

REPORT DOCUMENTATION PAGE

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 29-12-2008		2. REPORT TYPE STTR--Final Technical Report		3. DATES COVERED (From - To) 30-09-2005-29-09-2008	
4. TITLE AND SUBTITLE Simulation and Analysis Toolset for an Industry Standard Embedded Systems Specification Language: Final Technical Report				5a. CONTRACT NUMBER FA9550-05-C-0187	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Duncan Clarke Fremont Associates, LLC; Camden, SC Oleg Sokolsky University of Pennsylvania; Philadelphia, PA				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Fremont Associates, LLC 813 Market Street Camden, SC 29020				B. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research 875 North Randolph Street Arlington, VA 22203-1768				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Distribution Statement A. Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The Architecture Analysis and Design Language (AADL) is an SAE standard language for describing the software and hardware architecture of performance-critical real-time systems. In addition, the AADL standard allows the definition of annexes, i.e., formal extensions to the standard language to enhance the design specifications of hardware or software components. Our work has leveraged the AADL language and tool development efforts to create a new toolset that incorporates simulation and analysis technologies for embedded real-time systems developed within the Charon and ACSR/VERSA projects at the University of Pennsylvania. Our integration of AADL with Charon and VERSA has extended the capabilities of AADL to allow analysis and simulation at the architecture level, detailed analysis at the module level, and provided support for implementation. This document constitutes the final technical report for our Phase II A FOSR STTR (FY 2004, Topic 23, "Modeling Languages and Analysis Tools for Complex Distributed").					
15. SUBJECT TERMS					
<h1>20091218091</h1>					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	OF PAGES	Duncan Clarke
a. REPORT	b. ABSTRACT	c. THIS PAGE			
U	U	U	SAR	15	(803) 432-8272

Final Technical Report—December 2008

Simulation and Analysis Toolset for an Industry Standard Embedded Systems Specification Language

Contract FA9550-05-C-0187

STTR Phase II Topic AF04-T023: Modeling Languages and Analysis Tools for Complex Distributed Systems

Program Manager

Robert Herklotz
Air Force Office of Scientific Research

Principal Investigator

Duncan Clarke
Fremont Associates, LLC
813 Market Street
Camden, SC 29020
(803) 432-8272
dclarke@fremontassociates.com

Academic Partner

Oleg Sokolsky
Department of Computer and Information Science
University of Pennsylvania
3330 Walnut Street
Philadelphia, PA 19104
(215) 898-4448
sokolsky@saul.cis.upenn.edu

2. Introduction

The Architecture Analysis and Design Language (AADL) is an SAE standard language for describing the software and hardware architecture of performance-critical real-time systems. In addition, the AADL standard allows the definition of annexes, *i.e.*, formal extensions to the standard language to enhance the design specifications of hardware or software components.

The goal of this project was to leverage the AADL language and tool development efforts to create a new toolset incorporating simulation and analysis technologies for embedded real-time systems developed within the Charon and ACSR/VERSA projects at the University of Pennsylvania. Our integration of AADL with Charon and VERSA extended the capabilities of AADL to allow analysis and simulation at the architecture level, detailed analysis at the module level, and support for implementation.

This toolset is marketed for use by DoD and private-sector embedded systems developers as part of an architecture-driven development methodology for safety-critical embedded systems.

3. Objectives

Our effort was focused on (1) creating an end-to-end systems development methodology for embedded real-time systems, (2) developing commercial quality tools to support this methodology based on the OSATE plug-in for Eclipse, and the Charon and ACSR formalisms; and (3) preliminary marketing of professional services based on the methodology and tools so developed.

The technical objectives for Phase 2 were as follows:

Develop tools to support a methodology that enumerates steps for the design, development and implementation of distributed real-time systems. Emphasize verification and validation activities that are unique to safety-critical embedded systems. Manage abstraction/complexity so that models can be analyzed effectively.

Front-End

- Develop quality assurance processes to insure the marketability of OSATE tools as the core of our tools offering. Though the OSATE development is outside our scope, their position as the front-end for our analysis tools will impact the perceived quality of our effort. Therefore we intend to implement a quality control process to verify conformance of front-end tools to the AADL standard, and evaluate overall quality.
- Develop a unified look-and-feel for all tools that emphasizes their role in support of the methodology. The Eclipse workbench presents a large

number of controls and sub-windows to the user. It is necessary to plan from the outset to insure that our tools provide a useful, usable interface to the user.

Analysis

- Develop a Charon plug-in for Eclipse allowing direct manipulation of Charon objects and access to Charon analysis back-end. The prototype tool development effort leveraged existing stand-alone tools to perform analysis of AADL specifications. We intend to invest effort to properly integrate the Charon and VERSA tools into the Eclipse framework.
- Develop an ACSR plug-in for Eclipse allowing direct manipulation of ACSR objects and access to ACSR analysis back-end.
- Complete the mapping of all AADL concepts to ACSR language elements, building on the prototype effort, which focused on proof-of-concept.
- Define and implement a translation algorithm from AADL concepts to ACSR language elements that foresees the need to translate analysis results back to AADL concepts.
- Exploit ACSR analysis tools to analyze and report on AADL specifications.
- Exploit Charon analysis tools to analyze and report on AADL specifications incorporating continuous behaviors described using Charon.

Simulation

- Develop simulation tools to demonstrate the dynamic behavior of AADL models. We plan to exploit the AADL to ACSR translation and existing tools for interactive execution of ACSR processes.
- Integrate existing simulation tools for Charon to allow simulation of models incorporating continuous behaviors.

Implementation Support

- Develop code generation framework for AADL specifications. Care will be taken to insure that the code generation effort can be both practical (*i.e.*, targeted toward an existing real-time programming language) and extensible (*i.e.*, easily retargeted toward other existing or future technologies).
- Exploit existing Charon code generation modules to incorporate code for continuous behaviors. Within the Eclipse framework, integrate custom

AADL code generation and Charon code generation with commercial tools for real-time systems implementation. The system will be architected to enable third-party extension without modification of the core Furness toolset source code.

Test

- Develop test generation techniques for AADL specifications. Research will be carried out to determine appropriate coverage criteria and testing techniques for AADL models.
- Develop test generation techniques for user-specified system properties. Practical, testing-based system verification will be central to the methodology. As such, it will be important to incorporate tests as first-class entities in the tools framework.
- Integrate analysis, test generation and implementation features to provide a user-friendly verification and validation workbench within the Eclipse framework. In the spirit of the current Eclipse tools development effort, our system will be architected to enable third-party extension without modification of the core Furness toolset source code.

4. Status of Effort

The project team developed and distributed the Furness Toolset™, a set of plugins for the Eclipse workbench that extends the functionality of the Open Source AADL Tool Environment (OSATE) created at the Software Engineering Institute. The Furness Toolset is freely available from a dedicated web site at <http://www.furnesstoolset.com>.

Version 1.6 of the Furness Toolset, distributed 9 July 2007, includes many of the features outlined in the original work plan for this contract, including:

- 1) Quality assurance processes to insure quality of the combined OSATE and Furness Toolset employed by end users.
- 2) A unified look-and-feel supporting task-oriented work.
- 3) The application of ACSR analysis tools to analyze and report on AADL specifications.
- 4) Extensive simulation tools to demonstrate the dynamic behavior of AADL models.

A one year, no-cost extension was requested and approved to enable the use of residual contract funds to continue the project effort at Fremont Associates and the University of Pennsylvania. The main technical objectives that we continue to pursue are:

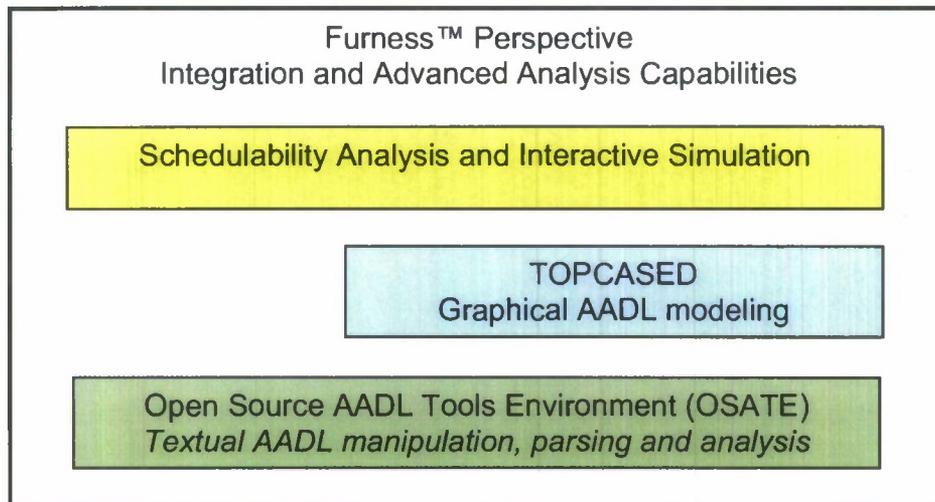
- 1) Creation of an ACSR plug-in for Eclipse allowing direct manipulation of ACSR objects and access to ACSR analysis back-end.
- 2) Creation of a code generation framework for AADL specifications targeted at the LynxOS-178 operating system for DO-178B level-A certified systems.

The Furness Toolset is supported primarily through contract FA9550-05-C-0187. A permanent revenue stream for support of the toolset was initiated during the first project year with the sale of support licenses, renewable annually.

5. Work Carried Out

Eight people were employed in planning, design, implementation, testing, distribution and support of the Furness Toolset. During the period of performance of this grant their effort produced several public releases of the software.

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.

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.

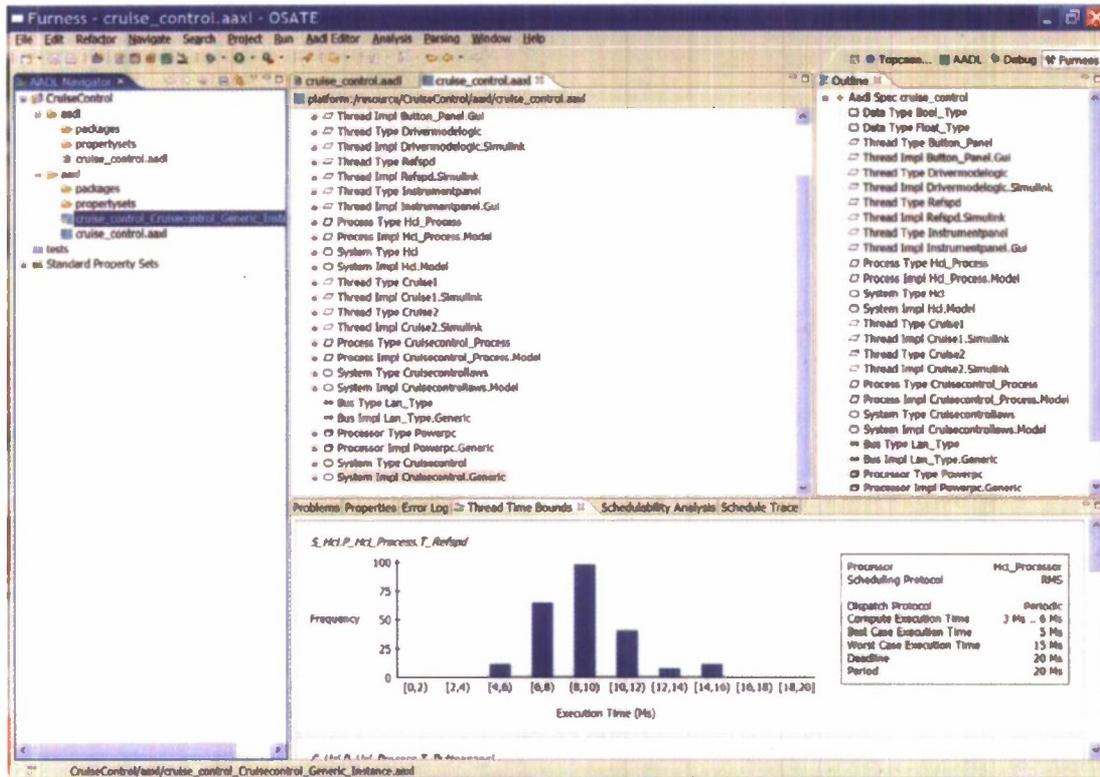


Figure 1: Furness Perspective Overview

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

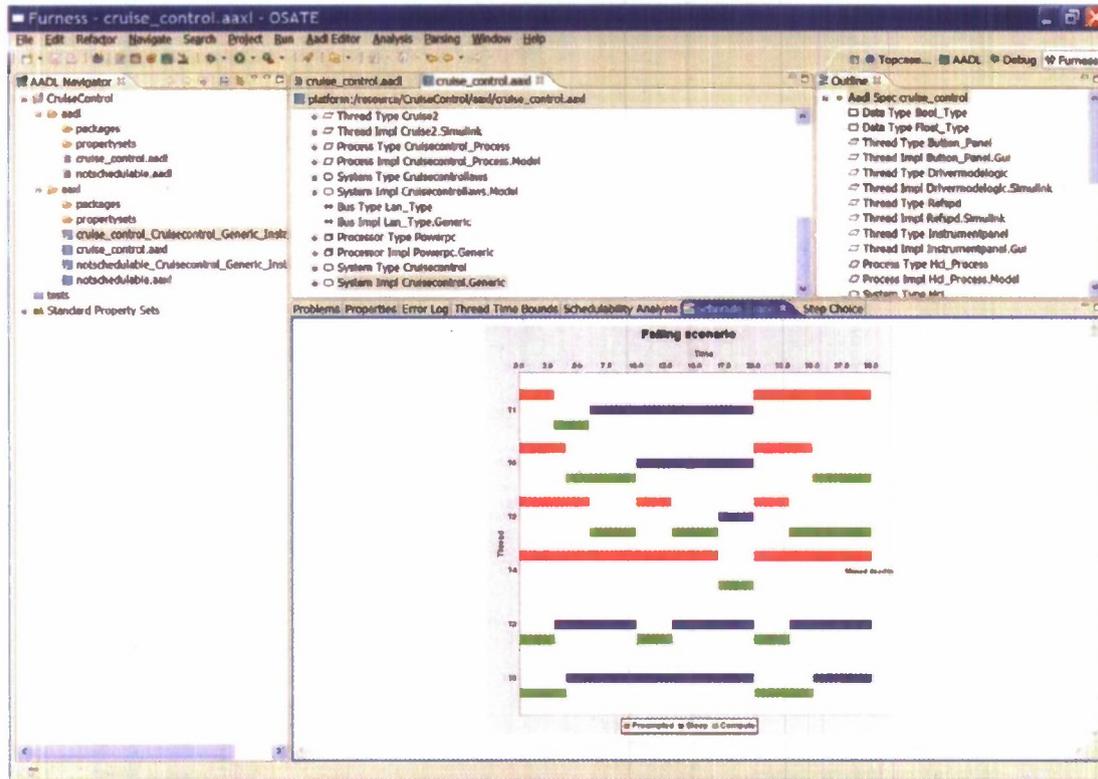


Figure 2: Schedulability Analysis Overview—System Not Schedulable

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.

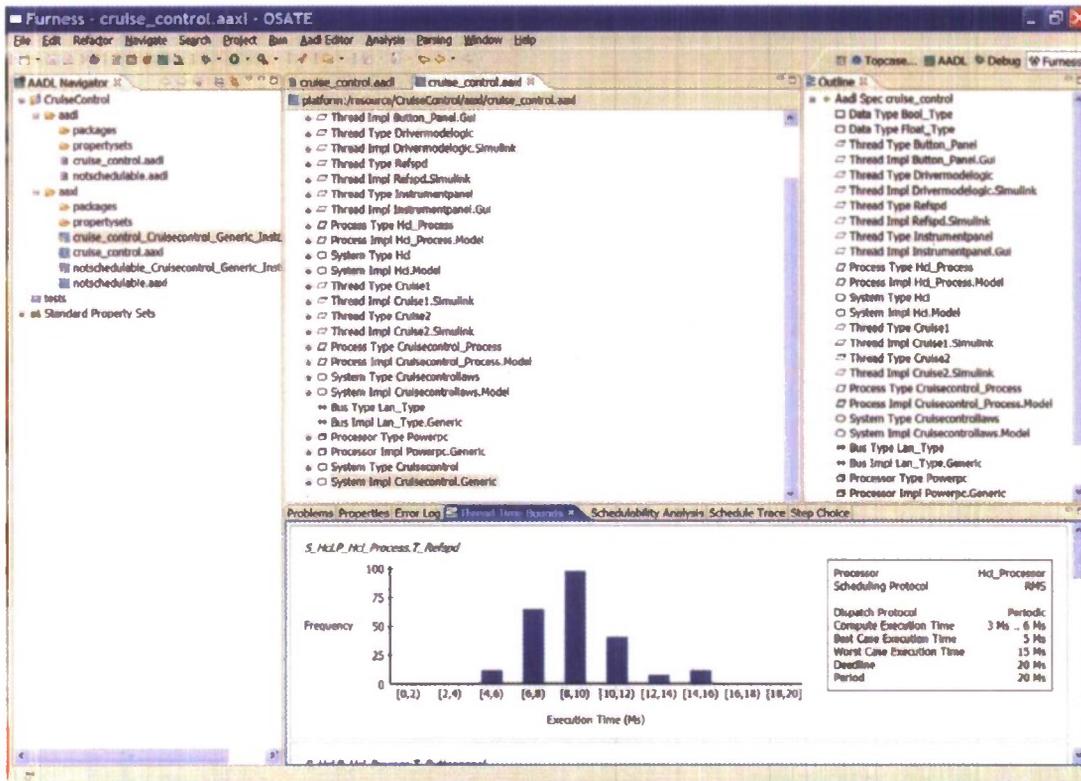


Figure 3: Schedulability Analysis Overview—System Schedulable

AADL System Instance Simulation/Debug

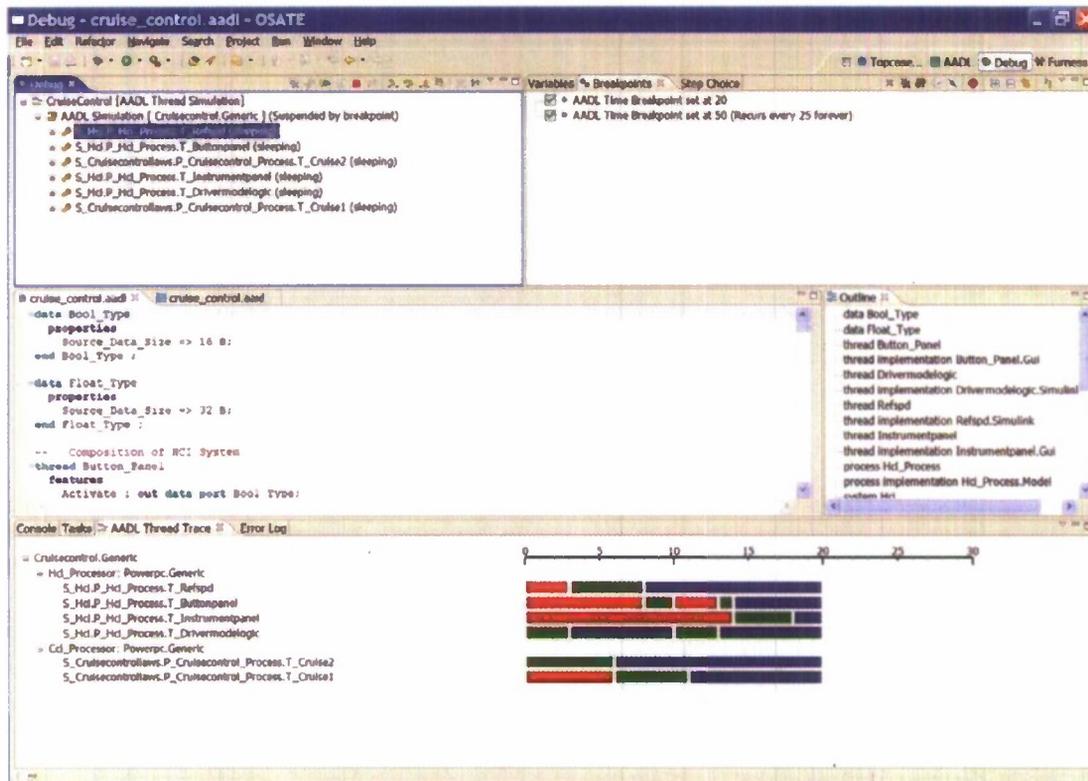


Figure 4: Simulation/Debug View—Breakpoints and Trace

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.

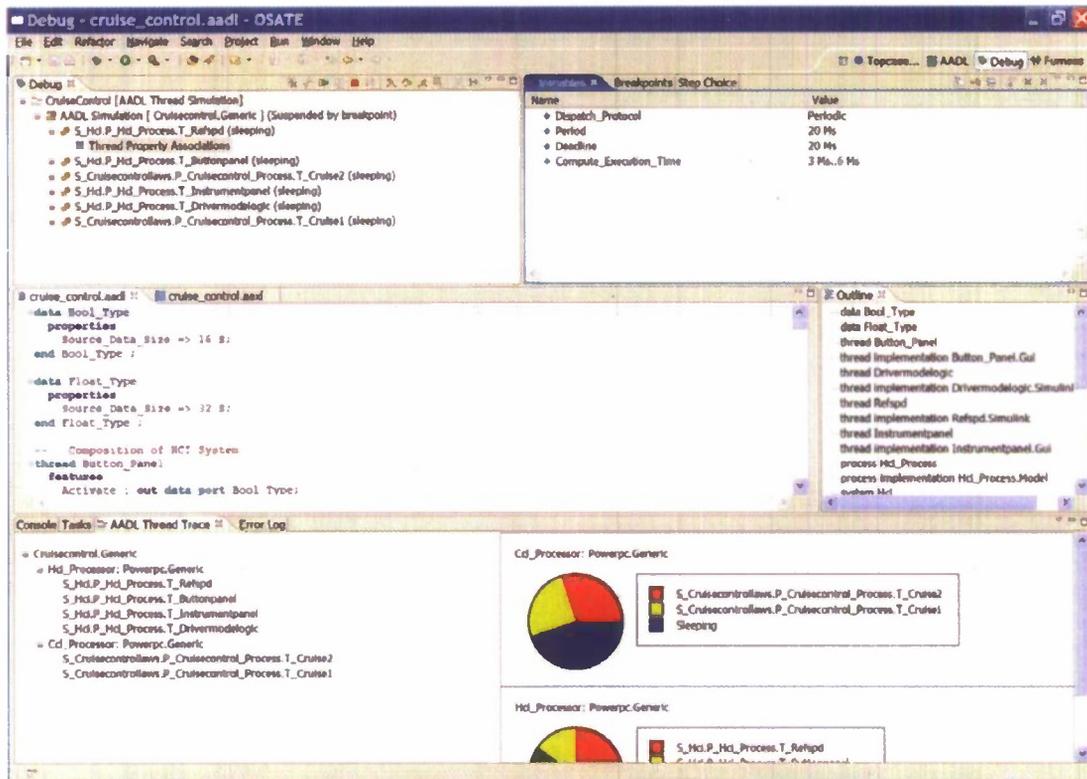


Figure 5: Simulation/Debug View—Thread Property Associations and CPU Utilization

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.

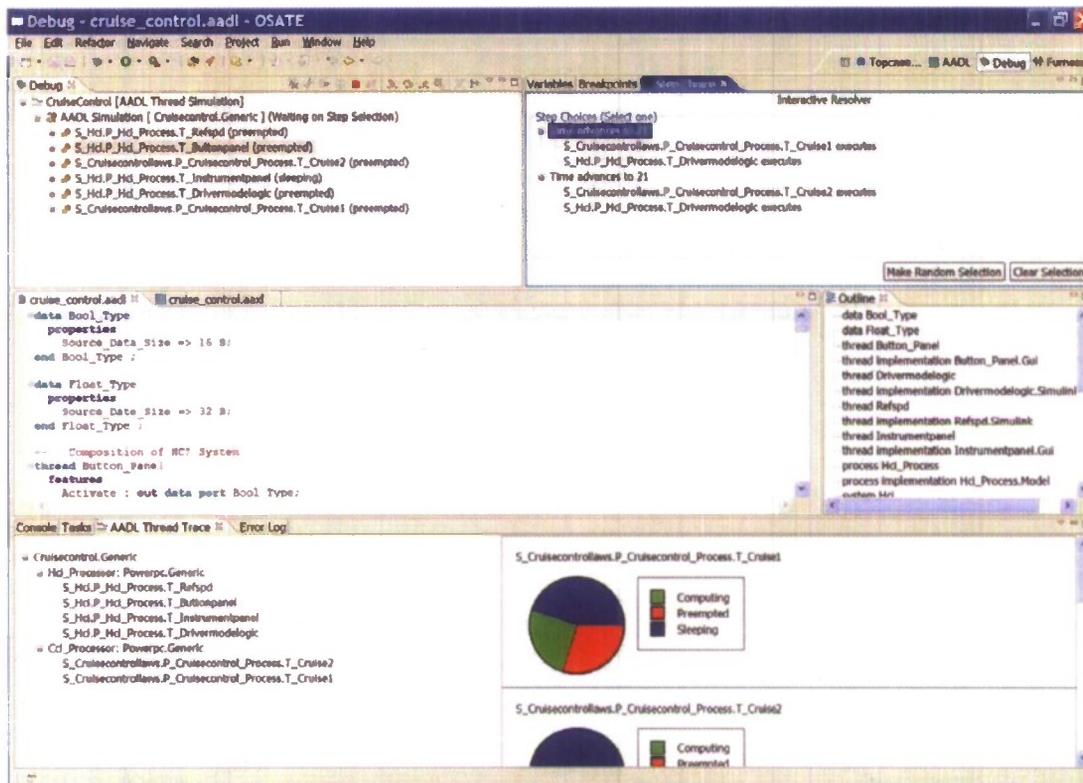


Figure 6: Simulation/Debug View—Interactive Non-Determinism and Thread States

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.

6. Results Obtained

During the period of performance of this contract accomplishments have been concentrated in the area of developing tools and techniques for model-driven design and analysis of embedded real-time systems. The tools effort has been focused on expanding the simulation functionality of the Furness Toolset. A minimal implementation first released during project year one was expanded to cover a much larger portion of the syntactic and semantic features of the AADL language during project year two and the no-cost extension year.

New findings focused on developing the conceptual framework to allow AADL models to be translated into the ACSR formalism, for which we have extensive techniques and tools for performing analysis of real-time system models. This bi-directional translation from ACSR and AADL (and back) enabled the creation of tools for analysis of high-level architecture specifications including schedulability analysis for key scheduling protocols, and dynamic simulation of system models.

The accomplishments/new findings described here represent the realization of the research and development plan presented in the project's original proposal. As described in that proposal, the fruit of our efforts is directly applicable to Air Force, broader DoD, and civilian technology challenges wherever safety critical embedded real-time systems are designed, analyzed, implemented and deployed. This includes (but is certainly not limited to) military and civilian avionics systems, airborne and ground-based weapons systems, automotive applications, medical devices for military and civilian applications, etc.

Over the period of performance Clarke and Sokolsky have become active participants in the Society of Automotive Engineers AS-2C subcommittee responsible for creating and refining the AADL language standard. They have made presentations on the Furness Toolset and on basic principles for refining/extending the AADL at numerous quarterly meetings. During project year two Sokolsky was nominated and approved for the post of standards committee co-chair by the SAE AS-2C standards committee charged with defining the AADL.

Clarke and Sokolsky have made presentations on the Furness Toolset at each quarterly meeting. They have also made presentations on basic principles for refining/extending the AADL at numerous quarterly meetings, topics including net-centric systems modeling, indexing/replication of specification elements, protocol modeling and connection patterns.

Fremont Associates took a leadership role in defining/refining a tools strategy for the AADL through a tools working group. This includes holding teleconferences among interested parties, and reporting on tools working group activities at the quarterly AS-2C subcommittee meetings.

Clarke and Sokolsky have had numerous contacts with Bruce Lewis of the Army's Software Engineering Directorate at the Redstone Arsenal in Huntsville, Alabama. One specific instance includes email and telephone interactions with Bruce Lewis regarding the relative scopes, merits and capabilities of the AADL vs. SysML that occurred on and about August 7, 2006.

7. Estimates of Technical Feasibility

The creation and release of the Furness Toolset has demonstrated the technical feasibility of the approach that was proposed. Ongoing issues related to current and future viability of the toolset include the following:

1. Maturity of the AADL standard. The effort expended during the period of performance was focused on tools to support the Version 1 AADL language specification. In late 2008 a new language standard, commonly referred to as AADL Version 2, was published by the SAE.

Work on AADL Version 2 began immediately upon release of the Version 1 standard, and was perceived to be an impediment to early adoption of tools. Enthusiasm for the Version 1 language suffered as users focused on key features to be contributed to the Version 2 language.

2. Commercial potential of the OSATE implementation. OSATE was planned and executed largely as a proof-of-concept for the AADL language and preliminary tools concepts. As the Furness Toolset developed and users considered the use of Furness Toolset (based on OSATE) for production models, the prototype nature of the tools became an impediment to application.

OSATE's handling of workspaces, concurrent development, large models, configuration management, *etc.*, will all have to be addressed if the tools are to be more widely adopted in military and industrial settings. The choice to implement the tools in Java and integrate them with the Eclipse tool bench may also have to be reconsidered if large models and models with large underlying state spaces are to be handled efficiently.

3. Integration with existing, proprietary tools. A monolithic AADL-focused tools suite is unlikely to succeed without complimentary support from existing proprietary tools vendors. To date, these vendors have taken a wait-and-see approach, watching the language develop and attempting to assess its potential for widespread adoption.

8. Personnel Supported

Duncan Clarke, Fremont Associates, LLC
Oleg Sokolsky, University of Pennsylvania
Insup Lee, consultant to Fremont Associates, LLC
Judy Baxley, project manager, Fremont Associates, LLC
Daryl Prickett, developer, Fremont Associates, LLC
Andrew Weaver, developer, Fremont Associates, LLC
Valentina Sokolskaya, subcontractor to Fremont Associates, LLC
Jesung Kim, University of Pennsylvania

9. Publications

Sokolsky, O., Lee, I., and Clarke, D., "Schedulability Analysis of AADL Models," The 14th International Workshop on Parallel and Distributed Real-Time Systems, a workshop of the 2006 IEEE International Parallel and Distributed Processing Symposium, Island of Rhodes, Greece, April, 2006.

I. Lee, A. Philippou, and O. Sokolsky, "Resources in Process Algebra," Journal of Logic and Algebraic Programming, Vol. 72, pp. 98--122, May/June 2007.

A. Philippou and O. Sokolsky, "Process-Algebraic Analysis of Timing and Schedulability Properties," Handbook of Real-Time and Embedded Systems, Chapman and Hall/CRC, 2007.

O. Sokolsky, I. Lee, and D. Clarke, "Process-Algebraic Interpretation of AADL Models," The 14th International Conference on Reliable Software Technologies - Ada-Europe 2009, Brest, France, June 2009 (to appear).