Real-Time Java for Flight Applications

An Update

Background: The Real-Time Specification for Java (RTSJ)

The RTSJ is a specification for supporting real-time execution in the Java programming language. The specification has been shaped by several guiding principles, particularly: predictable execution as the first priority in all tradeoffs, no syntactic extensions to Java, and backward compatibility. The RTSJ defines APIs in seven major categories: threads, scheduling, memory management, synchronization, time, timers, and asynchrony. The specification achieved a status of "Final Release" in January 2002 as JSR 1 of the Java Community Process. The specification is available at http://rtj.org. A commercial implementation of the RTSJ became available from TimeSys Corporation in March 2003, in combination with a low-latency version of Linux (see http://www.timesys.com).

Objectives & Approach

JPL has initiated a task to assess the risks in transitioning to a new software platform for real-time embedded systems based on real-time Java and real-time Linux. This task is examining five risk areas—real-time response, performance, programming model, multi-language development, and maturity—as identified in the 1999 Nichols report "Using Java for Flight Implementations". Part of the approach for this assessment involves porting some existing rover software from C++/VxWorks to RTSJ/Linux, and then conducting a side-by-side comparison. Another part of the approach involves a substantial collection of metrics, including interrupt response latency, timing jitter, floating-point performance, CPU and memory utilization, and cache hit ratio. This task is leveraging the expertise of Dr. Greg Bollella, lead for the Real-Time Java Expert Group, and Dr. James Gosling, creator of the Java language.

Current Status

Most of the work to date has focused on porting rover software for controlling driving and steering motors, as well cameras, and testing that software on JPL's Rocky 7 rover. To date, the team has successfully controlled the motors and cameras individually and has begun testing the rover position & heading controller for coordinated 6-wheel control. In getting to this point, the team has installed TimeSys Linux and the JTime virtual machine on an embedded PowerPC 750, developed Linux drivers for a digital I/O board and frame grabber, ported framework software from C++ to RTSJ, and ported motor and camera control software from C++ to RTSJ. Approximately 13K lines of code have been ported, and some of that has required design changes due to language differences between C++ and Java. As a whole, this effort has helped us learn how to use the facilities of RTSJ and understand the nuances of working with scoped memory and immortal memory.

Plans

The Rocky 7 rover driving and imaging scenario is leading up to a demonstration at the JavaOne™ conference at James Gosling's keynote talk on June 11, with Rocky 7 driving on stage. Next, the team will focus on metrics (real-time response, throughput, etc) and will also reexamine some design choices in light of what has been learned. The team will document its first-year findings by September 2003. Follow-on work in FY 2004 will expand the metrics effort with measurements on three platforms—RTSJ/Linux, C++/Linux, and C++/VxWorks—enabling us to separate language differences (RTSJ vs. C++) and RTOS differences (Linux vs. VxWorks).

Contact Information

For more information on this task, contact Daniel Dvorak (Daniel.Dvorak@jpl.nasa.gov, 818-393-1986) or Kenny Meyer (Kenneth.Meyer@jpl.nasa.gov, 818-393-4871) or Kirk Reinholtz (Kirk.Reinholtz@jpl.nasa.gov, 818-354-6419).