beyond reason." Apparently, the interactive TV market still isn't ripe. Two recent examples: An interactive cable TV trial of 50,000 homes in Omaha, NE put together by U.S. West (network), 3DO (set-top boxes), and DEC (video servers) was cancelled this spring after two years; and Viacom just stopped work on its full-scale test in the San Francisco bay area -- to refocus on applying its miles of cable TV lines to a tried and true market: telephone service.

Naughton says he waged an eventually successful campaign to stop pursuing set-top boxes and instead focus on online services, CD-ROMs, and desktop platforms. FirstPerson dissolved, and about half of its staff moved to Sun Interactive to develop digital video data servers. But a few people still pursued applying Java's technology to network-based desktop systems.

### **Spinning into the Web**

By mid-1994, the World Wide Web was big. "We realized we could build a really cool browser," says Gosling. "It was one of the few things in the client/server mainstream that needed some of the weird things we'd done: architecture-neutral, real-time, reliable, secure -- issues that weren't terribly important in the workstation world. So we built a browser."

By early fall, Naughton and fellow Sun engineer Jonathan Payne finished writing WebRunner, a Web browser written using the Java language. This early incarnation of HotJava showed off Java in a new light, and a demo impressed SunLabs director Bert Sutherland and Eric Schmidt, Sun chief technology officer -- no doubt in part because they could envision Sun reaping rewards.

## What it all means

"The browser equals something that creates a market" for tools, servers, development environments, Gosling says. And Java plays a key role in those tools. "In the pre-Java world, you look at the WWW world, and a page is essentially a piece of paper. In the Java world, a browser becomes a framework. Content providers are empowered to describe behavior and data formats and everything."

More generically, Gosling envisions Java will make people rethink what client/server computing is about. "In the standard model, you have some databases, write a bunch of clients that interact with the database, and build some front end." In this model it can be difficult to organize multiple systems and maintain upgrades, especially when they come from different places, Gosling says.

With Java and Web tools, in contrast, you have inherent organization, Gosling says. "If you build the client side of an application in Java, then launching a client app becomes just switching to a page. Installing is trivial -- just put it on a Web server. And there are no ports, just one version of the application." Already, Gosling says, lots of companies organize databases as Web pages using the Common Gateway Interface (CGI) -- the specified standard for running external programs under an HTTP server.

## Looking ahead

At this point, Gosling says the Java language is fairly solid, and he doesn't see any major changes, just some fleshing out. It's the browser that needs work, and that's where the team's efforts are largely concentrated. Gosling says it should be complete by the end of summer. What then? "We've been building a tool, now we'd like to use it," says Gosling -- to build some commercial products, including Web authoring tools.

And none too soon. After the browser, "Sun's biggest challenge is authoring tools" to help develop content, says Dwain Aidala, VP and general manager of Mitsubishi Electronics' North American Multimedia Business Center.

Aidala, whose company has been working with the Java technology in embedded systems for the last two years, says Sun also should make Java a bit lighter and expand its potential applications beyond the Internet and the Web. "Java is limited primarily by how small they can make the interpreter," Aidala says. "It has the potential to go anywhere there's networked processors. ... Telco, CATV, closed systems ... all in the future."

Indeed, Sun acknowledges its efforts to employ Java technology in interactive televisions/set-top boxes, handheld devices, PDAs, telephones, VCRs -- even light switches. "The Internet is the first platform, the perfect way" to introduce Java, says Kim Polese, senior product manager of Java and HotJava.

Still, Java has a ways to go. Of the 713 people who filled out SunWorld Online surveys at its June prototype Web site, only 13 people (1.8 percent) use HotJava -- currently the only choice for taking advantage of Java applets -- as their primary browser. Netscape Communications' agreement to license Java technology means a Java-aware version of the Netscape Navigator browser could appear around the end of the year; with the number-one browser up to speed, Java will become a common component in the broader Web community.

## Competition? What competition?

Sun insists it has no direct competition. Although nobody is advertising an equal technology, languages such as Kaleida Labs' ScriptX and Lingo (Macromedia Director's animation scripting language) have come up in the same breath as Java. And General Magic offers somewhat similar technology, its Telescript messaging and agent-based system. But as Polese explains, Telescript differs greatly from Java.

Instead of the server sending information based on client requests, the servers sends things when the server deems appropriate, so code comes to clients without clients asking for it. This raises security questions,

Polese says. Since Telescript is aimed at phone networks and PDAs, not desktops, the security issues may not apply.

When one considers the applications of the technology, ever-present Microsoft enters the ring. Beyond the typical browser companies or service providers, and beyond Microsoft's monolithic force and looming Microsoft Network, Microsoft has Visual Basic. While this language lacks the portability of Java, it can run on any Windows-based PC. And while VB applications may never be as light as the heaviest Java applet, they are downloadable. "VBA [Visual Basic for Applications, a built-in programming language for MS applications that replaces older macro languages and provides a unified programming interface to the outside world] is interpreted, extensible and downloadable," says HotJava co-author Patrick Naughton.

"Visual Basic can and will do what Java does," says Naughton, who is now the vice president of technology at Starwave Corp. (Starwave is Microsoft co-founder Paul Allen's Seattle-based interactive consumer products and services company and maker of the popular ESPNET SportsZone Web site, among others.) "Visual Basic already has 30 million users." Who cares about cross-platform issues when you own the PC market?

"Visual Basic makes up for its inferiority as a programming language for serious object oriented design by having an easy to use, visual application construction tool which is already in use by a massive developer community."

"The only thing it doesn't have that matters is security. That's its Achilles' heel -- it's gonna kill 'em." Still, Naughton adds, "PC viruses have been around for 15 years" and haven't prevented the dominance of Windows and DOS. "The issue is that today's viruses have been spread by the rather slow method of floppy disks and downloads from BBSes. Once MSN [the Microsoft Network, due to launch August 24] has people downloading VBA chunks of code as a matter of course, the virus threat is more tangible." In an environment that downloads things for a living, security takes on paramount importance.

In addition to Visual Basic, Microsoft hopes to deliver a tight integration of online network-based services with CD-ROMs. This would allow things like Encarta CD-based encyclopedias to be kept current via supplements seamlessly added via the online network.

"I think this is what Microsoft Network is going to be," says Mark Winther, VP of worldwide telecommunication at IDC's Link Resources Corp., a New York City-based market research and analysis company focusing on the consumer information market. "It doesn't give you live, dynamic, 3-D home pages, but is very powerful and close to what the mass market wants."

Despite all its strengths, Microsoft's concern about Java indicates Sun has the upper hand at this point. "Bill [Gates] knows about and asks about it," says Naughton. "Microsoft sent two guys to pick my brain."

## Marketing strategy

Rather than hoarding the technology, Sun has realized the importance of generating broad product interest and acceptance, and therefore freely offers the binaries -- and even the source code -- of key Java components via the Internet.

Sun plans to license Java technology widely to companies such as Netscape that offer Web browsers, online service providers, and software OEMs to "make the long term more solid," Polese says.

"The basic strategy is to license Java to people who have a need for network-centric applications," says Eric Schmidt, Sun's chief technology officer. "The first and obvious target is the browser world. Nothing in the design of Java limits it to Unix or any other operating system. ... It needs to be on all [major] platforms to be successful, and we are going to make sure that happens."

Secondly, Sun is working with third parties to build development tools and object libraries, Polese says, noting that some of these tools are designed for nonprogrammers and offer "completely WYSIWYG" interfaces that let Web creators do things such as drag and drop images and other objects into pages.

"There's no reason why layers can't be built on top" of Java, ranging from scripting tools to more sophisticated tools such as RAD Technologies' PowerMedia authoring tool or Dimension X's animation tools, Polese says.

"What we would really like to see are new types of applications developed using this technology," Schmidt says. "We are trying to avoid the fate of NeWS [Sun's proprietary windowing environment that lost a standards battle with the X Window System] by working more aggressively with everyone in the industry. I think the terms are low enough and the value high enough that we have a good chance of having most of the movers and shakers sign up."

Perhaps these combined efforts will cause tomorrow's Internet -- as well as other networked computer environments -- to overflow with ubiquitous Java objects.

"The people [Sun] needs to market Java to are not those reading [programming-related newsgroups]," says Naughton, but less sophisticated users. "If average consumer can see these things and use them, they'll demand it. Naughton says Sun needs to land deals with major commercial players such as America Online and CompuServe.

"The key is not Sun, but how many service providers and publishers use Java on their servers," says IDC/Link Resources' Winther. "I see no reason why an increasing number of new sites won't employ it. ... It's real powerful."

## About the author

Since writing this cover story, Michael O'Connell has launched JavaWorld (<http://www.javaworld.com>), IDG's monthly Web-only magazine for Java developers and professionals, where he is editor-in-chief. Previously, he was an editor with SunWorld and a reviews editor with Advanced Systems magazine.

## Sidebar: Lessons Sun learned from the consumer market

Because of the consumer electronics industry's utter lack of commitment to particular processors, Gosling realized early on that "apps couldn't be compiled to a specific CPU." He and fellow Sun engineers learned a few other key lessons as well. In the consumer market, product developers must:

- Allow interfaces to evolve gracefully.
- Tackle security and reliability problems.
- Recognize that multitasking helps a lot; existing standards ("which come from the teletype of 25 years ago") don't.
- Consider alternatives to C and C++, whose features encourage creation of unreliable, fragile code.

## Sidebar: Java's evolution: a timeline

Java's development as outlined by Patrick Naughton, co-author of the HotJava browser and current VP of technology at Starwave Corp.

**December 5, 1990**

Naughton turns down offer to join NeXT, starts work on what becomes the "Green" consumer environment project at Sun.

**January 15, 1991**

"Stealth Project" (as named by Scott McNealy) brainstorming meeting in Aspen with Bill Joy, Andy Bechtolsheim, Wayne Rosing, Mike Sheridan, James Gosling and Patrick Naughton.

**February 1, 1991**

Gosling, Sheridan, and Naughton begin work in earnest. Naughton focuses on "Aspen" graphics system, Gosling on programming language ideas, Sheridan on business development.

**April 8, 1991**

Move off-site to 2180 Sand Hill Road; break direct LAN connection (and most communication) to Sun; project settles upon the name "Green."

April 15, 1991

Ed Frank (a SPARCstation 10 architect), Craig Forrest (SS10 chip designer), and Chris Warth (NeWS software developer) join Green.

May 1991

Ed Frank coins \*7 (pronounced "star-seven") as the hardware prototype's name. (\*7 was the code you pressed in the Sand Hill office to answer any ringing phone from any telephone.)

June 1991

Gosling starts working on the "Oak" interpreter, which, several years later (following a trademark search), is renamed "Java."

August 1, 1991

Oak and Aspen merge and run their first real program.

August 19, 1991

Green team demonstrates basic user interface ideas and graphics system to Sun co-founders Scott McNealy and Bill Joy.

October 17, 1991

Sheridan and Naughton coin "1st Person" as the name for the team's design philosophy, and eventually the name of the company.

November 17, 1991

Green reconnects to Sun main network with 56K line.

March 1, 1992

Jonathan Payne, who later co-wrote HotJava, joins Green.

Summer 1992

Massive amounts of hacking on Oak, the Green OS, the UI, the Star7 hardware, and related components.

September 4, 1992

Star 7 device is complete, and demonstrated to Joy and McNealy.

October 1, 1992

Wayne Rosing joins from SunLabs (which had formed in July 1990) and assumes management of the team.

November 1, 1992

FirstPerson is incorporated.

January 15, 1992

The team moves to 100 Hamilton Avenue in Palo Alto. Poetically, this is the former DEC Western Research Lab, where the original "Hamilton Group", (aka OSF) was founded.

March 15, 1993

FirstPerson focuses on interactive television after learning about Time Warner's RFP for its interactive cable TV trial in Orlando, FL.

April, 1993

NCSA Mosaic 1.0, the first graphical browser for the Internet, is released.

June 14, 1993

Time Warner goes with SGI for its interactive cable TV trial, despite acknowledged superiority of Sun technology and assurances in mid-April that Sun won the deal.

Summer, 1993

Naughton flies 300,000 miles selling Oak to anyone involved in consumer electronics and interactive television; meanwhile, the rate at which people are gaining access to the Internet reaches breakneck speed.

August, 1993

After months of promising negotiations with 3DO to provide set-top box OS, 3DO president Trip Hawkins offers to buy technology outright. McNealy refuses, and deal falls through.

September, 1993

Arthur Van Hoff joins team, originally to do application development environment aimed at interactive television; ends up doing mostly language design.

December 7, 1993

A high-level operations review of FirstPerson finds the group has no real partners or marketing strategy and a looming launch date.

February 8, 1994

FirstPerson's planned public launch at Richard Saul Wurman's Technology, Entertainment and Design Conference, "TED5," is canceled.

February 17, 1994

Alternative FirstPerson business plan for doing CD-ROM/online multimedia platform based on Oak presented to Sun executives to very mixed reviews.

April 25, 1994

Sun Interactive created, half of FirstPerson employees leave to join it.

June, 1994

"Liveoak" project started. Designed by Bill Joy to use Oak for a big small operating system project.

July, 1994

Naughton reduces the "Liveoak" project's scope to simply retargeting Oak at the Internet after writing a throwaway implementation of a Web browser in a long weekend hack.

September 16, 1994

Payne and Naughton start writing "WebRunner," a Mosaic-like browser later renamed "HotJava"

September 29, 1994

HotJava prototype is first demonstrated to Sun executives.

October 11, 1994

Naughton leaves for Starwave.

Autumn, 1994

Van Hoff implements Java compiler in Java. (Gosling had previously implemented it in C.)

May 23, 1995

Sun formally announces Java and HotJava at SunWorld '95.

---

Copyright 1995 Web Publishing Inc., an IDG Communications company