

AD-A220 510

FINAL TECHNICAL REPORT

ADVANCED DEVELOPMENT ENVIRONMENTS, CONTINUATION

SPAWAR Contract No. N00039-84-C-0211 T12

Submitted by:

David C. Luckham
Principal Investigator
Department of Electrical Engineering
Stanford University
Stanford, California 94305-4055

DTIC
ELECTE
APR 11 1990
S D D
D

Prepared for:

John Pucci
SPAWAR 3241C2
Department of the Navy
Space and Naval Warfare Systems Command
Washington, D.C. 20363-5100

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

Approved:

David C. Luckham, Project Director
Professor of Electrical Engineering (Research)
Computer Systems Laboratory
Stanford University
(415) 723-1242
luckham@anna.stanford.edu

based on machine processable formal specifications. The new wide spectrum specification language and environment tools will be targeted to large systems implemented in Ada, and particularly concurrent, distributed, trusted, and time-critical systems. It will provide features for specifying systems containing both software and hardware components. The environment support tools will automate applications of formal specifications to every stage of the systems development process.

3 ACCOMPLISHMENTS

1. Development of the Anna-1 runtime checking and debugging system

Under Task 12 the Anna support tools for transforming annotations into Ada code have been developed to alpha status for distribution. This toolsuite is described in detail in Appendix C. It provides a very flexible and portable facility for testing and debugging Ada programs utilizing high level annotations written in Anna. It now includes a substantial user interface for interactive debugging, and support of new methodologies, such as two-dimensional pinpointing, for debugging hierarchically structured Ada software such as nested packages. A current distribution list for this toolsuite is given in Appendix A.

2. Computational checking of algebraic specifications

Under Task 12 we have developed algorithms for runtime computational checking of algebraic specifications. This includes, for example, Anna package axioms used to specify the interfaces of Ada packages. This work represents a breakthrough in developing automated techniques for applying high level formal specifications in software engineering. Previously, runtime checking methods were applicable only to assertion-like specifications such as input/output annotations of subprograms, type annotations, Ada range constraints, and assertions.

These new checking algorithms have been implemented in the Anna checking system, and their practicality and applicability is the subject of a number of studies under task 12, task 22, and also by other organizations. This work is documented in Appendix C.

3. Semi-formal methods of applying formal specifications

Under Task 12 we have developed and experimented with new methods of using formal specifications in software development processes. These methods are called "semi-formal" because they utilize formal specifications, but do not require use of formal proof. They are therefore not as demanding as strict formal methods. The objective is to develop practical methods that are acceptable in present-day industrial software engineering practice, and improve upon current practice. The automated Anna and TSL toolsets are used as a support base for semi-formal methods. In particular we have developed semi-formal methods for locating errors in complex layered Ada software, whenever an inconsistency between top level specifications and code is detected by the tools. These methods are independent of any Ada compiler/environment. The user interacts with the toolset using only the formal specifications of the software.

This work is documented in Appendix B of this report, and is scheduled for publication in IEEE Software in 1990.

4. Anna package specification analyzer

Under Task 12 an Anna package specification analyzer has been developed for interactive analysis of formal specifications of Ada packages. The analyzer applies logical rules to package specifications written in Anna to predict the behavior of a package prior to implementation. It provides facilities to hypothesize various uses of a package, and to answer questions about the consequences. This tool is built using formally specified module interfaces for component tools. Component tools include the Ada/Logic prover, the Ada/Anna incremental semantic rule checker, and a graphics windowing system.

This work is documented in Appendix D of this report.

5. Multi-processor debugger for Ada tasking programs

Under Task 12 the TSL-1 toolset has been engineered as an environment tool for debugging Ada tasking programs running on multi-processor machines such as Encore and Sequent. It monitors Ada tasking programs at runtime for consistency with TSL-1 pattern specifications, and automatically provides traces of predefined Ada tasking actions. Traces are automatically produced by an Ada program after it has been processed by the TSL toolset. This tool fills a gap in current Ada environments for multi-processors, since there is no viable commercial debugger for multiple processors. It is expected to gain wide commercial acceptance in the Ada community.

Appendix E is a draft implementation guide for the latest TSL support tool.

4 TECHNOLOGY TRANSFER UNDER TASK 12

1. General distribution

Anna tools are distributed worldwide, and are usually requested by universities for teaching and research, and by industry for evaluation of applicability to specific problems or for general Ada environment enhancement. Recent distributions include, for example: University of New Mexico, Martin Marrietta Corp., Northrup Corp., Software Engineering Institute at CMU, Florida Atlantic University, and Queen's University, Belfast, Northern Ireland.

Appendix A gives a list of distribution of Anna tools under both Task 12 and the on-going Task 22.

5 OTHER INFORMATION

5.1 Major personnel changes

None.

5.2 Recent publications during last year of Task 12

1. L.M. Augustin, B.A. Gennart, Y. Huh, D.C. Luckham and A.G. Stanculescu, "Verification of VHDL Designs using VAL". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1988.
2. Ganapathi, M. and Mendal, G.O., "Issues in Ada Compiler Technology". IEEE Computer, pp. 52-60, February 1989.
3. D.C. Luckham, S. Sankar, and S. Takahashi, "Two Dimensional Pinpointing: An Application of Formal Specifications to Debugging Packages". To appear in IEEE Software. (Also Stanford University Computer Systems Laboratory Technical Report CSL-TR-89-379.)
4. D.S. Rosenblum and D.C. Luckham, "Testing the Correctness of Tasking Supervisors with TSL Specifications". Proceedings of TAV3-SIGSoft 89: Software Testing, Analysis and Verification Symposium, December 13-15, 1989, Key West, Florida, 1989.
5. S. Meldal and D.C. Luckham, "Specifying and Observing Concurrent Programs". Presented at the Workshop on Large Grain Parallelism, CMU, October, 1989.
6. D.C. Luckham, "Panelist Viewpoint: Will There Be an Ada 10X?". Presented at the Tri-Ada 89, Panel: Point/Counterpoint: Ada 9X in Context, Pittsburgh, October, 1989, Proceedings of Tri-Ada Conference, pp. 656-658.
7. L. Augustin, "An Algebra of Waveforms". Proceedings of the IFIP International Workshop on Applied Formal Methods for Correct VLSI Design, Belgium, November, 1989, pp. 159-168.
8. L. Augustin, "Timing Models in VAL/VHDL". International Conference on Computer-Aided Design (ICCAD) '89 Digest of Technical Papers, Santa Clara, CA, November, 1989.
9. D.C. Luckham, N. Madhav, and W. Mann, "On the Software Engineering of Automated Deduction Systems". To appear in the 60th Birthday Commemorative Volume dedicated to J.A. Robinson.
10. S. Meldal, "Predicting the Future: An Axiomatic Semantics of Spawning", Accepted for publication in Distributed Computing, 1989.
11. D.P. Helmbold, "The Meaning of TSL: An Abstract Implementation of TSL-1". Technical Report No. CSL-TR-88-353, March 1988.

12. D.C. Luckham, S. Sankar, S. Takahashi, "Two Dimensional Pinpointing: An Application of Formal Specification to Debugging Packages". Technical Report No. CSL-TR-89-379, April 1989.
13. R. Neff, "Ada/Anna Specification Analysis". Technical Report No. CSL-TR-89-406, December 1989.
14. S. Sankar, "Automatic Runtime Consistency Checking and Debugging of Formally Specified Programs". Technical Report No. CSL-TR-89-391, August 1989.