A Survey and Comparison of Concurrency and Real-Time Programming Support in Java, Ada, and Posix

Benjamin Brosgol

Unlike sequential programming, the debate over whose programming language support was largely settled in the Structured Programming revolution of the early 1970s, concurrency and especially real-time programming remain subjects that elicit considerable controversy. Different languages have taken different approaches; some languages ignore the matter, believing that these topics are more in the realm of an operating system or real-time kernel.

This tutorial identifies the issues that underlie concurrency and real-time programming and describes how they are addressed by Java, Ada, and Posix. It will cover thread/task lifetime properties (creation, termination), mutual exclusion, coordination / communication, asynchrony, dealing with time, and scheduling, with a focus on real-time requirements such as management of priority inversion. Some common examples (e.g. bounded buffers, periodic activities) will be used to illustrate the different approaches, which will be compared with respect to software engineering support (readability, reliability), predictability and performance. The Java approach will be presented in terms of the two current proposed real-time extensions: the Real-Time Specification for Java (from the Sun-sponsored Real-Time for Java Expert Group) and the Real-Time Core Extensions (from the J-Consortium). The main emphasis will be on uniprocessor systems.

Topics include:

- Mechanisms for mutual exclusion
- Priority Inheritance and Highest Lockers protocols
- Asynchronous transfer of control
- How real-time Java deals with Garbage Collection
- Event handling
- Interactions between concurrency and exception handling
- Rationale for deprecation of most asynchronous thread communication in Java
- Functionality of Ada 95 Real-Time Annex

Benjamin Brosgol has over 25 years of experience in the computer software industry, with a focus on programming languages, software development methods, and real-time systems. He is a primary member of the Real-Time for Java Experts Group, and a reviewer of the specification produced by the J-Consortium’s RTJWG.

He has been delivering Java tutorials regularly since 1997, and he has taught real-time programming courses since 1987. He has published several technical papers on Java and is currently writing a book (with a co-author) titled Real-Time Programming in Java.

In addition to his Java activities he has had a long involvement with the Ada language effort. He participated in both the initial language design and the Ada 95 revision, and he is currently the chairman of the ACM’s Special Interest Group on Ada (SIGAda).

He is currently an independent consultant in the Boston area.