5524[illegible] Johnson Grant 16
P36 IN-61-CR
243102
P-37

Ada Task Scheduling
A Focused Ada Investigation

Sue LeGrand

SofTech, Inc.

October 5, 1988

Cooperative Agreement NCC 9-16
Research Activity No. SE. 17

Research Institute for Computing and Information Systems
University of Houston - Clear Lake

T·E·C·H·N·I·C·A·L R·E·P·O·R·T

(NASA-CR-186072) ADA TASK SCHEDULING: A FOCUSED ADA INVESTIGATION Final Report (Houston Univ.) 37 p CSCL 09B

N90-13967

Unclas

G3/61 0243102

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information systems in 1986 to encourage NASA Johnson Space Center and local industry to actively support research in the computing and information sciences. As part of this endeavor, UH-Clear Lake proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a three-year cooperative agreement with UH-Clear Lake beginning in May, 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The mission of RICIS is to conduct, coordinate and disseminate research on computing and information systems among researchers, sponsors and users from UH-Clear Lake, NASA/JSC, and other research organizations. Within UH-Clear Lake, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business, Education, Human Sciences and Humanities, and Natural and Applied Sciences.

Other research organizations are involved via the "gateway" concept. UH-Clear Lake establishes relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research.

A major role of RICIS is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. Working jointly with NASA/JSC, RICIS advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research, and integrates technical results into the cooperative goals of UH-Clear Lake and NASA/JSC.

Ada Task Scheduling
A Focused Ada Investigation

Preface

This research was conducted under the auspices of the Research Institute for Computing and Information Systems by Sue LeGrand of SofTech, Inc. under the direction of John McBride, also of SofTech, Inc. Charles McKay, Director of the Software Engineering Research Center at the Univeristy of Houston-Clear Lake served as technical representative for RICIS.

Funding has been provided by the Mission Support Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA Technical Monitor for this activity was Robert MacDonald, Assistant to the Director for Research, Education and University Programs, NASA/JSC.

Ada Task Scheduling

A Focused Ada Investigation

October 5, 1988

RICIS Final Report, Contract SE.17

Prepared for

NASA Space Station Program Office
NASA Headquarters

Prepared by

Sue LeGrand

SofTech, Inc.
1300 Hercules Drive, Suite 105
Houston, TX 77058-2747

Contents

Preface

The design of real time systems often requires precise and predictable response from different components of the complete system. Ada tasking can be used to partition a real time system into potentially concurrent components. Sufficient control over these Ada tasks must be exercised to guarantee an acceptable level of total real time system response. Industrial and academic communities are addressing Ada task scheduling issues in several forums and through different research projects. Some users contend that commercially available Ada development systems do not offer the level of task scheduling control required to meet hard real time scheduling constraints. While efforts to address this problem continue, NASA must develop an effective plan to deal with requirements to build real time systems in Ada.

This paper discusses the types of control which are important for real time task scheduling. It briefly mentions some closely related Ada real time issues and delineates major committee and research activities in this area. Readers are reminded that although there are some issues with Ada and its real time task scheduling, Ada presents fewer problems in the development of real time systems than any other known alternative. Ada compilers are available now which can produce object code whose execution time efficiency is comparable or superior to similar programs coded in other languages. Ada is being used today to build effective real time systems. The question is which implementations of task scheduling are adequate to allow this feature of Ada to be used for NASA applications.

Overview

It is a highly desirable goal from a software engineering, full life cycle viewpoint to use Ada tasks for encapsulating units of independent, asynchronous processing. Such tasks are traditionally managed through the establishment of a time-lined sequence of task processing which is repeated on a cyclic basis or through the setting of the relative priority of tasks in an event-driven system. In both cases real time system programmers often need a fine granularity of control over the the task start time and the processing time allotted to each task. There are hard real-time scheduling constraints for both cyclically oriented and purely event driven configurations.

Ada was designed for the domain of real time embedded systems. The Ada language contains some features which had, in the past, been frequently delegated to the surrounding operating system or a custom executive environment. These features include the ability to support potentially concurrent tasks and to communicate among them. The Ada designers tried to give language vendors the greatest possible latitude in designing their implementations. Ada compilers can therefore pass stringent validation requirements, but may not contain a level of task scheduling support which is adequate for all real time applications.

Resolutions to provide better task scheduling are being considered by three major groups:

1. The Association for Computing Machinery (ACM) Special Interest Group on Ada (SIGAda) Ada Runtime Environment Working Group (ARTEWG) is defining standard compiler pragmas, features and optimizations. They are also defining standard interfaces providing application visibility into the runtime support library to control tasking behavior.

2. The Ada Joint Program Office (AJPO) is launching a project to revise the Ada Language MIL-STD-1815A. This is currently known as Ada 9X and may deal with hard language issues. This would bring control options into the language at a level of standardization above the ARTEWG.

3. Vendors are seeking to optimize their Ada Program Support Environments (APSEs) and are taking initiatives to better implement task scheduling.

Issues

Although all real time systems must provide predictable response, the precision of that response may vary widely among applications or even instances of applications. A system designed to respond to an event once a second may be able to afford the overhead of some relatively inefficient task scheduling primitives which a more demanding application could not tolerate. Most researchers and users agree that there should be a means of tailoring the real time support environment such that applications which do not require certain features need not suffer the overhead of including them in the system.

1. Issues regarding the usage of Ada tasking

General inadequacies for Ada tasking are listed below:

a. Implementation Dependency -
The Ada Language Reference Manual (LRM) leaves much latitude for scheduling tasks.

b. Priority levels -
The LRM provides only fixed priority levels.

c. Queue processing -
Entry queues are first in, first out (FIFO) only.

d. Performance -
Many APSE implementations have very general and inefficient approaches to task scheduling.

2. Features Directly Concerned with Task Scheduling

Any primitive which controls the "precise and predictable" response of a real time system is a candidate for being placed on the shopping list of task scheduling features. Those impacting task scheduling directly are:

a. Non preemptible sections -
   The ability may be needed to distinquish sections of code which will not be preempted by other tasks becoming eligible to run.
b. Dynamic Priorities -
   There may be a need to be able to change the priority of an Ada task at run time. Most implementations allow this only at compile time, or in a few systems at link time.
c. Non blocking task Input/Output (I/O) -
   Tasks performing I/O should not be allowed to block the Ada main program and/or other tasks, as this usually defeats the intent of any real time task scheduling goals.
d. Scheduling Control -
   Real time systems should be able to schedule tasks by exercising control at four critical points with a specific precision:

   Task Initiation - Start immediately, at a specified time, after a specified delay, or on a specified event.
   Task Repetition - Repeat every (time unit) or no repetition allowed.
   Task Execution - Suspend, Resume, Abort, Restart
   Task Completion - Complete at a specified time, on a specified event, or only as specified by initiation, repetition, or execution control
e. Pragmas facilitating optimization and scheduling -
   Since Ada uses the rendezvous (a task scheduling) mechanism to communicate with I/O handlers, the relative inefficiency of rendezvous on many Ada systems suggest a need for such things as "Fast Interrupts," so that well-behaved real time I/O handlers may execute without the full rendezvous overhead and at a more precise time.

3. Features Somewhat Related to Task Scheduling

Some primitives concern task scheduling indirectly. They are shown below:

a. Memory management-
Memory management, including control of dynamically allocated data structures, and pre-elaboration of important information for real time tasks is related to task scheduling in that control over these areas is required to guarantee predictability of task response. This may also include control over the memory location of program units, although this may be addressed satisfactorily by smart linkers.
b. Task Identifiers -
The ability may be needed to identify tasks uniquely, even when their Ada names may be out of the dynamic scope of the calling program unit.
c. Asynchronous communication and
d. Prioritized queues-
It is often the case that tasks are used to process information which must be conveyed efficiently between the servant and client. The Ada rendezvous requires synchronous communication. Entry to Ada tasks are on a first come first serve basis. Prioritized entry queues would provide additional (more efficient) task scheduling constructs and perhaps allow some Ada systems to meet more demanding real time constraints.

4. Implementation

There are various approaches to achieving the kind of task control required. The challenge is to implement the desired real time task scheduling features in an efficient and robust way. Three basic approaches were mentioned by the researchers. They were:

1) Don't use Ada tasks. Instead, use the facilities of a real time executive or operating system to accomplish the same logical goals.

2) Create special Ada run time environments which are designed especially for real time use.

3) Add new features and/or modify the Ada language definition to provide these real time services.

The first approach, while not recommended, is a viable option over not using Ada at all. However, this approach does not provide the software engineering aspects of an Ada program and soon should not be necessary for most real-time requirements. Dr. Richard Volz, at Texas A&M University, has done some recent benchmarking that shows new Ada compilers which can accomplish task rendezvous in 200, 75 and 35 microseconds.

The second approach provides services beyond those required in the Ada language definition and may therefore not be portable or supported by all vendors. The ARTEWG is defining a common set of services which they hope will be adopted by the major vendors.

The third approach is being examined by the planners of Ada 9X. The effort to revise the Ada language is expected to begin early in 1989 and be accomplished sometime after 1995. Upward compatibility is a foremost concern.

Possible NASA Actions

Suggestions are offered which will help NASA implement real time systems today, while protecting their long term interest in reaching a common solution to Ada task scheduling and other closely related issues. The suggestions are:

Establish single point of contact/coordination at each appropriate NASA project office for real time tasking issues. This contact should cooperate closely with a similar point of contact for all Ada run time and/or Ada real time issues.

Continue effort to collect requirements for NASA real time applications (current and planned), and determine the task scheduling needs from those requirements.

Decide on a minimal set of task scheduling (and other real time) features needed to support NASA projects.

Create NASA standard interfaces (an Ada interface Set) for these real-time features:

-- use available ARTEWG and/or Ada 9X interfaces
-- important point is ISOLATION so they can be updated later if needed
-- Implementors should care little whether done in a real time OS or by Ada run time environment, as long as interface is well-defined.

Consider a software integration environment in arriving at the set of interfaces (see McKay, 1988). This environment may simulate real time or in some instances have similar scheduling interfaces. Such consideration could make a difference in the choice of preferred interfaces.

Establish standard real time architecture(s) for popular application types such as data acquisition, processing, display. This could reduce number of different interfaces, promote reusability.

Participate/Support in Ada/9X , ARTEWG , and real time Ada evaluations at the Software Engineering Institute and other appropriate labs.

Follow/support performance benchmarking projects.

Support stronger ACVC test for run time features.

Supplement NASA interfaces for real-time features when international standards are accepted.

Research Organizations

Organizations generally concerned with Ada real-time issues and task scheduling are shown below with a point of contact for each:

International Standards Organization (ISO) Ada Rapporteur Group, formerly the Ada Language Maintenance Committee.
    John Goodenough, Chairman, Software Engineering Institute, Pittsburgh, PA.

Ada Board Task force for Ada9x.
    J. Kramer, Chairman, Institute for Defense Analysis, Alexandria, VA.

Ada Runtime Environment Working Group (ARTEWG) of SIGAda.
    M. Kamrad, Chairman, Honeywell Systems and Research Center, Minneapolis, MN.

Research Institute for Computing and Information Systems (RICIS).
    Glen Houston, Director, University of Houston at Clear Lake

SIGAda Performance Issues Working Group.
    John S. Squire, Chairman, Westinghouse, Baltimore, MD.

Individual groups of researchers and the issues they are investigating are listed in Table 1. A summary of each group follows the table according to the latest information available from the point of contact.

Table 1

List of Individual Group Activities

Group / / Issues	1a	1b	1c	1d	2a	2b	2c	2d	2e	3a	3b	3c	3d	4
ARTEWG Pat Rogers					X	X		X	X	X	X	X	X	
Ames Andy Goforth	X	X		X		X				X			X	X
Draper Roger Racine	X	X	X	X			X		X	X	X	X	X	X
General Dynamics Sally Kline	X			X					X					X
Goddard Dan Ferry	(NO RESPONSE)													
Goddard Betty Brinker	x			X					X	X				X
Goddard Daniel Roy	X			X			X							
IBM Research Doug Locke	X	X				X	X		X			X	X	X
Johnson Robert Shuler	X	X	X	X			X		X	X	X	X	X	X
Kyushu Un., Japan Jingde Cheng	X						X			X				X
Langley Paul Hayes	(NO RESPONSE)													
Marshall Larry Taormina	(NO RESPONSE)													
Marshall J. Gregory	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Ohio State P. Vishnubhotla					X	X	X	X		X		X		

Group / / Issues	1a	1b	1c	1d	2a	2b	2c	2d	2e	3a	3b	3c	3d	4
RICIS Charles McKay	X	X	X	X			X			X				X
SofTech, Inc. P.V. Raman	X	X	X	X	X	X			X					X
SEI N. Weiderman	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Texas A&M Un. Richard Volz	X			X	X	X	X	X	X		X	X	X	X
Un. of Bradford Alan Burns	X	X	X		X	X	X					X	X	X
Un. of Colorado Alain Jouchoux	(NO RESPONSE)													
Un. of York Andy Wellings	X	X			X	X		X			X	X	X	X
Virginia Tech Tim Lindquist	X		X					X		X		X	X	X

Descriptions of Individual Group Activities

Ames Research Center, Parallel Ada Research Project

Description: Evaluation of performance of parallel processes.

Most important issue(s): Performance

Additional issues: Task hierarchy, compiler validation with a run-time environment added.

Comments: Processing fairness is sensitive to the number of processors.

Documents: Report due October, 1988.

Point of Contact:

Andy Goforth
ms 244-4
Ames Research Center
NASA
Moffett Field, CA 94035
(415) 694-5000 or 6525

Charles Stark Draper Laboratory, Advanced Information Processing System (AIPS)

Description: Use of Ada in real time avionics, and on a SVID operating system.

Most important issue(s): Performance, Scheduling control

Additional issues: Distributed interprocess communication

Documents: [2], [7], [8], [16], [17]

Point of Contact: Roger Racine
555 Technology Square
Cambridge, MA 02139
(617) 258-2489
FAX (617) 258-2214

General Dynamics, San Diego, Tasking Ada Simulation Kit (TASKIT)

Description: Set of tools to use Ada as the development language for simulation.

Most important issue(s): Degree of parallel processing

Additional issues: Performance. Benchmarked on Encore.

Documents: [23], [24], [25]

Point of Contact:

Sally Kline, Program Manager.
PO Box 85808
San Diego, CA 92138
(619) 573-3763
FAX (619) 573-3161

Goddard Space Flight Center Network Control Program (NCP) (Code 250)

Description: Ada used to implement the Mission Operations and Data Systems Directorate Network (MNET) protocol on various nodes. (Project canceled)

Most important issue(s): Rendezvous, Shared Memory

Additional issues: Inefficiencies due to strict requirements of language.

Point of Contact:

Dan Ferry
5th Floor
Computer Science Corp.
8728 Colesville Rd.
Silver Springs, MD 20910
(301) 589-1545

Goddard Space Flight Center Remote Science Operations Center Project (RSOC), (CODE 520)

Description: Tests ability of Ada implementation to interface with an external program written in C. Performance data is compared between Ada and C components.

Most important issue(s): Mailboxes may be more efficient than rendezvous when using VAX VMS.

Additional issues: Pure tasking of message buffering effective,
Task priorities not effective.

Documents: [6]

Point of Contact:

Betty Brinker, APR
code 522.1
Greenbelt, MD 20771
(301) 286-3192

Goddard Space Flight Center, MSOCC Ada Compiler Benchmark Suite
(CODE 510),

Description: Support packages and test programs to asses Ada compilers.

Most important issue(s): Tasking overhead

Additional issues: PAMELA idioms overhead, I/O issues.

Documents: [11]

Point of Contact:

Daniel Roy
Ford Aerospace
7375 Executive Place
Seabrook, MD 20706

Goddard Space Flight Center, Flexible Ada Simulation Tool (FAST),

Description: Discreet event simulation language and tool.

Most important issue(s): Exception handling, control of context switching

Additional issues: Entry priorities, task activation order, I/O, dynamic strings.

Documents: [11]

Point of Contact:

Daniel Roy
Ford Aerospace
7375 Executive Place
Seabrook, MD 20706

IBM Research (with Software Engineering Institute)

Description: Building of real-time systems using the Ada tasking model

Most important issue(s): Priorities, Implementation issues

Additional issues: Rendezvous issues such as:

Delay statement
Timed entry calls
Conditional entry calls

Comment: Dynamic management of resources is generally avoided in real-time systems. Performance is not the fault of the Ada language.

Documents: [32]

Point of Contact:

Dr. C. Douglass Locke
IBM Systems Integration Division
Owego, NY 13827
(607) 751-4291

Johnson Space Center, Ada Production Rule System (APRS)

Description: A system for specifying rule based expert systems directly in Ada.

Most important issue(s): Memory Management.

Additional issues: Priority inversion

Comments: Tasking paradigms are for systems with large numbers of independent processors.

Point of Contact:

Robert Shuler
FR4
Houston, TX 77058
(713) 483-5258

Kyushu Un., Japan

Description: Study of tasking communication deadlocks in concurrent Ada programs

Most important issues: Asynchronous communication, implementations

Additional issues: Intertask communication

Comments: Tasking communication depends on run-time environment implementation.

Documents: [12]

Point of Contact: Jingde Cheng
Department of Computer Science & Communication Eng.
Kyshu University
Hakozaki
Higashi-ku
Fukuoka 812, Japan

Langley Research Center, Advanced Transport Operating System (ATOPS),

Description: Rewriting code of navigation and control system in Ada

Most important issue(s): Distributivity

Additional issues: Fault-tolerance.

Documents: [11]

Point of Contact:

Paul Hayes
ms 473
Langley Research Center
Hampton, VA 23665-5225
(804) 865-3777

Marshall Space Flight Center, Space Station Operating System Study,

Description: Ada runs comparative to FORTRAN speeds

Most important issue(s): Performance

Additional issues: Ada/FORTRAN interface

Comments: Using pragma SUPPRESS_ALL, Ada was faster.

Documents: [11]

Point of Contact:

Larry Taormina
(205) 544-3782

Marshall Space Flight Center, OMV project, Simulation for RV Transfer

Description: Investigation of memory management issues.

Most important issue(s): Memory management

Additional issues: Performance

Comments: Cannot use Ada tasking due to critical time constraints, except possibly for ground support software.

Documents: [11]

Point of Contact:

Judith Gregory
Systems Software Branch
Information & Electronics Lab
NASA Marshall Space Flight Center, Alabama 35812
(205) 544-3728

Ohio State University

Description: Object oriented parallel programming system

Most important issue(s): Scheduling control

Additional issues: Resource allocation

Comments: In this project, scheduling is a programmable activity.

Documents: [38]

Point of Contact:

Prasad Vishnubhotla
Dept. of Computer & Information Science
2036 Neil Ave. Mall
Columbus, OH 43210
(614) 292-1553
vishnu@tut.cis.ohio-state.edu

Research Institute for Computing and Information Systems (RICIS), Software Engineering Research Center (SERC), Portable Common Execution Environment (PCEE)

Description: Includes support for real-time as well as data-driven applications in a multiprogramming, distributed, heterogeneous target environment for Ada programs.

Most important issue(s): A tailorable, extendible run-time support environment and standard language interfaces and implementation dependencies.

Additional issues: System software design precluding advancing Ada architectures and plans for multiprocessor architectures.

Comments: The Ada language definition should not be too restrictive. Many issues are being answered by the ARTEWG Catalog of Interface Features and Options (CIFO).

Documents: [35]

Point of Contact:

Charles McKay, Director or
Patrick Rogers, Associate Director for Research
Software Engineering Research Center
University of Houston at Clear Lake
2700 Bay Area Blvd.
Houston, TX 77058
(713) 488-9490

SofTech, Inc. Ada-86

Description: Cross-compiler hosted on DEC VAX/VMS, targeted to 8086 and 80X86 processor-based systems.

Major issues: Priorities

Other issues: Rendezvous

Comments: System targeted to bare hardware with no programming support environment.

P. Venkat Raman, Systems Consultant
SofTech, Inc.
460 Totten Pond Rd.
Waltham, MA 02254-9197
(617) 890-6900

Texas A&M University, Performance Study

Description: Projects to study:

Benchmarks for Ada compilers
Distributed Ada translator
Comparison of Ada and Lisp benchmarks

Most important issue(s): Implementation dependency and Priorities

Additional issues:

Scheduling control issues such as placement of repetitive tasks within their cycle interval.

Distribution issues such as:

- propagation time between processors
- timed remote entry calls
- all of the issues listed from a distributed system perspective

Comments: Dr. Volz disagrees with the idea that pragmas are required in order to provide optimization. He has tested compilers that produced rendezvous in 200, 75 and 35 microseconds on IBM PC class computers.

Point of Contact:

Dr. Richard Volz (formerly at University of Michigan)
Director, Department of Computer Science
Zachry Engineering Center
Texas A&M University
College Station, Texas 77843-3112
(409) 845-8873
volz@cssun.tamu.edu

University of Colorado, Operations and Science Instrument Support (OASIS),

Description: (No response from Dr. Jouchoux

Most important issue(s): Rendezvous overhead excessive with large number of tasks.

Point of Contact:

Alain Jouchoux
Laboratory for Atmospheric and Space Physics (LASP)
5525 Central Ave.
Boulder, CO 80301
(303) 492-6792

University of York, UK

Description: Research in distribution of Ada

Most important issue(s): Synchronization primitives and priorities

Additional issues: General issues of remote interprocess communication

Comments: Appropriate and legal run-time systems can be used for task scheduling until the Ada language can be modified.

Point of Contact:

Dr. Andy Wellings
Department of Computer Science
Heslington
York, U.K.

Virginia Tech Department of Computer Science,

Description: Tasking using shared memory and fixed assignment of tasks to processors.

Most important issue(s): Queue processing

Additional issues: Timed and conditional entry calls

Point of Contact:

Dr. Tim Lindquist, project leader (now at)
Computer Science Dept.
Arizona State University
Tempe, Arizona 85287

Bibliography

[1] ACM, "A Catalog of Interface Features and Options for the Ada Runtime Environment," Association for Computing Machinery, Inc., Special Interest Group for Ada, Ada Runtime Environment Working Group, Release 2.0, December, 1987.

[2] Alonso, Ramon, Keith Dimorier, Lance Drane, "The NASA Space Station Operating Systems A White Paper (Revision 1), The Charles Stark Draper Laboratory, Inc., April 1, 1985, Final Draft October 31, 1985.

[3] Ardo, Anders, "Real-Time Efficiency of Ada in a Multiprocessor Environment," Department of Computer Engineering, University of Lund, Sweden, ACM, International Workshop on Real-Time Ada Issues, 1987.

[4] ARTEWG, "A White Paper on Ada Runtime Environment Research and Development," February 13, 1987.

[5] ARTEWG, "A Model Runtime Environment Interface for Ada-Draft," December, 1987.

[6] Brinker, Emerson, Hughes, "Preliminary Report on Ada Real-Time Evaluation," GSFC Code 522, August 28, 1986.

[7] Brown, Robert A., "Ada-100 Ada Tools Availability and Costs," The Charles Stark Draper Laboratory, Inc., February 10, 1986.

[8] Brown, Robert A., Racine, Roger, Sodano, Nancy, Whalen, Mary, "Ada Runtime Environment Study," The Charles Stark Draper Laboratory, Inc., October 1, 1986.

[9] Burns, Alan, "Scheduling Hard Real-Time Systems: A Review," Technical Note AB3.1, Draft, SERC, August 15, 1988.

[10] Burns, A., Wellings, A.J., "Real-Time Ada Issues," ACM, International Workshop on Real-time Ada Issues, 1987.

[11] Century Computing, Inc., "Ada Projects at NASA, Runtime Environment Issues, Final Report, Revision 4," September 15, 1987.

[12] Cheng, Jingde, Araki, Keijiro, and Ushijima, Kazuo, "Tasking Communication Deadlocks in Concurrent Ada Programs," Ada Letters, Sept/Oct. 1988.

[13] Cohen, Norman H., "Dependence on Ada Task Scheduling Is Not 'Erroneous'," IBM Research, Thomas J. Watson Research Center.

[14] Cornhill, D.T., Sha, L., Lehoczky, J.P., Rajkumar, R., Rokuda, H., "Limitations of Ada For Real-Time Scheduling," ACM, International Workshop on Real-Time Ada Issues, 1987.

[15] Cornhill, D. and Sha, L., "Priority Inversion in Ada or What Should be the Priority of an Ada Server Task?" Department of Computer Science, Carnegie Mellon University, and Honeywell Systems and Research Center.

[16] Dimorier, K., Drane, L., Goodman, B., Silver, L., "The NASA Space Station System Support Task White Paper," The Charles Stark Draper Laboratory, Inc., January 16, 1986.

[17] Dimorier, K., Drane, L., Whalen, M., "Software Support Environment System Software Studies - Final Report," The Charles Stark Draper Laboratory, Inc., October 1, 1986.

[18] Dubois, Michel and Scheurich, Christoph, and Briggs, Faye A., "Synchronization, Coherence, and Event Ordering in Multiprocessors," IEEE Computer, February, 1988.

[19] Fainter, Robert G., Moerdyk, Alan A., Lindquist, Timothy E., "An Investigation of Run Time Performance."

[20] Faulk, Stuart R. and Parnas, David L., "On Synchronization in Hard-Real-Time Systems," Communications of the ACM, Volume 31 Number 3, March 1988, pp. 274-287.

[21] Felsinger, Richard C., "Integrating Object Oriented Design, Structured Analysis/Structured Design, and Ada."

[22] Frankel, Gary, "Improving Ada Tasking Performance," ACM, International Workshop on Real-Time Ada Issues, 1987.

[23] General Dynamics, "Software User's Manual for the Tasking Ada Simulation KIT (TASKIT)," March 18, 1988.

[24] General Dynamics, "Tasking Ada Simulation Kit (TASKIT)."

[25] General Dynamics, "TASKIT Summary."

[26] Gilles, Jeff and Ford, Ray, "A Window Based Visual Debugger for a Real Time Ada Tasking Environment," ACM, Proceedings of the Fifth Washington Ada Symposium, June 27-30, 1988, pp. 59-67.

[27] Gonzalez, Dean W., "An Ada Tasking Demo," Ada Letters, Sept/Oct. 1988, pp. 87-91.

[28] Hood, Philip, "Deadlines and Throughput in Real Time Systems," SofTech, Inc., July 3, 1985.

[29] Hood, Philip and Grover, Vinod, "Designing Real time Systems in Ada - Final Report," SofTech, Inc., January 8, 1986.

[30] Howes, Norman R. and Weaver, Alfred C., "On Implementing the OSI Model in Ada Via Tasking," ACM, Proceedings of the Fifth Washington Ada Symposium, June 27-30, 1988, pp. 103-108.

[31] Lindquist, Timothy E. and Joyce, Richard C., "Ada Task Synchronization in a Multiprocessor System with Shared Memory," Journal of Pascal, Ada, & Modula-2, January/February, 1985, pp. 9-19.

[32] Locke, C. Douglass and Vogel, David R., "Problems in Ada Runtime Task Scheduling," ACM, International Workshop on Real-Time Ada Issues, 1987, pp. 51-53.

[33] Malec, Carl, "Interrupts," Boeing Advanced Systems Co.

[34] McCormick, Frank, "Scheduling Difficulties of Ada in the Hard Real-Time Environment," ACM, International Workshop on Real-Time Ada Issues, 1987, pp. 49-50.

[35] McKay, Charles W., "Conceptual and Implementation Models Which Support Life Cycle Reusability of Processes and Products in Computer Systems and Software Engineering," Software Engineering Research Center High Technologies Lab, UHCL, October 1988.

[36] Seigel, Harold, "Implementing Tasking on UNIX," SofTech, Inc., Waltham, Mass.

[37] Shumate, Ken, "An Example Case Study on Ada Tasking," Huges Aircraft Co., San Diego, CA.

[38] Vishnubhotla, Prasad, "Issues in Implementing Ada on Multiprocessors," ACM, Proceedings of the Fifth Washington Ada Symposium, June 27-30, 1988, pp. 45-48.

[39] Westinghouse Command and Control Processing, "A Process Limited by Contention for a Software Structure (Beacon Target Extraction)."