

The Battle over Real-Time Java

Sixto Ortiz Jr.

A battle is taking place over the use of Java technology in one of the most important technologies in the computer industry: the smart appliance.

Today, devices such as cellular phones, VCRs, telephone switches, video games, military defense surveillance systems, cars, network routers and switches, and even household appliances are sporting an array of embedded systems that permit such smart capabilities as the real-time monitoring of mechanical and electronic processes, automation control, and the connection of devices via a network or the Internet.

Craig Roth, an analyst with the Meta Group, a market research and consulting firm, said, "The market for embedded systems is growing at a rapid pace as consumers and corporate customers expect their portable devices and equipment to be more intelligent."

As an indication of the technology's growing popularity, Hambrecht & Quist, an investment banking firm, predicted revenue from the sale of real-time embedded operating systems and development tools will increase from \$421 million in 1997 to \$658 million this year to \$1.02 billion in 2001, as shown in Figure 1.

Developers have used a variety of languages to code real-time embedded applications, including C, C++, and Ada.

In addition, the real-time embedded



Developers want to use Java to write real-time embedded applications for multiple platforms.

system universe features many commercial and proprietary real-time operating systems and dozens of microprocessors, noted E. Douglas Jensen, a consulting scientist for the Mitre Corp., which performs systems-engineering- and information-technology-related work for the US government.

This plethora of software and hardware choices, Jensen said, leaves developers with little or no cross-platform portability. This has caused some obvious problems for the industry. For example, vendors frequently must maintain multiple versions of applications for different hardware platforms, compile for different target devices, and rewrite and retest applications whenever software is ported to a new host processor.

Developers are looking closely at Java as a way to write real-time applications

for embedded devices on various platforms. However, Java currently is not well suited for use in real-time applications.

To remedy this, Java would require extensions. Some vendors, such as NewMonics and Rockwell Collins, are using proprietary extensions for real-time Java applications. However, the industry wants to enjoy the benefit of standardization for these extensions. Two groups are pursuing this goal: the Hewlett-Packard-led J Consortium and the Real Time Java Experts Group, led by Sun Microsystems, which developed Java.

Roth said the J Consortium's formation questions Sun's stewardship of Java. He said it represents a particular challenge to Sun because embedded devices could be an important market for Java that the company would like to dominate.

Because there are two real-time Java groups, appliance and application vendors fear the process could yield competing, incompatible standards. This would hurt vendors by forcing them to either spend additional time and money supporting two standards or ignore part of the potential market by backing only one standard.

REAL-TIME JAVA

Using Java for real-time applications offers a number of advantages and disadvantages.

Advantages

Java's primary advantage for writing real-time applications is its cross-platform capabilities. Devices on different platforms are able to interpret Java bytecodes via a Java Virtual Machine implementation in the host processor.

Java is also more object-oriented than C or C++, allowing the creation of high-level real-time abstractions, according to Kelvin Nilsen, NewMonics' chief technology officer.

This would help real-time application developers by letting them generate code that is easily reusable in other applications, which would increase portability and decrease development costs, said Vicki Shipkowitz, Sun's Embedded Java product line manager.

In addition, Nilsen said, Java's dynamic nature—for example, it simplifies

Editor: Lee Garber, *Computer*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; l.garber@computer.org

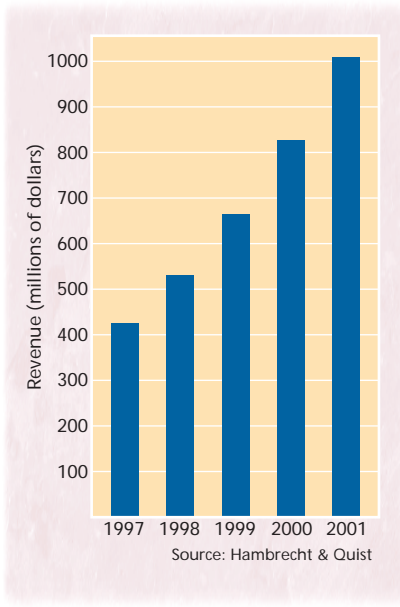


Figure 1. Hambrecht & Quist, an investment banking firm, said the real-time embedded market has been steadily growing. For example, the firm said, the market for real-time embedded operating systems and development tools has grown about 25 percent annually during the past two years and will continue to grow at the same brisk rate for the next couple of years.

the use of dynamically allocated memory objects and supports dynamic class loading—would permit real-time applications to support the on-the-fly reconfiguration necessary to handle changing workloads.

Traditional real-time systems have been designed to be static because they have had trouble meeting real-time constraints while keeping up with varying workloads. To handle mission-critical functions on a real-time basis, systems must perform predictably, always executing the same task the same way. This requires programmers to precisely specify this predictable behavior on a real-time basis, which is complicated when workloads change. However, vendors would like to develop more flexible real-time systems that can handle variable workloads.

Meanwhile, users could take advantage of Java's built-in features that help communication over networks and the Internet.

Disadvantages

According to Nilsen, Java's semantics are not specified precisely enough for programmers to clearly and predictably define and establish real-time constraints, such as prioritization, for task execution.

For example, he said, Java lacks a mechanism for handling time-outs and asynchronous events. Java-based systems also lack the ability to precisely determine a software component's behavior, such as the amount of CPU time and memory it requires for execution. This occurs largely because in Java, the automatic garbage-collection process can unpredictably make demands on the CPU and memory.

Meanwhile, Nilsen said, Java's just-in-time compiler hampers real-time performance because it translates code on the fly, which is difficult to predict.

Achieving real-time Java

IBM Senior Engineer Greg Bollella, head of the Real Time Java Experts Group, said Java extensions will have to retrofit or enhance Java's semantics to let developers precisely define real-time behavior.

Nilsen said these extensions could include additional object classes, behavioral requirements, and/or programming-language syntaxes.

Figure 2 shows how NewMonics' Perc (portable execution for reliable control) system uses proprietary extensions to achieve real-time performance. Perc uses extensions to add new classes to Java, as well as syntactic enhancements that let programmers specify real-time behavior.

J CONSORTIUM

A group of companies, including many that develop embedded real-time applications, have formed the J Consortium (<http://www.j-consortium.com>), which they originally called the Real-Time Java Working Group. The Working Group (<http://www.j-consortium.com/rtjwg.html>) is now the consortium's technical arm.

In addition to HP, the members include such companies as Aonix, Ericsson, Microsoft, Mitre, and NewMonics.

Policies

The consortium says it will conduct an

open real-time Java development process.

For example, when the group's specifications are finished, it plans to publish them on its Web site for public review.

Dave Wood, an Aonix product manager and consortium board member, said the group "seeks broad representation from users, developers, and platform and tool suppliers of all sizes on an international basis."

In fact, he said, "The J Consortium would be happy to join forces with Sun. Indeed, we have a standing offer to Sun to join."

However, Wood said, the group's by-laws are designed to prevent domination by any vendor or group of vendors. Sun's Java development process, on the other hand, gives the company too much of an advantage in the marketplace, he said.

This reflects a political split regarding Java technology. HP, along with a number of other companies, has said Sun's overall process for developing Java and related standards is not open enough and would unfairly benefit the company by giving it too much control over this important technology's future.

Shipkowitz said Sun's specifications are open, audited, and developed with review from interested parties.

She said, "Sun's role in all this is to shepherd a process for specification development that can ensure the developer community that the Java platform is secure and that it will remain compatible with a large number of operating systems."

Technical approach

The J Consortium would create a real-time Java core with functionality not found in the traditional Java core.

For example, the group says its specifications would

- offer minimal latency by limiting how long a real-time interrupt handler can take to respond to an asynchronous event, and
- let real-time Java programs give up the benefit of garbage collection to improve throughput and decrease latency and complexity.

Nilsen said the consortium also calls

for coding all aspects of real-time applications in Java. Past attempts to provide real-time Java functionality have implemented only parts of an application in Java, leaving the rest in another language. However, Nilsen said, developers then had to maintain two sets of code in two languages, which added complexity to the process.

Meanwhile, he said, the consortium supports the native compilation, rather than the interpretation, of code to achieve increased performance. However, because platform changes would require recompilation, this would sacrifice some portability. Nonetheless, Nilsen said, Java source code is more portable than C or C++ source code, and Java bytecode is more portable than Unix or Windows objects.

In the near future, Wood said, the J Consortium expects to release a draft specification, an initial set of conformance tests, and one or more implementations of real-time Java technology that complies with the standard.

The group recently proposed to work with the US National Committee for Information Technology Standardization—which helps develop domestic information-technology standards and which provides US input on international IT standards—on its real-time Java specifications. The NCITS would have managed and overseen the consortium's activities.

However, the NCITS rejected the proposal. Opponents said they were concerned about Sun's intellectual-property rights and noted that Sun has expressed willingness to make its Java development process more open.

REAL TIME JAVA EXPERTS GROUP

The Real Time Java Experts Group (<http://www.rtj.org>) is a consortium that includes such organizations as Carnegie Mellon University, IBM, Lucent, Motorola, and Sun.

Aonix, Mitre, and Rockwell Collins are on both the J Consortium and the Experts Group. Wood said, "Aonix feels that by participating in both groups, we dramatically increase the likelihood that resulting specifications will meet the rigorous needs of our key customers."

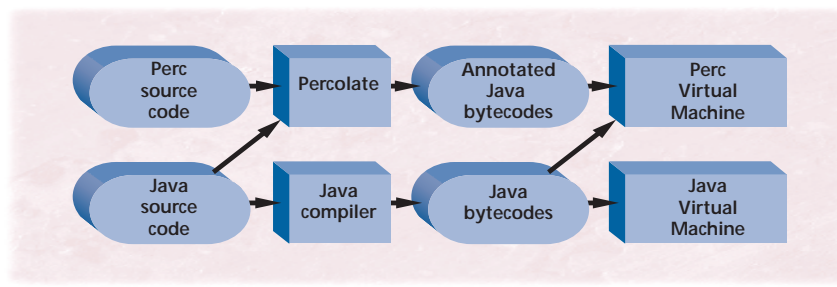


Figure 2. *NewMonics has proposed a real-time Java dialect, called Perc, that achieves real-time performance via proprietary extensions. The Percolate compiler translates Perc source code into Java bytecodes, which are sent to the Perc Virtual Machine and then on to the host device's hardware platform for execution. Percolate performs a critical role by, for example, recognizing Perc's real-time extensions and helping with the time analysis of real-time source code. The Perc system also runs standard Java source code or bytecodes, but not on a real-time basis.*

According to Bollella, the Real Time Java Experts Group wants to develop a specification that addresses the requirements of a wide range of real-time systems and programming styles.

Sun and Bollella both declined to comment about the technical approach the Experts Group will take.

Bollella said a proposed specification will be ready for participant review by August and for public review by the end of this year. He said a reference specification will be released shortly thereafter.

Clearly, the stakes are high for real-time embedded-system developers who are eager to exploit Java's cross-platform capabilities. However, the competition between the J Consortium and the Experts Group is raising concerns about two incompatible standards emerging from the fray.

According to the Meta Group's Roth, "In the worst case, vendors will have to code to two different APIs." However, he said, it is likely that the two specifications would share many elements. Nonetheless, he said, they would probably also have subtle differences that would force programmers to carefully develop and extensively test their applications to achieve cross-platform portability.

Because the NCITS decided not to work with the J Consortium, the Experts Group will probably have more control over the development of a real-time em-

bedded standard, said Matt Belkin, Hambrecht & Quist's embedded-systems analyst.

However, Belkin said, the most important factor in the marketplace may be which technology users want.

Many users want the J Consortium and the Experts Group to join forces to develop a unified specification.

Sun has met with the J Consortium to discuss its standards-development process for Java.

Bollella said, "The opportunity for sales of real-time devices is so huge that it dwarfs PC sales, so we are hopeful that the efforts will eventually be combined, and we will have one specification for customers."

Aonix's Wood said the J Consortium could endorse the specifications developed by the Experts Group if they meet the needs of real-time and embedded-system users. However, he said, the J Consortium can't tell whether that will be the case because the Experts Group is not opening its work to public scrutiny.

"We cannot wait for Sun to divulge those specifications, or we will lose precious time in this fast-moving industry. If the Sun process were truly open and neutral, there would be no problems," he said. "However, it is our hope that the two groups will reach compatible conclusions." ♦

Sixto Ortiz Jr. is a freelance technology writer based in Clute, Texas. Contact him at sortiz@computron.net.