

Technical Report
CMU/SEI-87-TR-030
ESD-TR-87-193

Criteria for Constructing and Using an Ada Embedded System Testbed

Nelson H. Weiderman
November 1987

Technical Report

CMU/SEI-87-TR-30

ESD-TR-87-193

November 1987

Criteria for Constructing and Using an Ada Embedded System Testbed



Nelson H. Weiderman

Ada Embedded Systems Testbed Project

Approved for public release.
Distribution unlimited.

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This technical report was prepared for the

SEI Joint Program Office
ESD/XRS
Hanscom AFB, MA 01731

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

This report has been reviewed and is approved for publication.

FOR THE COMMANDER

Karl H. Shingler SIGNATURE ON FILE
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

Copyright © 1987 by the Software Engineering Institute.

This document is available through the Defense Technical Information Center. DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145.

Copies of this document are also available through the National Technical Information Services. For information on ordering, please contact NTIS directly: National Technical Information Services, U.S. Department of Commerce, Springfield, VA 22161.

Criteria for Constructing and Using an Ada Embedded System Testbed

Abstract. The purpose of this report is to list some of the criteria used in five aspects of the project: the hardware configuration, the software configuration, the real-time application, the Ada real-time experiments, and the benchmarking and instrumentation techniques. Each criterion will include a rationale. Each of the criteria listed in this report will be categorized as either essential, highly desirable, or desirable.

1. Introduction

The Ada Embedded Systems Testbed (AEST) Project was initiated in October 1986 and will be continued in 1987-88. The purpose of the AEST Project is to investigate some of the critical issues in using Ada for real-time embedded applications, particularly the extent and quality of the run-time support facility provided by Ada implementations. The Ada run-time is an execution environment that provides services such as process management, storage management, and exception handling for supporting the execution of Ada programs. These services were, in the past, provided either by the application programmer or by a small real-time executive.

One project objective is to collect, classify, track, and disseminate information about the use of Ada in real-time embedded systems. This will help to make the SEI a center of expertise for such information. Another objective is to create and expand a general testbed for experimentation. The testbed must accommodate different target processors, different compilers, and different toolsets. It must be designed to be flexible, reconfigurable, and evolvable. There should be both hardware and software measurement techniques so that performance data can be independently collected and verified in a non-intrusive manner. Finally it is an objective of the project to generate new information about using Ada in real-time embedded systems. This information would be in the form of benchmark test results, higher level experiment results, and lessons learned in designing and implementing real applications in Ada.

Work to date has concentrated on identifying the appropriate issues to investigate, assembling some of the necessary hardware and software for the testbed, and acquiring some of the existing benchmark test suites. A summary description of the benchmarks currently available and an evaluation of their applicability to the AEST Project is contained in a report entitled *A Summary of Real-Time Performance Benchmarks for the Ada Programming Language* [Donohoe 87]. A report entitled *Ada for Embedded Systems: Issues and Questions* [Weiderman 87] contains issues related to the use of Ada language, the facilities provided by the Ada run-time system, and the tools required for system development.

The purpose of this report is to list some of the criteria used in five aspects of the project: the hardware configuration, the software configuration, the real-time application, the Ada real-time experiments, and the benchmarking and instrumentation techniques. Each criterion will include a rationale. Each of the criteria listed in this report will be categorized as either essential, highly desirable, or desirable. Essential criteria are those which, when not met, will severely impact the purpose and objectives of the AEST Project. Highly desirable criteria are those which, when not

met, will limit the value or extent of the results that are obtained. Desirable criteria are those which, when not met, may adversely impact the productivity of the project personnel or the cost of the project, but do not severely impact the results.

It should be recognized that it may be difficult or impossible to meet many of the criteria defined in this report. Some of the criteria may be conflicting to one degree or another. Furthermore, all the criteria are not at the same level of detail. For completeness, general as well as specific criteria have been included so that some of the criteria are overlapping. The criteria have been ordered within each group from the essential criteria to the desirable criteria.

2. Criteria for the Hardware Configuration

One of the components of the testbed is the hardware. It includes the target computer as well as the ancillary processors, connectors, I/O equipment, and test equipment necessary to carry out experimentation. The purpose of this section is to list the criteria for the suite of hardware needed for the AEST Project.

Criterion HW1 : The generic testbed hardware must support compilation of Ada programs, downloading of Ada programs to a target system, simulation of the environment for a target system, and monitoring of Ada programs running in the target environment.

Rationale: These are basic activities of software development for mission-critical computer resource (MCCR) systems.

Criticality: Essential

Criterion HW2 : The target systems should be representative of those which are presently used in MCCR applications or for which future use is anticipated or desirable.

Rationale: We must address the needs of the DoD.

Criticality: Essential

Criterion HW3 : There should be four interacting computing systems — a host development system, a target system, an environment simulator, and a monitor system (see Figure 1).

Rationale: This configuration provides all the necessary functionality and allows the flexibility to change the target without changing any of the other components.

Criticality: Highly desirable

Criterion HW4 : The host system should be one for which there are a variety of cross compilers.

Rationale: This is required to leverage the expertise of the project group and to get maximum information and benefit from the available hardware.

Criticality: Highly desirable

Figure 1: Generic Embedded System Testbed

Criterion HW5 : The target systems should be ones for which there are a variety of cross compilers.

Rationale: This criterion allows the comparison of different compilers on identical architectures. The targets for which there are multiple cross compilers also represent those targets that are technologically advanced or DoD standards.

Criticality: Highly desirable

Criterion HW6 : The target systems should represent modern architectures which are capable of supporting the demands of higher level languages such as Ada.

Rationale: We must be aware of the current state-of-the-art with regard to target processors and their ability to efficiently handle Ada.

Criticality: Highly desirable

Criterion HW7 : The test equipment should include a logic analyzer.

Rationale: The project needs non-intrusive testing capabilities due to the sensitivity of the timing constraints. A logic analyzer provides the capability to check the periodicity of activities, uncover bottlenecks, and check software timings independent of the software.

Criticality: Highly desirable

Criterion HW8 : The hardware testbed should be flexible, reconfigurable, and evolvable.

Rationale: This is a requirement for any hardware system on which experimentation will occur. The IEEE Computer special issue on distributed system testbeds [Berg 82] contains some general guidelines for building testbeds. The alternative of building separate hardware for each experiment is impractical and expensive.

Criticality: Desirable

Criterion HW9 : There should be the ability to replace one target system with another without changing or replacing the entire testbed. This requires that the interfaces between the various systems and between systems and IO devices be reasonably standard so that costly devices can be shared.

Rationale: While the mechanism for downloading or interfacing with other computers can be expected to change, it is desirable to reuse the other components of the testbed for multiple targets.

Criticality: Desirable

3. Criteria for the Software Configuration

The software criteria for the testbed can be subdivided into two categories. The first is the Ada cross compiler criteria, and the second is the cross development environment criteria. The cross compiler must satisfy certain minimum criteria to be usable in an embedded system application. Other criteria are desirable, but not essential. The cross development environment contains a set of tools that make the job of development of MCCR software easier.

3.1. Cross Compiler Criteria

Criterion SW1 : The compiler should be targeted to MicroVAX-II or MC680x0 microprocessors.

Rationale: These are the first two targets to be explored by the project. In the second year this constraint will be relaxed.

Criticality: Highly desirable

Criterion SW2 : The following options related to pragmas should be supported:

- Warning messages for all unrecognized pragmas *Reference Manual for the Ada Programming Language*¹ [2.8(11)] [ANSI 83].
- Pragma INLINE should be supported (the compiler should detect and flag any situations where the pragma cannot be executed, for example, a recursive subprogram) [LRM 6.3.2(4)]
- Pragma INTERFACE should be supported for the assembly language of the target machine and for other languages for which appropriate library software exists [LRM Appendix B]
- Pragmas SUPPRESS, ELABORATE, LIST, and PAGE should be supported [LRM Appendix B]

Rationale: The compiler should let the programmer know what actions are being taken. The pragmas INLINE, SUPPRESS, and ELABORATE are necessary for investigating performance issues. Pragma INTERFACE is necessary for optimization and reusability. LIST and PAGE are useful for suppressing and organizing source code listings.

Criticality: Highly desirable

Criterion SW3 : The compiler should provide SHORT_INTEGER, LONG_INTEGER, SHORT_FLOAT, and LONG_FLOAT types [LRM 3.5.7(7,8)]. The compiler should provide unsigned data types UNSIGNED_LONGWORD, UNSIGNED_WORD, and UNSIGNED_BYTE.

Rationale: It can be expected that most real-time applications will use a variety of short and long data types.

Criticality: Highly desirable

¹Reference Manual for the Ada Programming Language will be referred to as the LRM throughout the rest of this report.

Criterion SW4 : The minimum range on predefined type PRIORITY should be 0 to 15.
Rationale: Multiple levels of priority are necessary to support different scheduling models.
Criticality: Highly desirable

Criterion SW5 : The following features (discussed in Chapter 13 of the LRM) should be supported:

- Representation clauses [LRM 13.1]
- Enumeration representation clauses [LRM 13.3]
- Record representation clauses [LRM 13.4]
- Address clauses (interrupts) [LRM 13.5]
- Change of representation [LRM 13.6]
- Interface to other languages (assembler, HOL) [LRM 13.9]
- Unchecked type conversions [LRM 13.10.1]
- Length clauses [LRM 13.2]
- Unchecked storage deallocation [LRM 13.10.1]

Rationale: These are precisely the features that were inserted into the language to support embedded computer system (ECS) applications.
Criticality: Highly desirable

Criterion SW6 : The compiler should have low level I/O packages to support real-time device drivers. [LRM 14.6]
Rationale: This is a requirement of ECS applications.
Criticality: Highly desirable

Criterion SW7 : The compiler should have the following real-time characteristics:

- DURATION'SMALL less than 100 microseconds
- SYSTEM.TICK less than 1 millisecond
- Tasks executing a delay should be rescheduled within SYSTEM.TICK of the expiration of the delay
- User selectable scheduling algorithm (preferably by pragma)

Rationale: These are necessitated by the severe timing constraints of real-time embedded systems. In many applications there are timing requirements in the 1 millisecond range. The scheduling algorithm must be known by, if not controlled by, the application programmer.
Criticality: Highly desirable

Criterion SW8 : Successive choices of new compilers for test and experimentation should be based on:

- the need to support a new target computer
- variety
- the inclusion of one or two compilers likely to be industry standards

Rationale: First priority is to support selected target computers. Experimentation with a variety of compilers will make the results more credible and less subject to the idiosyncrasies of a particular compiler. Including one or two "standards" will provide stable reference points for comparison.

Criticality: Highly desirable

Criterion SW9 : Tasks dependent on library packages are required to terminate. [LRM 9.4(13)]

Rationale: This ensures that all dependent tasks terminate before the main program.

Criticality: Desirable

Criterion SW10 : The compiler must be hosted on a MicroVAX-II running either the VMS or ULTRIX operating system.

Rationale: This is a practical constraint dictated by the hardware available at the SEI as well as the experience of the software engineers assigned to the project. This constraint may be relaxed as time passes.

Criticality: Desirable

Criterion SW11 : The compiler should be able to generate code for the host machine and the target machine.

Rationale: This permits some of the initial unit testing to be done on the host machine and frees the more restricted target machine for integration testing and operational testing.

Criticality: Desirable

Criterion SW12 : The compilation speed should be 500 lines per minute or greater on a MicroVAX-II. The disk space required should be less than 50 megabytes. The main memory required should be less than 8 megabytes. The virtual memory required should be less than 40 megabytes. The run-time resources should be less than 20 kilobytes.

Rationale: These numbers are known to be achievable. Numbers not achieving these standards are indications of poor engineering.

Criticality: Desirable

Criterion SW13 : Storage should be reclaimed when an object becomes inaccessible [LRM 4.8(7)]. Furthermore, the actual time of reclamation should be under programmer control.

Rationale: Without storage reclamation (garbage collection), there will be a tendency to run out of dynamic memory space. Programmer control is necessary so that critical timing deadlines are not missed.

Criticality: Desirable

Criterion SW14 : The only restrictions on the main program (which appear in Appendix F of the LRM) should be that the main program is a subprogram without parameters. In case of a function, the result type should be a discrete type [LRM 10.1(8)].

Rationale: In an embedded computer system there should be no operating system, and the mechanism for invoking a main program should be as simple as possible.

Criticality: Desirable

Criterion SW15 : The performance of the Ada run-time system should satisfy the following requirements:

- Interrupt latency less than 100 microseconds
- Overhead for simple task switch less than 200 microseconds
- Overhead for simple task rendezvous less than 200 microseconds

Rationale: Fast context switches are required by the severe timing constraints of real-time systems. Failure to meet these performance standards will encourage workarounds representing poor software engineering practices.

Criticality: Desirable

Criterion SW16 : The compilation system should have a sophisticated library management system. Among the functions to be supported are creation and deletion of program libraries, sharing of program libraries, interrogation of program libraries (list unit name and type, list unit dependencies, determine completeness, determine recompilation order), and manipulation of program libraries (removing a compilation unit, clearing the entire library).

Rationale: These are basic functions of a library management system of an Ada compilation system.

Criticality: Desirable

Criterion SW17 : The compilation system should support compilation management activities including batch and interactive compilation modes, listing of compilation units obsolesced by compiling another unit, and automatic recompilation of units obsolesced by compiling another unit.

Rationale: These functions make the software engineer much more productive in developing an application.

Criticality: Desirable

Criterion SW18 : The compilation system should have comprehensive documentation. This should include programming restrictions and known bugs.

Rationale: The system may be difficult to use without good documentation.

Criticality: Desirable

Criterion SW19 : The compilation system should have informative diagnostic (error) messages.

Rationale: This function makes the software engineer more productive in developing an application.

Criticality: Desirable

Criterion SW20 : The compilation system should clearly document all implementation dependent characteristics [LRM Appendix F] including:

- The form, allowed places, and effect of every implementation-dependent pragma [LRM 2.8(8)]
- The status of each language defined (standard) pragma: supported or unsupported by the implementation
- The name and type of every implementation-dependent attribute [LRM 4.1.4(4)]
- The specification of the package SYSTEM [LRM 13.7(1)]
- A list of all restrictions on representation clauses [LRM 13.1(10)]
- The conventions used for any implementation-generated names denoting implementation-dependent components [LRM 13.4]
- The interpretation of expressions that appear in address clauses, including those for interrupts [LRM 13.5(3)]
- Any restrictions on unchecked conversions [LRM 13.10.2]
- Any implementation-dependent characteristics of the input-output packages [LRM 14.1(1), 14.1(11), 14.2.1(13), 14.4(1)]

Rationale: These are specialized characteristics of the implementation and need to be part of a comprehensive documentation package.

Criticality: Desirable

3.2. Cross Development Environment Criteria

Criterion SW21 : The cross development environment must incorporate the following tools (see [Weiderman 87] for more detailed descriptions):

- Source code cross referencer
- Source code lister (optional assembly code interspersed in Ada listing)
- Ada linker

- Target load module (system) builder
- Symbolic, source level debugger (remote debugging capability)
- Load module downloader/receiver

Rationale: These functions are all critical to the efficient and well-structured development of real-time embedded systems.

Criticality: Essential

Criterion SW22 : The cross development environment should incorporate the following tools (see [Weiderman 87] for more detailed descriptions):

- Pretty printer
- Language sensitive editor
- Static profiler
- Frequency analyzer
- Cross assembler
- Target simulator
- Test manager
- Configuration manager
- Module manager

Rationale: These functions are all desirable for the efficient and well-structured development of real-time embedded systems.

Criticality: Desirable

Criterion SW23 : There must be comprehensive documentation and support for all the cross development environment tools.

Rationale: The usability of the tools is directly proportional to the quality of their documentation and support.

Criticality: Desirable

4. Criteria for the Real-Time Application

The purpose of writing a complete application for the testbed is to provide a vehicle to test the Ada language, the run-time system, and the target processor at the coarsest level of granularity. The application will provide a proof of concept that Ada can be used for the design and implementation of time-critical MCCR applications. It will also be a generator of additional issues, provide a context for the usability of the information gained through experimentation, and provide a software engineering exercise for real-time programming in Ada.

Criterion AP1 : This is typical ECS functionality that needs to be implemented in Ada. In particular, it should have strict timing demands, multiple concurrent activities, low-level I/O, error handling, interrupts, and periodic activities.

Rationale: This is precisely the environment for which Ada was designed. It is this environment for which Ada's utility needs to be demonstrated.

Criticality: Essential

Criterion AP2 : The application should be large enough to stress the system and small enough to be feasible with respect to the equipment and personnel resources available.

Rationale: All the Ada features and capabilities for real-time embedded systems should be used in combination in order to provide a true test for Ada. The implementation time should not exceed approximately six months (2-4 people) so that results will be timely.

Criticality: Essential

Criterion AP3 : The application should be an existing ECS application or should be derived from an existing ECS application. If a subset of an application is used, it should be a subset that does not simplify the functionality implemented.

Rationale: The application will be credible to the MCCR community only if it can be related to a "real" application.

Criticality: Highly desirable

Criterion AP4 : The application should be easily ported from one target processor to another. This requires that there be a minimum of specialized sensor and actuator hardware requiring individual interfaces.

Rationale: The usefulness of the application is enhanced if it can be reused across different target architectures.

Criticality: Highly desirable

Criterion AP5 : There should be a working version of the application available.

Rationale: The functionality and performance of the working application (in another higher level language or assembly language) can be compared with the Ada version for functionality and performance.

Criticality: Highly desirable

Criterion AP6 : The system should be visually oriented. That is, there should be a graphical user interface or devices that can be easily controlled.

Rationale: This criterion makes demonstrations more interesting and would add to the software showcase of the SEI.

Criticality: Highly desirable

Criterion AP7 : The schedule for the development of the application must include time for:

- recording design decisions
- analyzing alternative designs
- recording problems encountered in applying Ada to the application
- recording lessons learned

Rationale: The usefulness of the application work will be enhanced if it is more than a bare "existence proof."

Criticality: Highly desirable

Criterion AP8 : The application should require a minimum of domain-specific knowledge.

Rationale: The majority of the personnel assigned to design and implement the application will be software engineers. Knowledge of physics or Kalman filters, for example, will not enhance the information derived from the application with regard to the use of Ada in embedded systems.

Criticality: Desirable

Criterion AP9 : There should be an industrial or government sponsor that is willing and able to provide domain-specific knowledge about the application. Furthermore, there should be easy access to this knowledge, preferably in the form of an affiliate working at the SEI.

Rationale: To make the application relevant and credible requires that someone knowledgeable in the domain area work closely with the project personnel. This person would help to define the requirements and help with the design.

Criticality: Desirable

Criterion AP10 : There should be high quality documentation available for the system being implemented. The documentation should include a system overview, system requirements specification, software requirements specification, high level design, detailed design, code, and test suites.

Rationale: The project is not particularly concerned with the early portion of the lifecycle. While the design will be influenced by Ada, the project should invest as little resource as possible in defining the problem and specifying requirements.

Criticality: Desirable

5. Criteria for Ada Real-Time Experiments

The purpose of the Ada real-time experiments is to design and develop a set of experiments that assess the feasibility of implementing essential embedded system functionality in Ada. The number of experiments designed should be reasonable (with respect to resources) and should provide maximum coverage of the issues and questions defined in [Weiderman 87].

Criterion EX1 : The experiments must provide direct support of the application being developed by the AEST Project as well as indirect support to other similar MCCR applications.

Rationale: The AEST application is being designed to address many of the same problems that exist in the broader MCCR community. Support of the AEST application will guarantee support of the mission of the AEST Project.

Criticality: Essential

Criterion EX2 : The experiments should be defined broadly to examine a number of related issues rather than a single issue. They should be defined so as to address existing issues as well as raise new issues related to the use of Ada in real-time embedded systems.

Rationale: This will allow greater productivity in addressing the issues. Narrowly focused experiments are the purpose of the benchmarks and instrumentation task.

Criticality: Highly desirable

Criterion EX3 : Each group of experiments should be designed with a clear purpose, description, approach, set of measurements to be performed, and results to be achieved. All experiments should be carefully documented.

Rationale: This criteria represents sound experimental methodology and should be followed to facilitate transition of useful results.

Criticality: Highly desirable

Criterion EX4 : The analysis criteria must include functionality; i.e., can a certain function be implemented in a straightforward and efficient way using Ada? Alternatively, are there solutions (e.g., assembly language) that are straightforward and efficient?

Rationale: Not all functionality must be implemented in Ada. That which is straightforward and efficient in Ada should be written in Ada. Assembler should be used only when absolutely necessary.

Criticality: Highly desirable

Criterion EX5 : The analysis criteria should include objective performance measures. In particular, they should include execution speed on the target processor, system load module size, Ada run-time size, and source code size.

Rationale: These are important usability measures for Ada code running in MCCR systems.

Criticality: Highly desirable

Criterion EX6 : The analysis criteria should include subjective measures of the source code including its complexity, maintainability, readability, and portability.

Rationale: The software engineering benefits of using Ada in MCCR systems must be evaluated because they can, in the long term, overshadow some of the possible near-term deficiencies of Ada implementations. While these measures must be evaluated subjectively in experiments of this scope, they are important nevertheless. Some limited objective testing of portability can be done with the various targets of the testbed.

Criticality: Highly desirable

Criterion EX7 : The following experiment areas should be given priority as areas which support project application development of the AEST Project:

- representation clauses — bit manipulations, converting machine representations of values into other forms for external communication
- tasking with priorities
- periodic scheduling — tasks scheduled on basis of time intervals (e.g., every 5 milliseconds)
- interrupt handling — timer interrupts for scheduling purposes
- use of math library — trigonometric functions, matrix manipulation
- buffering mechanisms — shared data storage used for intertask communication
- data transfer mechanisms between machines — data transfer over parallel communication channels including actual data, protocol information, and handling of time-out conditions

Rationale: These are application-oriented functions that can be expected to be important for the AEST Project.

Criticality: Highly desirable

Criterion EX8 : The experiments should be feasible with respect to the equipment and personnel resources available.

Rationale: Approximately two to three people will be assigned to this task. The breadth of scope of the experimentation should take these constraints, as well as the hardware constraints into account.

Criticality: Desirable

Criterion EX9 : The following experiment areas should be given priority as areas which are relevant to application developers in general:

- low level I/O — interrupt handling, interfacing to devices
- concurrent control — multitasking capabilities, scheduling strategy employed
- application schedulers — cyclic, event/data driven, periodic
- time measurement — interrupt latency, context switch time, precision and overhead of delay
- time management — package calendar, clock resolution, SYSTEM.TICK, DURATION'SMALL
- internal representation — ability to access and manipulate bits
- error handling
- pragmas — which are supported, implementation semantics
- memory management — static and dynamic allocation, garbage collection
- numerical computation

Rationale: These experiment areas are important to application developers but of lower priority to the project application. They should be handled as time and resources permit.

Criticality: Desirable

6. Criteria for Benchmarks and Instrumentation Techniques

The benchmark tests and instrumentation techniques address the finest level of granularity for testing Ada for real-time embedded system applications. This area concentrates on the language features rather than the functionality required by the application programmer. The information generated by this activity is crucial to the application development and the Ada real-time experimentation. This activity must provide a firm practical and theoretical foundation on which other activities can draw.

Criterion BM1 : Whenever possible, high quality benchmark test suites should be imported, rather than designed and implemented at the SEI.

Rationale: Much good work has already been done. There is little point in reinventing test technology. Unfortunately, there is also much poor work that has been done. Project personnel must take great care to discriminate between the good work and the bad work.

Criticality: Essential

Criterion BM2 : The collection and reporting of test suite data should have strong practical and theoretical underpinnings.

Rationale: Tests cannot be designed, implemented, run, or interpreted haphazardly. There are many pitfalls in doing benchmark testing. It must be determined that you are measuring what you think you are. (A variety of purely software,

hardware-assisted, and purely hardware techniques should be investigated and compared against each other.) The configurations (hardware and software) must be tightly controlled and recorded.

Criticality: Essential

Criterion BM3 : The time and space implications of the various Ada features should be examined.

Rationale: These are the two most important criteria for MCCR application builders.

Criticality: Essential

Criterion BM4 : The benchmark tests should concentrate on individual language features rather than the interactions between language features.

Rationale: Other activities within the project will be addressing feature interactions.

Criticality: Highly desirable

Criterion BM5 : There should be hardware verification of software timing results.

Rationale: It has been our experience that software timings are very elusive because of software timers, daemons, and strange implementations by compilers. It has not been unusual to get negative values when the efficiency of a feature is determined by taking the time difference between a control program with a null loop and an experiment program with a loop containing the feature. Hardware verification of timings through the use of a logic analyzer is one method to increase the credibility of timing results.

Criticality: Highly desirable

Criterion BM6 : The test suite should include composite as well as individual benchmarks.

Rationale: Composite benchmarks give an overall figure of merit based on a group of features. Examples include the Whetstone and Dhrystone benchmark programs. Individual benchmarks attempt to isolate individual features for testing. Both these techniques provide useful results.

Criticality: Highly desirable

Criterion BM7 : The benchmark test suites should include measurements of the following Ada features (see [Donohoe 87] for more details):

- Subprogram calls
- Interrupt latency
- Context switching and synchronization (rendezvous)
- Dynamic storage allocation
- Exception handling
- Explicit type conversions (involving representation specs)
- Task elaboration, activation, and termination

- CLOCK function overhead
- TIME and DURATION evaluations

Rationale: These are important Ada features for real-time embedded system programming.

Criticality: Highly desirable

Criterion BM8 : Gaps in the existing benchmark test suites, particularly in the area of interrupt handling, interrupt latency, and context switching, need to be filled.

Rationale: Most of the existing tests have been run only on host systems where it is difficult to generate interrupts. Thus, there are few benchmark tests that deal with this important area.

Criticality: Highly desirable

Criterion BM9 : Testing should include results of compiler efficiency, but with a lower priority.

Rationale: Run-time efficiency is much more important than compile time efficiency.

Criticality: Desirable

Bibliography

- [ANSI 83] ANSI.
Reference Manual for the Ada Programming Language.
ANSI/MIL-STD-1815A, 1983.
- [Berg 82] Berg, H.K., ed.
Distributed System Testbeds: Experimentation with Distributed Systems.
IEEE Computer 15(10), October, 1982.
- [Donohoe 87] Donohoe, P.
A Survey of Real-Time Performance Benchmarks for the Ada Programming Language.
Technical Report CMU/SEI-87-TR-28, forthcoming, Software Engineering Institute, December, 1987.
- [Weiderman 87] Weiderman, N.H., Borger, M.W., Cappellini, A.L., Dart, S.A., Klein, M.H., and Landherr, S.F.
Ada for Embedded Systems: Issues and Questions.
Technical Report CMU/SEI-87-TR-26, Software Engineering Institute, December, 1987.

Table of Contents

1. Introduction	1
2. Criteria for the Hardware Configuration	2
3. Criteria for the Software Configuration	5
3.1. Cross Compiler Criteria	5
3.2. Cross Development Environment Criteria	9
4. Criteria for the Real-Time Application	10
5. Criteria for Ada Real-Time Experiments	13
6. Criteria for Benchmarks and Instrumentation Techniques	15
Bibliography	19

List of Figures

Figure 1: Generic Embedded System Testbed

3