

Session 5: Ada and Other Standards

Chair: Ben Brosgol

Rapporteur: Mario Aldea

1. Session Goals

The main goals of this session were to:

- consider whether a new binding between Ada and POSIX is needed/desirable
- look at current efforts to update POSIX and "Real-Time Java" in terms of impact on or "lessons learned" for Ada

2. POSIX Ada binding

Stephen Michell presented his paper on "Interfacing Ada to Operating Systems" [1]. He stated that the POSIX Ada Binding has not been updated since 1998, even though POSIX has undergone two significant revisions and the new Ada 2005 has been approved. Taking into account those changes, the current binding is mostly correct from the functional point of view, but:

- there are some minor errors/inconsistencies
- it does not include the new functionality added to POSIX during the last few years
- some functionality included in the binding is now directly supported in Annex D of Ada 2005

Stephen also stated there are other operating systems that do not follow the POSIX standard and are important nowadays, like the Windows family or other operating systems for embedded platforms.

Instead of adapting the current POSIX Ada binding to the new situation, Stephen proposed to update it to a general interface to operating systems, implemented in a set of packages which would be children of Ada.Interfaces

2.1. Discussions

Some general points about the POSIX Ada binding philosophy were noted:

- The general idea of the binding is to provide interfaces only for the functionality not provided by the Ada language

- One of the main objectives of the binding is to allow Ada tasks and POSIX threads to interoperate, for instance by being able to share data using mutexes, or to synchronize through condition variables. Jose Ruiz mentioned that the GNAT compiler allows POSIX threads to call Ada protected objects (the run-time library registers them as foreign threads)
- The difference between the priority band concept in Ada 2005 and the way scheduling policies are defined by POSIX was identified as a potential difficulty.

The following concerns about the utilization of the binding were raised:

- The general feeling was that the binding has had very few users. Many programmers just create a small binding for the few functions they actually use in each particular application. At this point some doubts appeared about whether it is worthy to put some effort on a new binding. There was consensus on at least putting some effort to correct the minor errors/inconsistencies.

One of the minor details that should be corrected is the fact that the current POSIX/Ada binding references the old POSIX standards instead of the new ones.

It was pointed out that the current binding is a mixture between Ada and C styles and it would be desirable to develop a more Ada-like binding. Some people agreed on that but it was questioned if it is worthy for the Ada community to put such a big effort in this project.

There was a suggestion of doing a minimum change in the binding in order to mark as obsolete those services currently included in the binding that have an equivalent in the Ada 2005 standard.

2.2. Conclusions

The main conclusions of this part of the session were:

- There was an agreement that future editions of this workshop could help in the technical decisions related with the new POSIX/Ada binding, but that the workshop itself

should not take over the responsibility of the revision of the binding

- The workshop participants agreed that, at least, a minimum update of the binding is desirable in order to correct minor errors/inconsistencies and wrong references to POSIX standards.

3. Real-Time Java

Ben Brosgol made a presentation about the history and current status of Real-time Java, including a summary of the main RTSJ (Real-Time Specification for Java) features and a list of the existing implementations.

Nowadays the main effort in the RT Java area is the Safety-Critical RT Java, officially named "Java Specification Request 302" (JSR-302). Two members of the IRTAW workshop are included in the "Expert Group" in charge of developing this project: Andy Wellings and Ben Brosgol. JSR-302 progress is slow since there are two competing proposals: HIJA (High-Integrity Java Applications) and Aonix with its "Scalable Real-Time Java proposal".

Ben pointed out several aspects of RT Java that could be interesting for Ada:

- Real-time programming paradigms
 - Periodic, aperiodic, sporadic threads
- General capabilities
 - Annotations to guide static analysis
- Interesting functionality
 - Asynchronous Event Handling:
 - Application control over "fire count" to deal with bursts
 - Pass data when an event is fired
 - General mix of Priority Inheritance and Priority Ceiling Emulation
 - Ability to awaken a suspended thread via synchronous exception
- "Exotasks" (from IBM, based on Giotto)
 - Not threads but periodic code that is dispatched based on a scheduler
 - Dedicated heap that is garbage collected
 - No shared memory; communication via deep copy over typed ports
 - Eclipse-based environment allowing annotated timing constraints

3.1. Discussions

The subsequent discussion were centered on the following issues:

- Including garbage collection in Ada: it could be interesting for some applications that use unchecked dealloca-

tion. It is noticed that nothing in Ada prevents from using a garbage collector, so maybe this is not actually an issue.

- Different kinds of "physical" memory: the ability to place data in a specific kind of memory (fast access memory, flash memory, ...). This service is provided by RTSJ and POSIX but not by Ada.
- Mix of Priority Inheritance and Priority Ceiling Emulation: can be a source of problems due to the nested locks, but it was pointed out that priority inheritance is an interesting functionality for large systems built from "components" developed independently.
- Awaken a suspended thread: a possible alternative in Ada consists of blocking the task in a protected entry.

3.2. Conclusions

The main conclusions of this part of the session were:

- The Real-time Ada community should keep an eye on the current efforts on RT Java and check if its advances could be interesting for future revisions of the Ada standard
- The workshop encourages the Real-Time Ada community to continue research on topics like garbage collection, different kinds of memory, or priority inheritance in Ada.

References

- [1] Stephen Michell, S.: *Interfacing Ada to Operating Systems*. Ada-Letters (this issue).