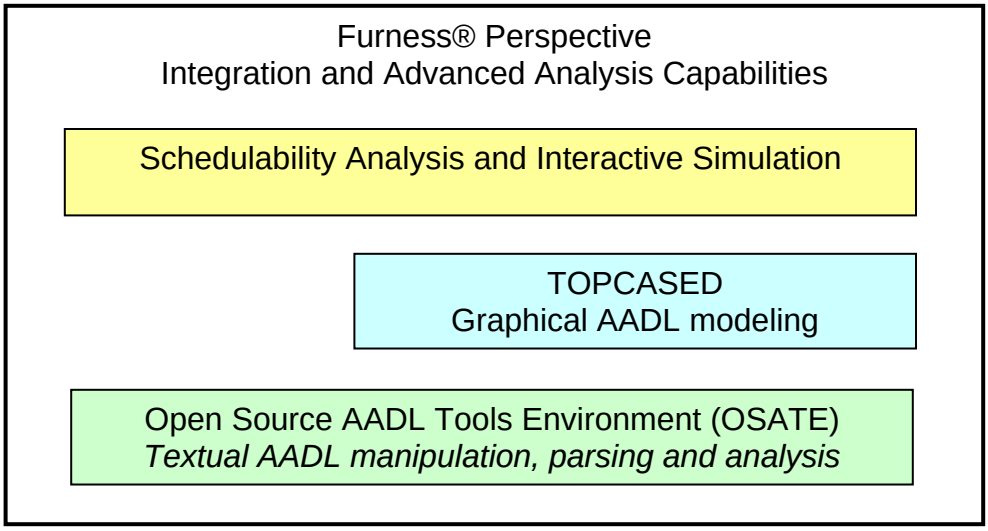
User Guide

Furness® Toolset v1.2

1. What it is

The Furness® Toolset collects the leading open-source AADL tools into a single, professionally supported release. The tools are integrated into the open-source Eclipse IDE and installed and updated through the Eclipse online update site feature.



The present release of the Furness Toolset includes:

- The Open Source AADL Tools Environment (OSATE) produced by the Software Engineering Institute of Carnegie Mellon University.
- The TOPCASED graphical meta-modeling framework provided by the TOPCASED Consortium with a graphical AADL profile for creating and manipulating AADL diagrams.
- The Furness Perspective, integrating features from OSATE and TOPCASED with advanced analysis capabilities provided directly by the Furness Toolset, provided by a joint venture of Fremont Associates and University of Pennsylvania researchers.

2. How to get it

Installing the Furness Toolset is a five-step process:

1. Download and install Eclipse 3.2 from the Eclipse website: http://www.eclipse.org
2. Use the Eclipse software updates feature to add TOPCASED to your Eclipse installation. The TOPCASED update site (including OSATE) is http://topcased-mm.gforge.enseeiht.fr/release/update-site/
3. Use the Eclipse software updates feature to add the Furness Toolset to your Eclipse installation. The Furness Toolset update site is http://eclipse.furnesstoolset.com

4. Download and install the VERSA tool (an analysis engine for process algebra models that lives outside the Eclipse framework) from the Furness Toolset web site downloads page at http://www.furnesstoolset.com/downloads.htm#furness. Record your install path for use in the next step.
5. Open the global Eclipse preferences page (Window, Preferences...), select the Furness Toolset preference page, and browse to the VERSA executable file installed above.

3. What it does

The Furness Perspective

The Furness Perspective organizes the views, action and editors of OSATE, TOPCASED and the Furness Toolset into a simplified interface that mimics existing Eclipse perspectives.

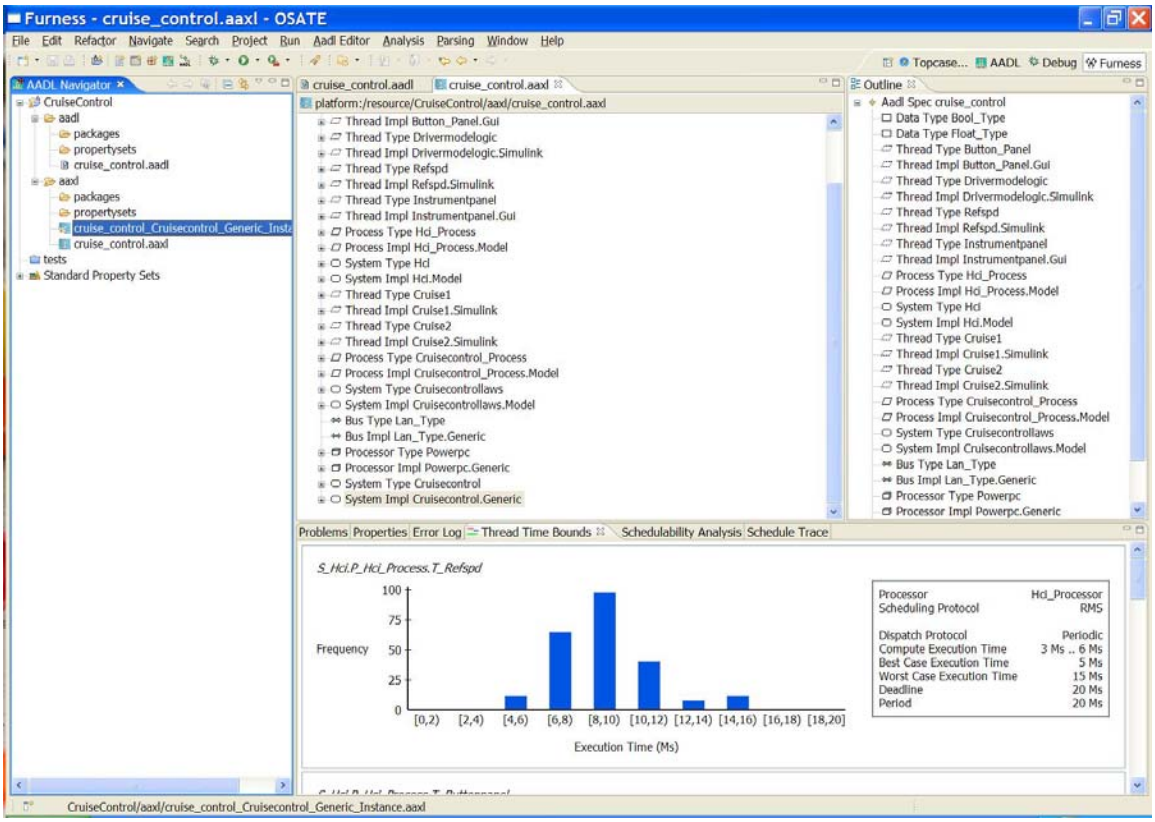


Figure 1: Furness Perspective Overview

Figure 1 depicts the general layout of the Furness Perspective. The AADL Navigator, outline view and editors appear in their typical Eclipse IDE locations. Operations/actions on AADL models are organized into two toolbar entries—Analysis (for model analysis task) and Parsing (for low-level manipulation of XML files and markers). Key actions also have toolbar buttons, including buttons to create new projects, files, etc., and projects to create simulation launches using the standard Eclipse debug launch button.

AADL System Instance Schedulability Analysis

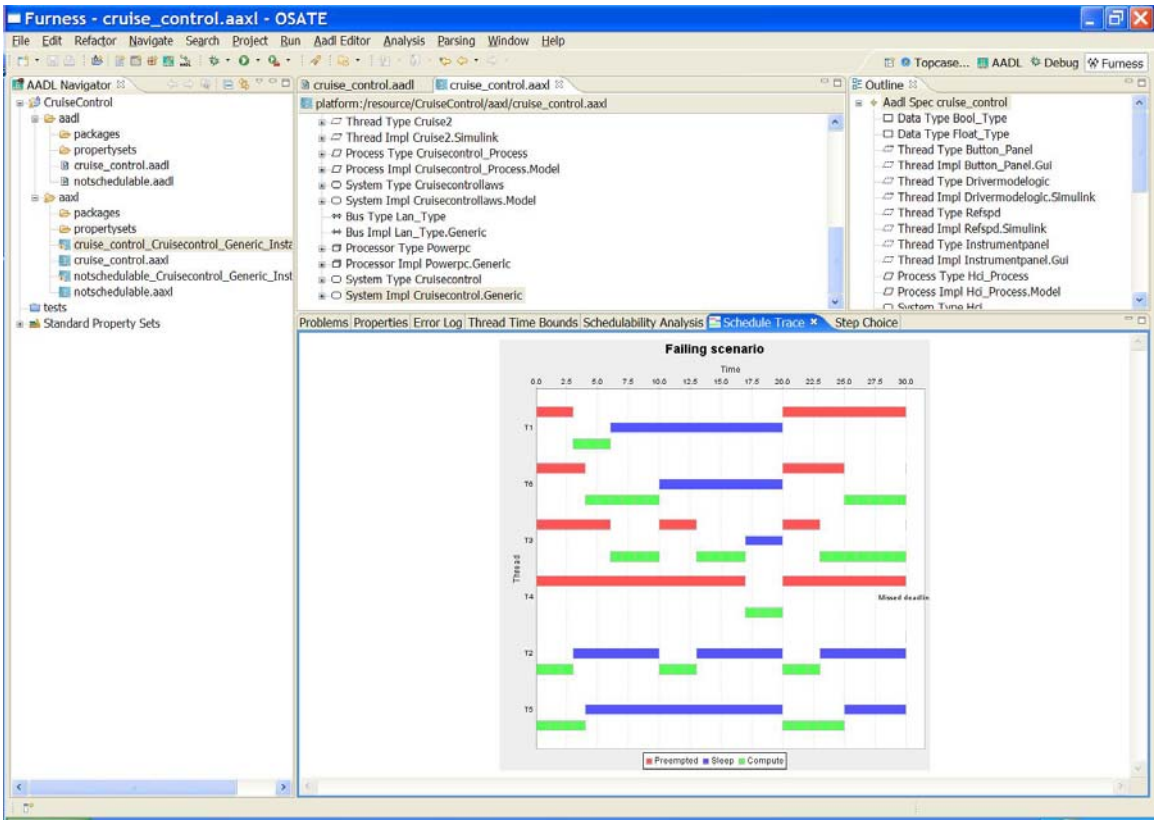


Figure 2: Schedulability Analysis Overview—System Not Schedulable

The schedulability analysis feature will analyze a subset of AADL system instance models to determine whether the thread scheduling constraints are satisfiable. If a system instance model is not schedulable, a failing trace will be displayed in the form of a timed system trace as shown in Figure 2. If a system instance model is schedulable, an analysis of best-case and worst-case response time can be viewed using the “Thread Time Bounds” view as shown in Figure 3.

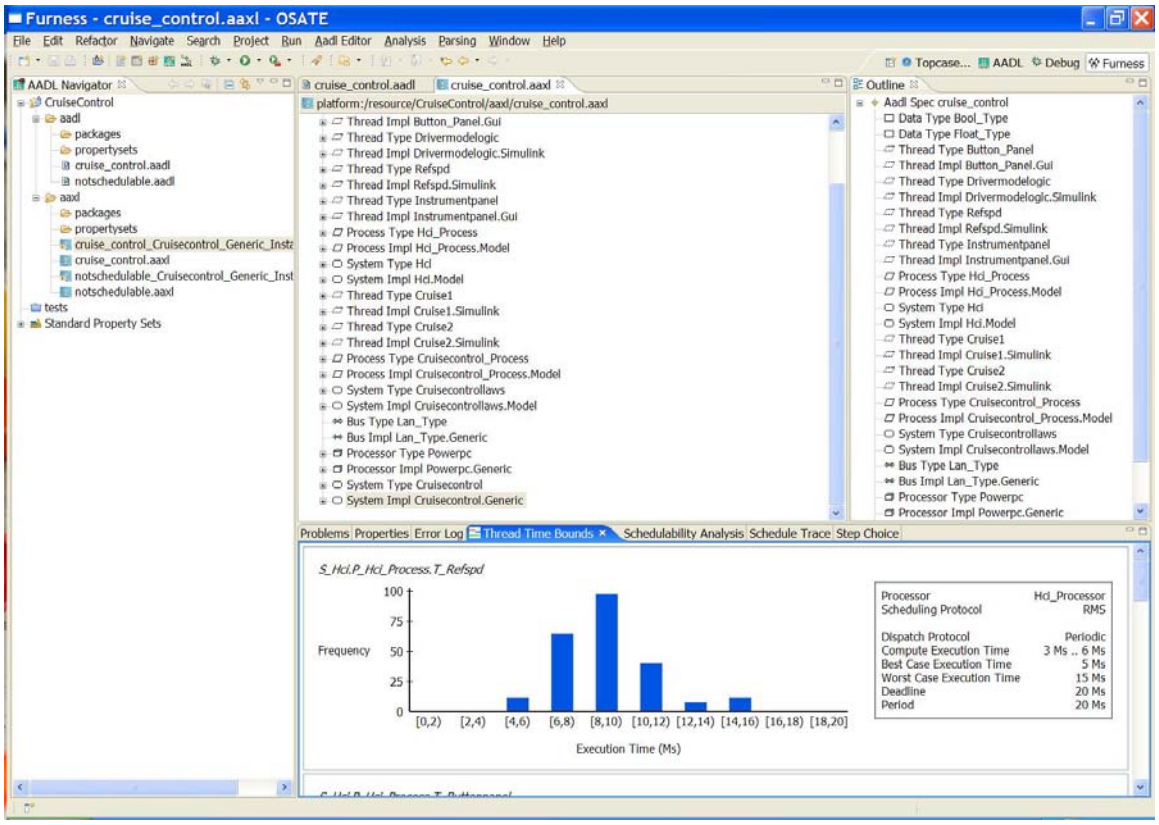


Figure 3: Schedulability Analysis Overview—System Schedulable

AADL System Instance Simulation/Debug

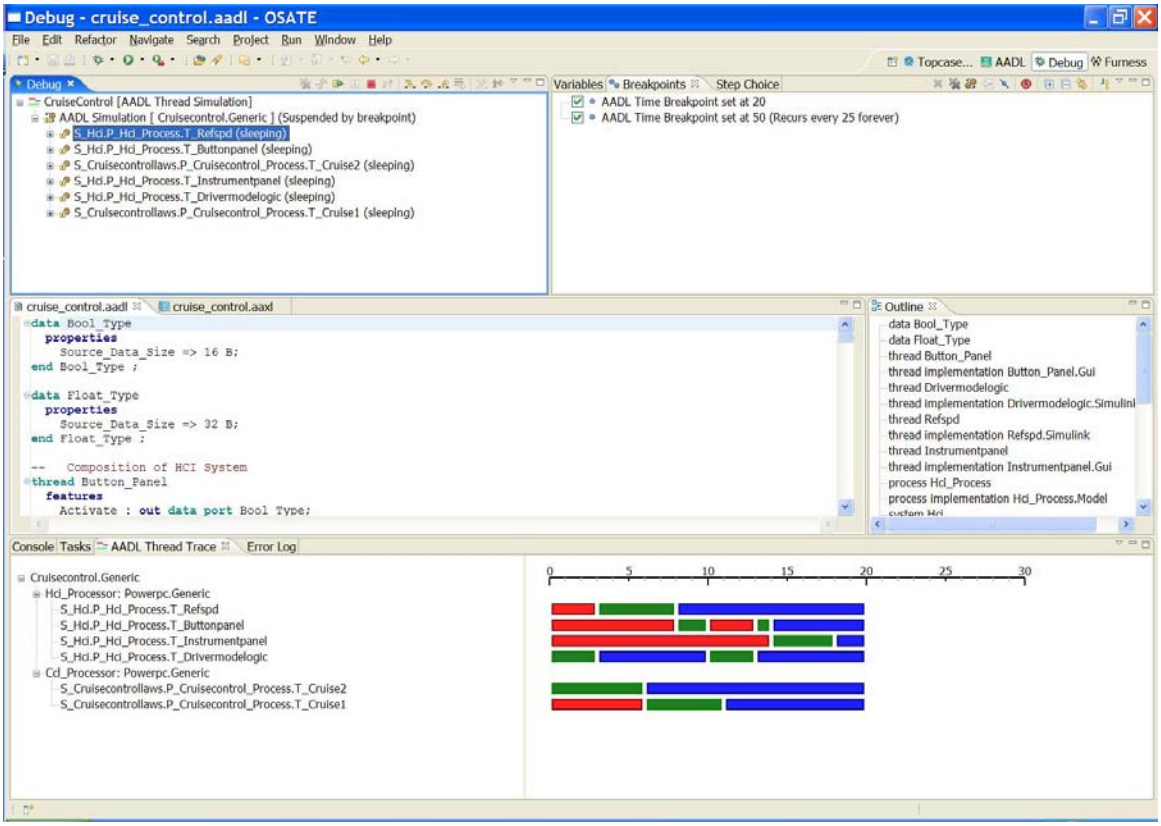


Figure 4: Simulation/Debug View—Breakpoints and Trace

The Simulation feature allows the thread behavior of synchronous AADL models to be analyzed interactively using the standard Eclipse debug perspective. Figure 4 shows the simulation view active for a two-processor, six thread system undergoing simulation. The simulation has advanced to time t=20 with the various thread activities indicated in the trace graph in the lower right hand corner. Users interact with the simulation using the standard Eclipse debugger controls, as shown in the debug view in the upper-left pane.

The AADL debugger includes the ability to set breakpoints at specific instants in time, or on recurring intervals, as shown in the upper right pane of Figure 4. The standard Eclipse debug view (upper left pane) shows the current state of all threads, the reason execution is suspended, and all buttons related to execution of suspended threads are enabled to allow single stepping or continuous running.

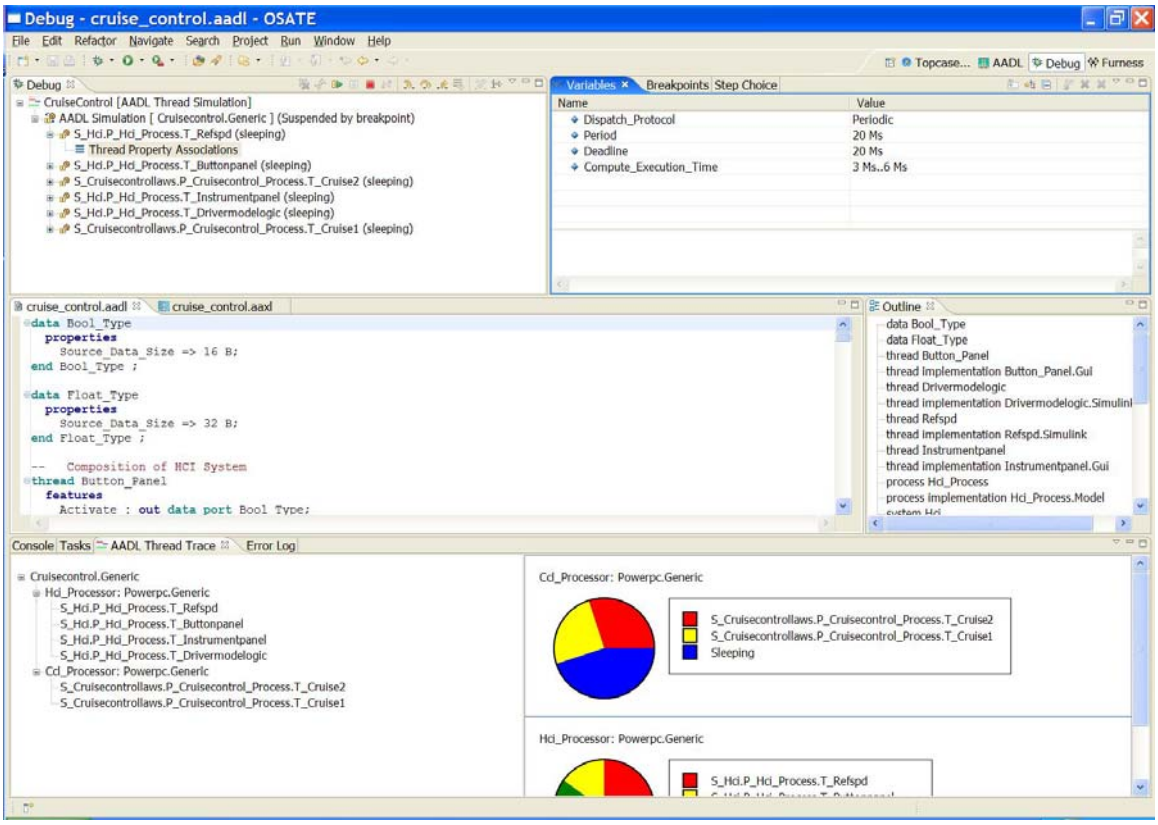


Figure 5:Simulation/Debug View—Thread Property Associations and CPU Utilization

Figure 5 illustrates two additional views available for simulations. The debug view (upper left pane) shows the property associations of a thread selected, and the Eclipse Variables view (upper right pane) lists the property associations for the selected thread. In the lower right pane the utilization of each system CPU is shown in a pie-graph form, broken down by thread active and sleeping states.

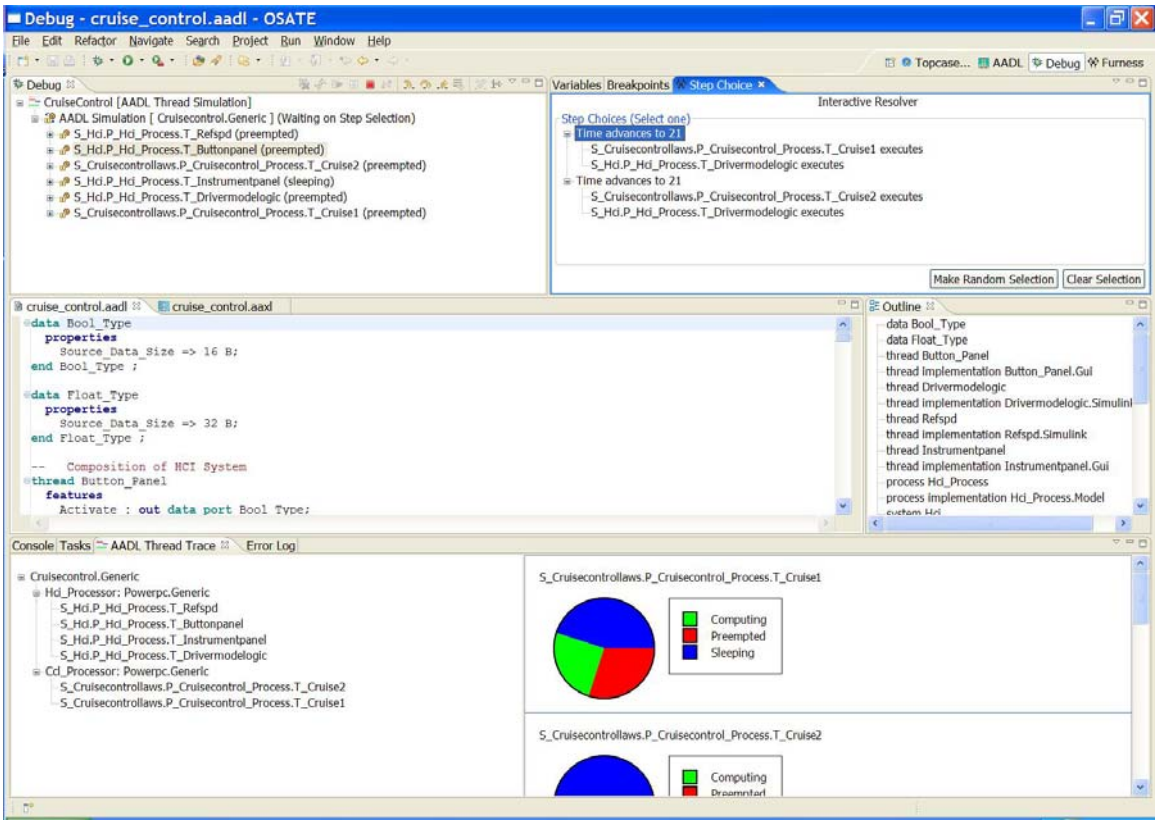


Figure 6: Simulation/Debug View—Interactive Non-Determinism and Thread States

Finally, Figure 6 illustrates two additional views. In the upper-right pane is the user interface to allow selection of specific transitions where non-deterministic choices arise in the modeled system. The lower-right pane illustrates thread states on a per-thread basis, broken down by proportion of time spent in the computing, preempted and sleeping states.

4. How to make it do that

The Furness Perspective

From the Eclipse menu bar, Window -> Open Perspective -> Other... and select “Furness” from the list.

AADL System Instance Schedulability Analysis

Create a system instance to analyze (after foo.aadl is compiled into foo.aaxl, open foo.aaxl by double-clicking; right-click the system instance you want to analze and from the pop-up context menu select OSATE -> Instantiate System).

Select the system instance model to analyze in the AADL Navigator. From the Eclipse menu bar select Analysis -> Schedulability ->

Schedulability Analysis. The basic result (schedulable or not schedulable) will be displayed in a dialog box. The more advanced views shown above are available from the Eclipse menu using the standard Window -> Show View feature.

AADL System Instance Simulation/Debug

Create a system instance to simulate (see directions above).

Select Run -> Debug... from the Eclipse menu bar (or use the "Debug..." toolbar button). Create a launch of the "AADL Thread Simulation" type and enter appropriate parameters.

Once the simulation is launched switch to the Debug perspective and manipulate the model using the actions on the debug view. Advanced views shown above are available from the Eclipse menu using the standard Window -> Show View feature.

5. "Features"

Notable known problems (*i.e.,* features) as of this writing are as follows:

- Schedulability analysis requires more than the default amount of Java heap space

If you haven't done so already you should modify the Eclipse start-up to override the defaults with values appropriate to your system. I use `-vmargs -Xms256M -Xmx1024M` (with 2 Gig of RAM installed under Windows XP).

- State space explosion

The model used for schedulability analysis and simulation is a straightforward labeled-transition system of the thread automata. If large parameter values are used, or combinations of parameters that yield large least-common-multiples are used, the construction of the state space may exhaust system memory and/or user patience.

We are developing new modeling techniques that should mitigate this problem in the next release of the Furness Toolset. If you run into a specific problem with a model of interest to you send it to us to analyze.

- Long-running simulations

There is a flow-control problem between the running simulation and the GUI. If you let a simulation run for a long time it will far outpace the GUI, exhausting Java VM resources and producing a lot of strange error messages. Set a

breakpoint at a time value beyond your area of interest in order to prevent the simulation running away from you.

We are addressing this problem as a bug and will produce a fix in the next release. As a work-around, the current release automatically sets a default breakpoint at t=500.

- "Not currently available" OSATE analysis actions

A bug in OSATE (SEI-OSATE Bugzilla #566) prohibits re-export of certain actions from the OSATE analysis set. If you see this error message when you try to use one of the OSATE actions, fetch the updated MANIFEST.MF file from the Furness Toolset downloads page

http://www.furnesstoolset.com/downloads.htm#furness

and install it in your Eclipse plug-ins directory as instructed.

6. Support

Support for the Furness Toolset is provided to all users on a resources-permitting basis. Licenses users receive support based on a service level agreement outlined at http://www.furnesstoolset.com/commerce.htm.

7. Acknowledgements

The Furness Toolset is created by Fremont Associates, LLC, and researchers at the University of Pennsylvania with support from licensed users and the Air Force Office of Scientific Research.

More Information:

http://www.furnesstoolset.com/
http://www.fremontassociates.com/
http://www.cis.upenn.edu/~rtg/index.php3

The OSATE features and plug-ins are created and supported by the Software Engineering Institute at Carnegie Mellon University.

More Information:

http://www.aadl.info

The TOPCASED Toolset are created and supported by the TOPCASED Consortium.

More Information:

http://www.topcased.org/