

# ADA USER JOURNAL

Volume 24

Number 2

June 2003

---

## Contents

	<i>Page</i>
Editorial Policy for <i>Ada User Journal</i>	66
Editorial	67
News	68
Conference Calendar	109
Forthcoming Events	114
Articles	
S Tucker Taft	
“ <i>Object-Oriented Programming Enhancements in Ada 200Y</i> ”	119
Ada-Europe 2002 Sponsors	128
Ada-Europe Associate Members (National Ada Organizations)	Inside Back Cover

# Editorial Policy for *Ada User Journal*

## Publication

*Ada User Journal* – The Journal for the international Ada Community – is published by Ada-Europe. It appears four times a year, on the last days of March, June, September and December. Copy date is the first of the month of publication.

## Aims

*Ada User Journal* aims to inform readers of developments in the Ada programming language and its use, general Ada-related software engineering issues and Ada-related activities in Europe and other parts of the world. The language of the journal is English.

Although the title of the Journal refers to the Ada language, any related topics are welcome. In particular papers in any of the areas related to reliable software technologies.

The Journal publishes the following types of material:

- Refereed original articles on technical matters concerning Ada and related topics.
- News and miscellany of interest to the Ada community.
- Reprints of articles published elsewhere that deserve a wider audience.
- Commentaries on matters relating to Ada and software engineering.
- Announcements and reports of conferences and workshops.
- Reviews of publications in the field of software engineering.
- Announcements regarding standards concerning Ada.

Further details on our approach to these are given below.

## Original Papers

Manuscripts should be submitted in accordance with the submission guidelines (below).

All original technical contributions are submitted to refereeing by at least two people. Names of referees will be kept confidential, but their comments will be relayed to the authors at the discretion of the Editor.

The first named author will receive a complimentary copy of the issue of the Journal in which their paper appears.

By submitting a manuscript, authors grant Ada-Europe an unlimited license to publish (and, if appropriate, republish) it, if and when the article is accepted for publication. We do not require that authors assign copyright to the Journal.

Unless the authors state explicitly otherwise, submission of an article is taken to imply that it represents original, unpublished work, not under consideration for publication elsewhere.

## News and Product Announcements

*Ada User Journal* is one of the ways in which people find out what is going on in the Ada community. Since not all of our readers have access to resources such as the World Wide Web and Usenet, or have enough time to search through the information that can be found in those resources, we reprint or report on items that may be of interest to them.

## Reprinted Articles

While original material is our first priority, we are willing to reprint (with the permission of the copyright holder) material previously submitted elsewhere if it is appropriate to give it a wider audience. This includes papers published in North America that are not easily available in Europe.

We have a reciprocal approach in granting permission for other publications to reprint papers originally published in *Ada User Journal*.

## Commentaries

We publish commentaries on Ada and software engineering topics. These may represent the views either of individuals or of organisations. Such articles can be of any length – inclusion is at the discretion of the Editor.

Opinions expressed within the *Ada User Journal* do not necessarily represent the views of the Editor, Ada-Europe or its directors.

## Announcements and Reports

We are happy to publicise and report on events that may be of interest to our readers.

## Reviews

Inclusion of any review in the Journal is at the discretion of the Editor. A reviewer will be selected by the Editor to review any book or other publication sent to us. We are also prepared to print reviews submitted from elsewhere at the discretion of the Editor.

## Submission Guidelines

All material for publication should be sent to the Editor, preferably in electronic format. The Editor will only accept typed manuscripts by prior arrangement.

Prospective authors are encouraged to contact the Editor by email to determine the best format for submission. Contact details can be found near the front of each edition. Example papers conforming to formatting requirements as well as some word processor templates are available from the editor. There is no limitation on the length of papers, though a paper longer than 10,000 words would be regarded as exceptional.

# Editorial

In this issue of the Ada User Journal we continue our close watch on the on-going Ada language revision process.

Experience with Ada 95 has brought about the urge to push the language forward along the two main avenues of its advanced support: high-integrity real-time programming and object-oriented programming (OOP). After covering the former topic, we now turn our attention to the latter and have the pleasure of hosting S Tucker Taft, who illustrates for us the rationale and the outline of the proposed completion of the OOP language support in Ada 200Y.

Our readers should know that the editing of the News section is by far the most effort-intensive part of the volunteer-based production of the Ada User Journal. Interestingly, this is so because of the health of the language user community and the impressive wealth of information that flows in the various channels monitored by our News editor. Over the last year we have seen the News section often account for about half of every 64-page issue. The feedback we receive suggests that our readers find the News section useful and informative. They should as well know that the section contents are only a small fragment (but hopefully a representative overview) of a much larger, some 30 times larger, material, which is to be logged, linked and excerpted. All in all a major effort, which may sometimes take longer than anticipated. Please, bear with us while we continue this effort to produce an enjoyable and timely journal.

*Tullio Vardanega*

*Padova*

*June 2003*

*Email: [tullio.vardanega@math.unipd.it](mailto:tullio.vardanega@math.unipd.it)*

# News

*Dirk Craeynest (ed)*

*Offis nv/sa and K.U.Leuven. Email Dirk.Craeynest@offis.be*

---

## Contents

	page
Ada-related Organizations	68
Ada-related Events	68
Ada and Education	71
Ada-related Resources	72
Ada-related Tools	75
Ada-related Products	82
Ada and Linux	87
Ada and Microsoft	88
References to Publications	89
Ada Inside	90
Ada in Context	97

---

## Ada-related Organizations

### ACT Europe Joins Ada Resource Association

*URL: <http://www.adaic.org/news/acte-ara.html>*

*Subject: ARA News Release*

Ada Resource Association Adds ACT Europe to Global Membership

Salt Lake City -- Software Technology Conference (April 30, 2003) --

The Ada Resource Association (ARA) is pleased to announce the addition of its newest member, ACT Europe, which develops components of the GNAT Pro technology for the Ada programming language.

Founded and led by Cyrille Comar and Franco Gasperoni, ACT Europe is the Paris-based sister company to the long-standing ARA member Ada Core Technologies. ACT Europe has played a major role in advancing Ada and GNAT Pro usage in the European software industry.

"ACT Europe represents a broad base of Ada developers and embedded systems users," said S. Tucker Taft, president of both the ARA and of SofCheck, Inc.

"They will add significantly to the ARA's effectiveness in serving the language's international community."

"We fully support the ARA's objectives," Dr. Gasperoni said. "Activities such as the careful evolution of the Ada language, publication of Ada successes, and maintenance of an effective Ada information web site are very important to the language's growth. By joining the ARA we hope to bring fresh ideas and a European perspective to these activities."

A leading European developer of Ada 95 technology, ACT Europe offers native, embedded and safety-critical Ada solutions on a large variety of systems. The company specializes in project support and consulting for users of GNAT Pro Development Environment for Ada 95, a programming language designed for embedded systems. ACT Europe also develops components of GNAT Pro technology including the GtkAda graphical toolkit, the new GNAT Programming System (GPS) IDE, and the GNAT Pro High Integrity Edition.

With members that span the globe, the ARA is a trade association of companies whose products and services represent more than 90 percent of the Ada development tools marketplace.

For more information on Ada and on ACT Europe, please see <http://www.adaic.org>. Or write to Ann Brandon, Communications Director, Ada Resource Association, [abrandon@sover.net](mailto:abrandon@sover.net).

---

## Ada-related Events

[The announcements reported below are a selection of the many Ada-related events organized by local groups. If you are organizing such an event, feel free to inform us as soon as possible. If you attended one please consider writing a small report for the Journal. -- dc]

### Apr 8 - ACM Baltimore/DC SIGAda Meeting [...]

*From: Currie Colket <colket@mitre.org>*  
*Date: Wed, 26 Mar 2003 14:55:07 -0500*  
*Organization: The MITRE Corporation*  
*Subject: Team-Ada: High Reliability, Safety, and Secure Solutions for Embedded Developers Presentation on 8 April*  
*To: team-ada@aem.org*

Our Next Meeting is scheduled for Tuesday, 8 April 2003, as a Joint Meeting of the Baltimore and DC SIGAda Chapters. John Warther and Gordon Uchenick will be speaking on "High Reliability, Safety, and Secure Solutions for Embedded Developers". [and using this as a basis to discuss Wind River's Trusted/Safety Critical VxWorks] [...]

Abstract: The advent of security requirements affects everyone involved in the research, development, design, fabrication and production of DOD systems that carry electronic information. To ensure our DOD Aerospace and Defense customers have a roadmap to

meet these requirements, Wind River is investing in technology that will meet the demands of IA Directive 8500.2. [...]

Embedded Systems and Real Time systems are all an important part of the military infrastructure and will be a foundation for our future infrastructure. Wind River will present their High Reliability, Safety and Secure Solutions for Embedded Developers. They will discuss how their architecture strategy addresses those programs that must meet high integrity, safety-critical, and high-assurance requirements for the Ada 95, C, and C++ programming languages.

The AE 653 operating system/software engineering environment is utilized for avionics applications and many other applications requiring temporal/spatial separation. It serves as a baseline for Wind Rivers' work in the formal methods arena supporting the Common Criteria Evaluation Assurance Level (EAL) / Multi Level Security (MLS) and Multiple Levels of Independent Security (MILS) requirements instrumental in Communications Security (COMSEC), Avionics, Unmanned Aerial Vehicles (UAV), and other mission critical applications. [...]

Clyde Roby's Presentation Available

At the DC SIGAda meeting on 9 January 2003, Clyde Roby gave an excellent presentation titled: Evolving Ada Bindings and Ada APIs. [See also "Jan 9" topic in AUJ 24-1 (Mar 2003), p.6. -- dc] Slides from his presentation are available online as a Powerpoint Presentation at [http://www.acm.org/sigada/locals/dc/2003\\_01\\_APIWG-Presentation.ppt](http://www.acm.org/sigada/locals/dc/2003_01_APIWG-Presentation.ppt) (ppt, 276KB). [...]

Jeff Castellow, Chair, DC SIGAda

### Apr 9-10 - Ada UK User Group Spring Conference

*From: John Robinson <John@JohnRobinsonAndAssociates.com>*  
*Date: Thu, 20 Feb 2003 21:49:36 -0000*  
*Organization: John Robinson And Associates Ltd*  
*Subject: CNF: Ada 0Y Workshop & Spring Ada UK Conference*  
*Newsgroups: comp.lang.ada*

The Spring Ada UK User Group conference is set for April 9th 2003. This will be followed by a post-conference workshop on Ada 0Y on April 10th.

To be held in Swindon (UK), the conference is a two stream event with material on requirements management,

UML, static code analysis, the future of IDEs, Ada 95 and more. The Ada 0Y workshop, lead by Tucker Taft, is a valuable opportunity for delegates to provide feedback into the design of the next version of the Ada language.

Ada users may also wish to note that a sister community to Ada UK has been launched. The "DSP Forum" will provide DSP users with a community similar to Ada UK, which is now the largest community of its type in Europe. The inaugural conference for the DSP Forum will be held on April 10th and will have sessions on RapidIO, StarFabric, FPGAs, real-time OS for DSP applications and a panel session on interconnect strategies. [...]

[www.DSPForum.org.uk](http://www.DSPForum.org.uk),  
[www.AdaUK.org.uk](http://www.AdaUK.org.uk),  
[www.EmbeddedSystemsClub.com](http://www.EmbeddedSystemsClub.com)

John Robinson, Conference Director

*From: John Robinson  
 <John@JohnRobinsonAndAssociates.com>  
 Date: Thu, 6 Mar 2003 16:06:20 -0000  
 Subject: ANN: Ada UK User Group Spring Conference  
 Newsgroups: comp.lang.ada*

A brochure and registration form for the Spring 2003 Ada UK/Embedded Systems Club Conference can now be found at <http://www.AdaUK.org.uk>.

[...] The Ada UK conference itself offers the following presentations:

"Latest Trends in IDE Development", Arnaud Charlet and Franco Gasperoni, ACT Europe

"Integrated Modular Avionics Application Development Case Study Using VxWorks AE653 and GNAT", Paul Parkinson, Wind River and Franco Gasperoni, ACT Europe

"Requirements: If you don't know where you are going you'll probably end up somewhere else", Glennan Carnie, Flight Refuelling Ltd (Digital Systems)

"Programmers Dozen: Thirteen Recommendations for Refactoring, Repairing and Regaining Control of Your Code", Kevlin Henney, Curbralan Ltd

"Industrial Strength Exception Freedom", Dr Rod Chapman, Praxis Critical Systems Ltd

"Fixing Software Before It Breaks", Tucker Taft, SofCheck Inc

"Abstract Interpretation Applied to Static Verification", Dr Alain Deutsch, PolySpace Technologies

There are also two Panel Sessions on "Development Trends" and "Status and Future of Static Code Analysis" [...]

*From: John Robinson  
 <John@JohnRobinsonAndAssociates.com>  
 Date: Tue, 25 Mar 2003 14:39:45 -0000*

*Subject: CNF: Abstracts now available for Embedded Systems Club, Ada UK and DSP Forum conference sessions  
 To: <news@adaxia.com>*

Abstracts for all presentations at the Embedded Systems Club, Ada UK User Group and DSP Forum conference sessions are now available at [URL's above --dc]. [...]

*From: John Robinson  
 <John@JohnRobinsonAndAssociates.com>  
 Date: Wed, 2 Apr 2003 00:50:59 +0100  
 Subject: CNF: Embedded Systems Club Sponsor & Partner Announcements  
 To: <News@Adaxia.com>*

Next week the Embedded Systems Club will host two essential events for embedded systems practitioners: April 9th - Embedded Systems Club & Ada UK User Group Conference; April 10th - DSP Forum Inaugural Conference. Both these events have exhibition areas in which you have the opportunity to meet our sponsors and partner companies. The exhibition is open to all registered delegates. [...]

Our sponsors and partners will be making the following announcements during these events:

ACT Europe will be announcing GPS, the GNAT Programming System IDE. The GPS Ada IDE integrates GNAT Pro tools in a single visual development environment. Designed by programmers for programmers, GPS is a new kind of IDE that offers the experience of designing software in a uniquely comfortable environment.

Aonix. The former Critical Development Solutions (CDS) Division of Aonix has divested from its parent to become an independent company. This retains the Aonix name and all CDS products. With strong commitment and investment, Aonix continues to address the Software Development tools and service requirements of its existing technical/mission/safety critical markets.

First Matrix. The SCORE\* Ada 95 / EC++ IDE and CsLEOS\* ARINC 653 compliant RTOS have been integrated for high-integrity embedded applications. CsLEOS\*, manufactured by BAE SYSTEM Controls, is already certified to FAA DO-178B level A. The combination provides a multi-language capability for ARINC 653 applications.

Green Hills Software announces FAA acceptance of a DO-178B, Level A, certification package for its INTEGRITY-178B RTOS. The INTEGRITY certification package was delivered to Rockwell Collins for use in Technical Standards Order certification for a new avionics system aboard the Sikorsky S-92 helicopter.

I-Logix provides enterprise solutions for real-time embedded applications development and product management. I-

Logix solutions significantly compress systems and software development cycles while improving product quality. These products allow engineers to graphically model the behavior and functionality of their embedded systems, analyze and validate the system and automatically generate production quality code in a variety of languages.

IPL is happy to announce that a new version of AdaTEST 95 is well-advanced in the planning and should be ready later this Spring. Cantata++ has recently been ported to Wind River's product 'Tornado' (V2). IPL will be jointly presenting some seminars on high-integrity real-time development with I-Logix and Green Hills. These will take place on April 29 (Reading) and May 1 (Derby).

John Robinson & Associates have been contracted to supply the RAF's Harrier Software Maintenance Unit with Ada language training, extending an 18-year relationship with the RAF. Tom Clackson, a former team leader on the Harrier II Mission Computer and head of the JR&A Ada team, will deliver the training.

Mercury Computer Systems. EDN Magazine listed Mercury's VantageRT 7410 PCI multiprocessor module as one of the top products of 2002. VantageRT modules can be configured to create systems of up to 60 PowerPC 7410 processors. These systems deliver a level of performance unsurpassed in the PCI form factor.

Objektum. Over the past 3 years Objektum has worked closely with BAE SYSTEMS on the Harrier GR7 upgrade, training and mentoring key personnel to Ada95, UML and the mapping of UML to Ada95. More recently, Objektum has specified a testing strategy and environment implemented in UML and Ada95, supported by AdaTEST.

Praxis Critical Systems are pleased to announce the availability of the Sample Chapters of the new book "High Integrity Software: the SPARK Approach to Safety and Security" by John Barnes. The Sample Chapters are available in PDF on [www.sparkada.com](http://www.sparkada.com).

TNI Europe, the leading supplier of HOOD method toolsets for OO embedded software development offers the following new releases. STOOD 4.3 on UNIX and Windows with Ada and C reverse engineering – available now. CP HOOD 5.5 on UNIX with upgraded editing, import/export, configuration and build capability - available May 2003. REQIFY 1.3 on UNIX and Windows for Requirements capture and tracability - available now.

Wind River. The world's largest offshore wind farm is to enter commercial operation. Located at Horns Rev, an aquatic reef 20 kilometres off the west

coast of Denmark in the North Sea, each wind turbine has Wind River technology controlling every aspect of the various regulation, monitoring, data collection and control tasks.

## Apr 23-25 - ACM SIGAda at Embedded Systems Conf.

From: Richard Riehle  
<richard@adaworks.com>  
Date: Thu, 24 Apr 2003 17:07:19 -0700  
Organization: AdaWorks Software Engineering  
Subject: Embedded Systems Conference  
Newsgroups: comp.lang.ada

It was nice to see the SIGAda booth at the Embedded Systems Conference this week in San Francisco. [...] Some interesting and interested people came by to ask questions. Kudos to the SIGAda and ARA people for funding this on-going effort.

## May 8 - ACM Baltimore/DC SIGAda Meeting [...]

From: Currie Colket <colket@mitre.org>  
Date: Wed, 30 Apr 2003 16:52:32 -0400  
Organization: The MITRE Corporation  
Subject: Team-Ada: SIGAda DC Meeting on Thursday, 8 May on OCR of Cryptographic Source Code  
To: team-ada@acm.org

Thursday, 8 May 2003, [...] at the new MITRE2 Building in McLean, Virginia, DC SIGAda and Baltimore SIGAda will feature the following presentation: Optical Character Recognition (OCR) of Cryptographic Source Code

Abstract: This paper describes a case study evaluating the efficacy of utilizing the published, printed copy of cryptographic source codes (Pretty Good Privacy) to reproduce the electronic equivalent source code using Optical Character Recognition (OCR) solutions. Accuracy measurements of the resulting OCR outputs at various scanning resolutions and estimates of the additional effort required to correct the output are provided for approximately 100 pages of training material. Correlation of these estimates against one of six volumes of the source code is provided.

You're probably wondering where Ada comes in, I suppose - the whole thing is written in Ada... The presentation will include the design decisions made throughout the development. The rationale for these design decisions could be applicable and valuable to other software developments.

Presenter: Karl Nyberg [...]

Slides From John Warther's and Gordon Uchenick's Presentation Available. [See also "Apr 8" topic above. -- dc] Slides from this presentation are available online as a Powerpoint Presentation at <http://www.acm.org/sigada/locals/dc/2003>

04\_Gordon\_Uchenick.ppt (ppt, 2.097 MB). [...]

Jeff Castellow, Chair, DC SIGAda

## Jun 16-20 - Ada-Europe 2003 Conference

From: dirk@ada.cs.kuleuven.ac.be (Dirk Craeynest)  
Date: 13 Apr 2003 20:52:47 +0200  
Organization: Ada-Europe, c/o Dept. of Computer Science, K.U.Leuven  
Subject: 8th Int.Conf.on Reliable Software Technologies, Ada-Europe'2003  
Newsgroups: comp.lang.ada,fr.comp.lang.ada

Call for Participation - Program Summary  
8th International Conference on Reliable Software Technologies - Ada-Europe 2003, 16-20 June 2003, Toulouse, France  
<http://www.ada-europe.org/conference2003.html>

Organized by CS, UPS and ONERA. Sponsored by Ada-Europe. In cooperation with ACM SIGAda, Ada-France, IRIT, Springer and IUP ISI

Ada-Europe organizes annual international conferences since the early 80's. This is the 8th event in the Reliable Software Technologies series, previous ones being held at Montreux, Switzerland ('96), London, UK ('97), Uppsala, Sweden ('98), Santander, Spain ('99), Potsdam, Germany ('00), Leuven, Belgium ('01), Vienna, Austria ('02).

[...] Quick overview: Mon 16 & Fri 20 tutorials and workshop; Tue 17 - Thu 19 paper and vendor presentation sessions, exhibition.

Program co-chairs: Jean-Pierre Rosen, Adalog, Arcueil, France, [Rosen@adalog.fr](mailto:Rosen@adalog.fr); Alfred Strohmeier, Swiss Fed. Inst. of Technology Lausanne (EPFL), Software Engineering Lab, Switzerland, [Alfred.Strohmeier@epfl.ch](mailto:Alfred.Strohmeier@epfl.ch).

Invited speakers: Pascal Leroy, Rational Software Corporation, France "An invitation to Ada 2005"; Mira Mezini, Darmstadt University of Technology, Germany "Aspect-Oriented Programming Beyond Hierarchical Modularity" Jörg Kienzle, McGill University, Canada; "Software Fault Tolerance: An Overview".

Tutorials (full day): "The Personal Software Process for Ada", Daniel Roy; "Developing High Integrity Systems with GNAT/ORK", Juan Antonio de la Puente & Juan Zamorano.

Tutorials (half day): "Implementing Design Patterns in Ada95", Matthew Heaney; "Principles of Physical Software Design with Ada95", Matthew Heaney; "High-integrity Ravenscar Using SPARK", Peter Amey; "Architecture Centric Development Using Ada and the Avionics Architecture Description

Language", Bruce Lewis & Edward Colbert; "A Semi Formal Approach to Software System Development", William Bail; "An Overview of Statistical-Based Testing", William Bail.

Workshop: "Quality of Service in Component-Based Software Engineering (QoS in CBSE 2003)"; chair Jean-Michel Bruel, [bruel@univ-pau.fr](mailto:bruel@univ-pau.fr).

Papers: 29 papers on Ravenscar, Static Analysis, Language Issues, Distributed Information Systems, Software Components, Formal Specification, Metrics, Real-Time Kernels, Real-Time Systems Design, and Testing; authors from 14 countries: Australia, Austria, Belgium, Finland, France, Germany, Hong-Kong, India, Israel, Italy, Portugal, Spain, United Kingdom, and USA.

Exhibition: 10 exhibitors already committed: ACT Europe, Adalog, Aonix, C-S, Green Hills, I-Logix, IPL, Praxis Critical Systems, Rational Software, and TNI-Europe, [...]; separate vendor presentation tracks for exhibitors.

Social evening events: [...].

Registration: includes copy of full proceedings, published by Springer-Verlag in Lecture Notes in Computer Science series (LNCS) [...]

Dirk.Craeynest@cs.kuleuven.ac.be, Ada-Europe'2003 Publicity Co-Chair

## Dec 7-11 - ACM SIGAda 2003 Conference

From: Sward Ricky E Lt Col USAFA/DFCS  
<Ricky.Sward@usafa.af.mil>  
Date: Mon, 14 Apr 2003 15:06:31 -0600  
Subject: SIGAda 2003 CFP  
To: SIGAda-announce@acm.org

Call for Participation - SIGAda 2003, 7-11 December 2003, San Diego, California, USA. Sponsored by ACM SIGAda [...]

Constructing highly reliable software is an engineering challenge that can now be met in many domains. The application of software engineering methods, tools, and languages interrelate to make the challenge easier or more difficult. This conference focuses on the interaction between these three aspects of software engineering, especially how features in Ada have and will permit the development of the tools and methods that result in correctness, reliability, and quality of the resulting software. Especially welcome are papers that analyze Ada with respect to these factors or in comparison with other languages. This conference will gather industrial experts, educators, software engineers, and researchers interested in developing, analyzing, and certifying reliable, cost-effective software. Technical or theoretical papers as well as experience reports with a focus on Ada are solicited.

See the SIGAda 2003 Home Page for the complete, printable Call for Participation: <http://www.acm.org/sigada/conf/sigada2003>. [...]

Please submit questions on the conference to the Conference Chair, Robert C. Leif [rleif@rleif.com](mailto:rleif@rleif.com) <<mailto:rleif@rleif.com>> or the Program Chair, Ricky E. Sward [rsward@usafa.edu](mailto:rsward@usafa.edu)

---

## Ada and Education

### Course and Training Material in French

*From: daneel*

<[supercaml.news@web2news.net](mailto:supercaml.news@web2news.net)>

*Date: Thu, 20 Feb 2003 23:03:37 +0100*

*Subject: Re: Tutoriel Ada pour débutant*

*Newsgroups: fr.comp.lang.ada*

[Extracts translated from French. Previously included pointers are not repeated here; see also similar topics in AUJ 24-1 (Mar 2003), p.7, AUJ 23-4 (Dec 2002), p.191, and AUJ 23-3 (Sep 2002), p.128. -- dc]

> Do some of you know where I can find a document in French, easy and clear, to start programming in Ada?

<http://www.lifl.fr/~touzet/ADA/> [...] but also <http://www.google.fr> and select the button "Pages francophones".

*From: Xavier Sautejeau*

<[xavier\\_sautejeau@yahoo.fr](mailto:xavier_sautejeau@yahoo.fr)>

*Date: Fri, 7 Mar 2003 12:00:19 +0100*

*Subject: Enseignement public de Ada*

*To: ada-france@ada-france.org*

I just wanted to inform about this URL proposing an Ada course: <http://www.laas.fr/~francois/INSA/algopr og/algoprog.html>

### Ada Education in France

*From: Daniel Feneuille*

<[feneuille@romarin.univ-aix.fr](mailto:feneuille@romarin.univ-aix.fr)>

*Date: Thu, 10 Apr 2003 20:03:51 +0200*

*Organization: IUT Aix*

*Subject: enseignement de Ada*

*To: Ada France <[ada-france@ada-france.org](mailto:ada-france@ada-france.org)>*

[Extracts translated from French. -- dc]

I intend to write a paper emphasizing the educational virtues of Ada. I teach it since 1988 and I am obviously very satisfied of it (that will not astonish the readers of this list!).

I'd like to make an overview of the various teaching units where Ada is still appreciated [...]. Anybody has an idea about what happens abroad (for Switzerland there's a lot, isn't there Pierre Breguet or Luigi Zaffalon!) but in the USA for example? [...]

*From: Laurens Andre*

<[Andre.Laurens@cnes.fr](mailto:Andre.Laurens@cnes.fr)>

*Date: Fri, 11 Apr 2003 12:19:31 +0200*

*Subject: RE: enseignement de Ada*

*To: Ada France <[ada-france@ada-france.org](mailto:ada-france@ada-france.org)>*

[...] in Toulouse, the Université Paul Sabatier is very active [...]. UPS, apparently, focusses on software engineering: they not only cover the language, but they integrate it very early into the development process by approaching in parallel also design methods, and in particular HOOD, which has roots in Toulouse.

Another aspect which can be interesting for your inventory is to look at in-house training given in companies to their developers.

For what I know in Toulouse, the aircraft and space industry got busy [...]. CNES, of which I am part, organized a few years ago a whole schedule of internal training (in collaboration with the UPS), and continues to do that in a sporadic way, upon request [...]. There too, the attention to software engineering is very much present (a frequent HOOD and Ada association). [...]

André Laurens, Nacelles Pointées, Centre National d'Etudes Spatiales, 31401 Toulouse, [http://ballons.cnes.fr/nacelles\\_pointees](http://ballons.cnes.fr/nacelles_pointees)

*From: aymeric@nmx.com*

*Date: Fri, 11 Apr 2003 15:23:08 +0200*

*Subject: ada en enseignement*

*To: ada-france@ada-france.org*

For North America I don't know, but for Montreal, I can answer. Ada is taught at the UQAM (Université du Québec à Montréal) for the introductory courses to "informatics", simple data structures (stack, etc.) and the algorithms (sorting...). Philippe Gabrini, author of a book on Ada, is a teacher at the UQAM. The book comes from the notes of course.

*From: Pierre Breguet*

<[pierre.breguet@eivd.ch](mailto:pierre.breguet@eivd.ch)>

*Date: Tue, 15 Apr 2003 10:20:40 +0200*

*Subject: Re: enseignement de Ada*

*cc: Ada France <[ada-france@ada-france.org](mailto:ada-france@ada-france.org)>*

Here in French-speaking Switzerland, Ada is taught in three HES-SO institutes (Haute école spécialisée de Suisse occidentale): the écoles d'ingénieurs in Genève (EIG), in the canton of Vaud à Yverdon-the-Baths (EIVD) and, partially, the écoles d'ingénieurs de l'Arc Jurassien (EIAJ).

Each year, more than two hundred qualified engineers of HES-SO thus followed a training in informatics implying the use of Ada [...]. But it is necessary to resist certain pressures, as always...

Prof. Pierre Breguet, EIVD (Ecole d'Ingénieurs du canton de Vaud), 1401 Yverdon-les-Bains, Switzerland

*From: Laurent Pautet <[pautet@inf.enst.fr](mailto:pautet@inf.enst.fr)>*

*Date: Wed, 23 Apr 2003 11:18:50 +0200*

*Subject: Re: enseignement de Ada*

Ada is taught at Telecom Paris since a long time. In a course of programming OO. Ada is also taught in the courses on real-time systems and those on parallelism as well as the courses on distributed systems. There are also many student projects and training courses in Ada (which have strongly contributed to the realization of free software like GLADE, AdaBroker, PolyORB).

[...] Paris VI has a course of programming in Ada [...] (ask F. Kordon).

*From: Julien Klein*

<[julien\\_klein@yahoo.fr](mailto:julien_klein@yahoo.fr)>

*Date: Thu, 24 Apr 2003 09:47:45 +0200*

*Subject: Re: enseignement de Ada*

I did not follow the whole thread, but I just send a small mail to announce that Ada is taught at the ENSMA of Poitiers. For the students in the first year of engineering, the Ada83 language is used as application language during scientific programming. The Ada95 language is used by the engineering students in their third year [among others in] projects relating to the "Object-Oriented Programming" and "Real-Time Systems" courses. This was still the case last year.

Julien Klein, Ingenieur ENSMA, promo 2001, Consultant AONIX

*From: michel.gauthier@unilim.fr (Michel Gauthier)*

*Date: Mon, 28 Apr 2003 16:24:56 +0100*

*Subject: Re: enseignement de Ada*

*To: ada-france@ada-france.org*

The "licence d'informatique" of the university of Limoges uses Ada as supporting language of the course on reuse with as argumentation that the other languages do not provide tools of sufficient power and flexibility for the implementation of the concepts [...].

It should be noted that reuse is seen on itself, independant of the object concept, in the sense that one can use objects without reuse and that reuse doesn't require objects. [...]

Michel Gauthier, Laboratoire d'informatique, F-87000 Limoges

### Ada Education in Norway

*From: Rune Winther*

<[rune.winther@hiof.no](mailto:rune.winther@hiof.no)>

*Date: Mon, 28 Apr 2003 09:38:32 +0200*

*Subject: Affordable Ada compiler for Motorola 68332?*

*Newsgroups: comp.lang.ada*

I'm considering using a Motorola 68332 based controller (Eyebot) in student projects, and I am therefore in need of a suitable Ada compiler.

If anyone can help me find a usable and cheap (academic/educational license?) compiler I would really appreciate this.

[...] For simplicity I would prefer a compiler that could be run on a PC under Windows or Linux.

More information on the Eyebot controller can be found at <http://www.ee.uwa.edu.au/~braunl/eyebot>

Rune Winther, Ph.D., Ostfold College, Norway

*From: Steve <steved94@atbi.com>  
Date: Mon, 28 Apr 2003 13:27:39 GMT  
Subject: Re: Affordable Ada compiler for  
Motorola 68332?  
Newsgroups: comp.lang.ada*

Take a look at RTEMS. <http://www.rtems.com>. It's free (without support). Support is available. [...]

[lsiever6619@yahoo.com (L. Siever) also responded: -- dc]

Looks like MaRTE(a RTOS) has some support for the 68332: [http://marte.unican.es/contents.html#current\\_developments](http://marte.unican.es/contents.html#current_developments)

## Ada Training

[This information is included as examples of public Ada training courses: many are being organized regularly. For more, see also pointers in several previous AUJ issues. -- dc]

*From: Ed Colbert <colbert@abssw.com>  
Date: Sat, 19 Apr 2003 12:22:48 -0700  
Subject: [Reminder] Public Intro & Real-Time Ada 95 Classes During May 2003 in Carlsbad CA  
To: team-ada@acm.org*

Absolute Software will be holding the following public Ada programming courses in Carlsbad, CA, during the month of May 2003.

12-16 May: "Ada 95 for Experienced Programmers" (Introductory).

20-23 May: "Developing Real-Time Systems in Ada 95".

You can find a full description and registration form on our web-site, [www.abssw.com](http://www.abssw.com). [...] (We also offer courses on object-oriented methods and other object-oriented languages.) [...]

---

## Ada-related Resources

### New Interview Section on Ada-France Web Site

*From: Samuel Tardieu <sam@ada-france.org>  
Date: Fri, 21 Mar 2003 15:23:46 +0100  
Subject: Inauguration de la nouvelle rubrique Entretien  
To: ada-france@ada-france.org*

[Translated from French: -- dc]

The Ada-France site inaugurates today a new "Interviews" section. The first person interviewed is Tristan Gingold, author of

GHDL, a VHDL compiler written in Ada 95.

<http://www.ada-france.org/article66.html>

[See also "France - GHDL, a VHDL Simulator" in AUJ 24-1 (Mar 2003), p.28. -- dc]

## Open Source Projects in Ada

*From: Yuri <yuri@tsoft.com>  
Date: Sat, 15 Feb 2003 17:12:55 GMT  
Subject: No Ada in OpenSource?  
Newsgroups: comp.lang.ada*

Are there any OpenSource projects in Ada? I looked through ports repository in FreeBSD and didn't find any except Ada related like adasockets or bindings to GTK. If no why don't you guys who have tons of experience in Ada do some OpenSource stuff to promote Ada ?

I am new to Ada but see how advantageous it is to development. And kinda missing why isn't it popular among people. Maybe it needs positive examples like I mentioned ? It probably can cut efforts and tremendously increase stability on "C++" projects like GnuCash or KDE or whatever else. So maybe instead of talking about advantages it's better to build some positive example ?

[And in response to the remark "A quick search at <http://www.sourceforge.net> shows that there are 50+ Ada-projects." -- dc]

True -- SourceForge currently lists 60 projects where Ada is used. But good half of them are purely Ada-related like Ada libraries, bindings, DB access tools. Also few tiny students projects. No standalone powerful packages (like Gimp, GnuCash, XNap, ...). Nothing for the "public" (OpenSource public) to use and mind: this is written in Ada.

*From: Preben Randhol <randhol@pvv.org>  
Date: Sat, 15 Feb 2003 20:02:43 +0000  
Organization: Norwegian university of science and technology  
Subject: Re: No Ada in OpenSource?  
Newsgroups: comp.lang.ada*

[http://freshmeat.net/browse/163/?topic\\_id=163](http://freshmeat.net/browse/163/?topic_id=163)

[This URL currently lists 32 Ada projects in the "freshmeat" database of Unix and cross-platform software, most are under an open source license. -- dc]

*From: Preben Randhol <randhol@pvv.org>  
Date: Sat, 15 Feb 2003 20:05:06 +0000  
Organization: Norwegian university of science and technology  
Subject: Re: No Ada in OpenSource?  
Newsgroups: comp.lang.ada*

[http://sourceforge.net/softwaremap/trove\\_list.php?form\\_cat=163](http://sourceforge.net/softwaremap/trove_list.php?form_cat=163)

[This URL currently lists 66 Ada projects on the "SourceForge" Open Source

software development website. To reach this page, select "software map" on the home page, then "browse by programming language", then Ada. -- dc]

*From: Marc A. Criley <mcq95@earthlink.net>  
Date: Sun, 16 Feb 2003 13:20:02 GMT  
Organization: Quadrus Corporation  
Subject: Re: No Ada in OpenSource?  
Newsgroups: comp.lang.ada*

For whatever reason, the majority of Ada open source projects reside in places other than SourceForge, such as [www.adapower.com](http://www.adapower.com).

*From: dennison@telepath.com (Ted Dennison)  
Date: 19 Feb 2003 06:42:36 -0800  
Subject: Re: No Ada in OpenSource?  
Newsgroups: comp.lang.ada*

I have 3 on my website at <http://www.telepath.com/~dennison/TedTED.html>. The most prominent one is the SETI@Home Service, which is used by SETI@Home enthusiasts all over the world. [...]

[See also "SETI@Home Service 2.1" in AUJ 22-3 (Sep 2001), p.148. -- dc]

> [...] why don't you guys who have tons of experience in Ada do some OpenSource stuff to promote Ada ?

I think you underplay the commitment involved in running a project. But in general, I agree this would be a good thing. [...]

*From: Ken O. Burtch <kburtch@sympatico.ca>  
Date: Thu, 27 Feb 2003 11:06:04 -0500  
Subject: Re: No Ada in OpenSource?  
Newsgroups: comp.lang.ada*

PegaSoft has several open source Ada projects. Visit our site at <http://www.pegasoft.ca>. [See also the "PegaSoft" news items in AUJ 23-4 (Dec 2002), pp.202-203. -- dc]

## AdaIC Opens Ada Sites Search Engine

*From: Randy Brukardt <randy@rrsoftware.com>  
Date: Wed, 12 Mar 2003 23:21:12 -0600  
Subject: AdaIC opens the Ada Sites search engine  
Newsgroups: comp.lang.ada*

The AdaIC is proud to announce the availability of the Ada Sites search engine.

The AdaIC search engine provides a way to search many Ada-related web sites in a single search. Unlike general search engines, the Ada Sites search engine only searches Ada sites listed by the AdaIC. Since only Ada-related sites are included, you won't get piles of unrelated pages, and you won't have to limit your search so much that you can't find the information you need.



The engine allows the inclusion or exclusion of site categories, along with phrase searches, excluded words, and more. When you click the "Search" button, the engine will search more than 25,000 pages of Ada information, presenting the most relevant pages along with short abstracts. You can directly access the search engine at <http://www.adaic.com/site/wide-search.html>.

The AdaIC search engine is created with a set of Ada applications, all written in Ada 95. For more on the search engine and how to use it, see <http://www.adaic.com/site/search-info.html>.

If you know of Ada-related web sites that are not listed in <http://www.adaic.com/links/index.html>, please write us at [webmaster@adaic.com](mailto:webmaster@adaic.com) with the URL and a short description. We'll add it to the search engine on the next update cycle. Comments on the search engine are also welcome.

[And from another message: -- dc]

Somewhat less interesting, but still useful, is the individual search engines (several of which people have been stumbling across and using for the last month). All of these are included in the board search, but it can be useful to look only in a particular site or document.

The entire AdaIC site (<http://www.adaic.com/site/search.html>); The AdaIC archives (<http://archive.adaic.com/search.html>); Ada 95 Quality and Style (<http://www.adaic.com/docs/95style/html/search.html>); The Ada 95 Reference Manual (<http://www.adaic.com/standards/95lrm/html/RM-SRCH.html>); and the Annotated Ada Reference Manual (<http://www.adaic.com/standards/95aarm/html/AA-SRCH.html>).

Also, if you are the webmaster of one of the sites indexed, we have a report of all of the links reached. Most of the sites we indexed had a number of broken links. Just send me a note to request it.

Randy Brukardt, Technical Webmaster, [Adaic.org/Adaic.com](mailto:Adaic.org/Adaic.com)

*From: Randy Brukardt  
<Randy@RRSoftware.Com>  
Date: Wed, 16 Apr 2003 23:06:55 -0500  
Subject: [Ada-Comment] AI search engine now available.  
To: Ada-Comment List <ada-comment@ada-auth.org>*

You can now search the entire set of AIs (including the attached e-mail, of course) at <http://www.ada-auth.org/search-ais.html>. This will search only the latest version of the AIs.

You also can now search the entire ACAA web site (which includes the AIs, the ARG minutes, the ACATS, and the

various ACAA documents) from the ACAA search page: <http://www.ada-auth.org/search.html>.

And all of this material has been added to the Ada-wide search engine at <http://www.adaic.com/site/wide-search.html>, along with some additional sites that were nominated.

Randy Brukardt, ARG Editor & Technical Webmaster, [Adaic.org/com](mailto:Adaic.org/com)

## AdaIC News Items

*From: Technical Webmaster  
<Webmaster@adaic.com>  
Date: Tue, 25 Feb 2003 18:57:52 -0600  
Subject: [AdaIC] New articles posted  
To: <announce@adaic.com>*

We've posted a pair of new articles by Jim Rogers.

First, *Introducing Ada* (<http://www.adaic.org/whyada/intro.html>) gives a tour of Ada for those who are not familiar with her. Jim especially investigates why Ada helps reliability of programs. [See also "Introducing Ada on CodeMages Software Forum" in AUJ 23-4 (Dec 2002), p.208. -- dc]

Second, *Shared Resource Design Patterns* (<http://www.adaic.org/learn/tech/shares.html>) describes various techniques for controlling access to shared resources in multi-tasking programs. It'll be a valuable overview for any Ada practitioner.

*From: Technical Webmaster  
<Webmaster@adaic.com>  
Date: Wed, 23 Apr 2003 19:35:49 -0500  
Subject: [AdaIC] Two news items.  
To: <announce@adaic.com>*

Ann is at the Embedded Systems conference, so she asked me to post these stories.

First, Green Hills Software indicated that their recent growth in profits was due in large part to selling Ada-based development tools. See the full story at <http://www.adaic.org/news/ghsprofit.html>

Second, Vector Software gave Ann an interview about their use of Ada in their VectorCast testing product. They describe how the use of Ada has made their test generation products more error-resistant and "almost impossible to crash". See the full story at <http://www.adaic.org/news/test-sw.html>.

[See also the respective items in the "Ada Products" section of this AUJ. -- dc].

*URL: http://www.adaic.org/news/index.html  
Subject: ARA's Ada Info: Ada in the News*

[Some news items from the AdaIC web site; more information on selected items further in this AUJ issue. -- dc]

Aonix's Object/Raven Helps Complete Aircraft Control Display Unit CMC Electronics Inc, has completed the CMA-2082D Control Display Unit using ObjectAda/Raven for the Canadian

Department of National Defense's CP140 Maritime Patrol Aircraft.

Barnes SPARK Ada Book Revised. High Integrity Software: The SPARK Approach to Safety and Security by John Barnes has been revised. Sample chapters are available at <http://www.sparkada.com/book.html>.

Ada Europe 2003 June 16-20. The Eighth International Conference on Reliable Software Technologies will be held in Toulouse, France.

SIGAda 2003 Slotted for Dec. 7-11. SIGAda is holding its annual conference in San Diego, Calif., this December.

Ada Resource Association adds ACT Europe to Global Membership. The ARA is pleased to announce the addition of its newest member, ACT Europe.

Vector Software Easily Writes Fail-Safe Tests in Ada. Vector Software describes how the use of Ada has made their test generation products more error-resistant and "almost impossible to crash".

Ada Tools Lead Green Hills Through Two Profitable Quarters. At the Embedded Systems Conference, Green Hills Software noted that their recent growth was due in large part to selling Ada-based development tools.

Aonix Grows into Two Companies. Aonix announced that it has split into two companies. The former Critical Systems Division (which includes the ObjectAda products) was purchased by a group of investors led by top management of the CDS division. The new company will keep the name Aonix; the remaining parts of the old Aonix will be known as Select Business Systems. [See also "Aonix - Commitment to Ada" in AUJ 24-1 (Mar 2003), p.18. -- dc]

IBM completes acquisition of Rational Software. IBM announced that it has completed its acquisition of Rational Software. Rational will become a division of IBM.

FAA Certifies INTEGRITY RTOS for DO-178B, Level A Use In Sikorsky S-92 Helicopter. Green Hills Software, Inc. has announced FAA acceptance of a DO-178B, Level A, certification package for its INTEGRITY-178B real-time operating system (RTOS) for a new avionics system aboard the Sikorsky S-92 helicopter.

Airbus Uses Green Hills RTOS. Northrop Grumman Selects Green Hills Software's INTEGRITY-178B RTOS for navigation system in Airbus and other airframe manufacturers.

Green Hills' Integrity RTOS in F-16 Cockpit. Lockheed Martin Aeronautics Company has selected Green Hills Software's, INTEGRITY real-time operating system for use in the F-16 fighter jet's on-board Color Display Processor (CDP).

---

## Ada-related Tools

### WORM\_Str - Write Once Read Many Strings

*From: tmoran@acm.org*  
*Date: Fri, 14 Feb 2003 20:38:47 GMT*  
*Subject: Re: Array Of Constant Strings?*  
*Newsgroups: comp.lang.ada*

Plug: One of the packages I recently posted ([www.adapower.com/os/notify.html](http://www.adapower.com/os/notify.html)) is WORM\_Str - Write Once Read Many Strings. It's exactly intended for situations where you never delete from a table of strings. [...]

WORM\_Str also includes an internal subpackage that adds binary search and sorted dump capabilities. WORM\_Str is not OS, compiler, or library dependent, but just plain Ada.

### Charles - Container Library

*From: mheaney@on2.com (Matthew Heaney)*  
*Date: 19 Feb 2003 16:41:30 -0800*  
*Subject: Re: No Ada in OpenSource?*  
*Newsgroups: comp.lang.ada*

One open source project in Ada95 is the Charles container library, which is modelled closely on the C++ STL. You can get the latest version from my home page.  
<http://home.earthlink.net/~matthewjheaney/charles/> [See also same topic in AUJ 23-4 (Dec 2002), p.192. -- dc]

I posted a new version on 18 Feb 2003, which contains changes to the vector containers, and includes new string containers that can be used as a replacement for Ada.Strings\*. There are new also new html docs for the vectors.

[And from a later message: -- dc]

[...] the STL has already been ported to Ada. The library is called "Charles", in honor of Charles Babbage, with whom Lady Ada was affiliated.

### Booch Components

*From: Simon Wright*  
*<simon@pushface.org>*  
*Date: Tue, 11 Mar 2003 18:05:52 GMT*  
*Subject: Booch Components 20030309*  
*To: team-ada@acm.org*

This release has been uploaded to <http://www.pushface.org/components/bc> and is mirrored at <http://www.adapower.net/booch/>.

Major features since 20030111:

Interface changes: New `unmanaged` forms are provided for all the monolithic Containers. This means you don't need to supply a storage pool at all (and means you can use GNAT 3.12 if you have to!).

Implementation changes: In `BC.Support.Memory_Streams`, `Write_Contents` did the actual writing one storage element at a time. `Read_Contents` requires a supplier Stream that has datagram properties (unlike, it turns out, GNAT.Sockets streams, even if based on a datagram protocol).

[See also same topic in AUJ 24-1 (Mar 2003), p.10. -- dc]

### External Libraries and the Ada Standard

*From: Stephen Leake*  
*<Stephen.A.Leake@nasa.gov>*  
*Date: 06 Mar 2003 14:52:56 -0500*  
*Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)*  
*Subject: Re: Ada2005 clear screen etc.*  
*Newsgroups: comp.lang.ada*

> If we had a committee that built and maintained a library of utility code as a "Reference Implementation" it could be packaged with the compilers but provided in an unsupported mode.

SIGAda is trying to start something like that (see <http://www.sigada.org/wg/apiwg/>). Maybe it will succeed.

[See also "Ada Components and APIs Working Group" in AUJ 24-1 (Mar 2003), p.10. -- dc]

*From: Robert I. Eachus*  
*<rieachus@attbi.com>*  
*Date: Sun, 16 Mar 2003 04:37:47 GMT*  
*Subject: Re: Ada2005 clear screen etc.*  
*Newsgroups: comp.lang.ada*

[On having external libraries such as containers and GUIs in the Ada language standard or not. -- dc]

Let me try to answer this from the ARG standpoint, plus some history. Way, way, back in the Mil-Std 1815 (Ada 80) days, I/O was an integral part of the language. By ANSI/Mil-Std 1815A (Ada 83) I/O required no special features within the language, and there was a set of standard packages, `Direct_IO`, `Sequential_IO`, `Text_IO`, and `Low_Level_IO`, that were required by the standard.

In ANSI/ISO 8652 (Ada 95) these packages were moved into Annex A, predefined language environment, along with other old language features and a few new ones. In addition several specialized needs annexes were added for things that were only of interest to small segments of the Ada population. The idea was that compilers need not implement all of the specialized needs annexes. In practice, the major front ends, including GNAT have tended to implement all of the annexes, it really isn't all that hard. (All you need is someone who can read and understand each annex. ;-)

But the annexes in my mind have been a success, because the design of the

annexes forced us to really work hard on limiting any low-level extensions required to implement an annex. For example, the Information Systems Annex requires a lot of compiler independent software which need only be written once. (Ben Brosgal, Dave Emery, and I wrote a prototype of those operations under contract to the AJPO, and the source code was made public domain.) It also requires supporting a decimal type with at least eighteen decimal digits, and that's about it.

Also there are some dependencies between special needs annexes, but with a structure that makes sense. (For example, it makes no sense to implement the Real-Time Systems annex without the Systems Programming Annex.

But when Ada 9X was in development we spent a lot of effort deciding what should and should not be added to the standard. The NRG had done a lot of work over a decade developing math packages. In 1982, the consensus was that adding a bad numerics library to the standard would be a mistake, and we couldn't delay the Ada standard until they were ready. By the early nineties they were ready, consistent with the IEEE work, and there was little argument that they should not be added. Similarly, for political reasons, support for other character sets was a requirement.

But even though we put a significant amount of effort into developing an SQL interface, the consensus was that it wasn't something that belonged in the standard. (And in my opinion a good thing too. There were three proposals each with its strengths and weaknesses, and I have never really used any of them for more than a two or three page program. However there are now some very nice database bindings that don't require a blinding headache--and that were not available in time for Ada 95. If someone were to take the GNADE work: <http://gnade.sourceforge.net/doc/gnade.html> and propose all or part of it as part of the new standard, go ahead. Personally, I find the thin ODBC binding easy enough to use, but I think that a thinker binding would be more appropriate for adding to the standard.

So what will be added to Ada 0Y? The only real consensus at this point is that there are some container libraries that should be considered. Check the Ada Issues database <http://www.ada-auth.org/~acats/ais.html> for details.

### Ada Standard Containers Library Working Group

*From: maa@liacc.up.pt*  
*Date: Mon, 14 Apr 2003 17:17:42 +0100*  
*Subject: [Ada-Comment] Ada Issues work procedure*  
*To: ada-comment@ada-auth.org*

[...] For [Ada Issue] 302 [Data Structures] I've got many issues as a member of the ASCLWG [1] and as an independent data structures library designer. [...]

[1] ASCLWG = Ada Standard Containers Library Working Group, vd. The Ada-Europe 2002 workshop, and groups.yahoo.com/group/asclwg/.

## Ada-aware gdb 5.3

*From: Oliver Kellogg <oliver.kellogg@t-online.de>*

*Date: Thu, 27 Feb 2003 07:59:02 +0100*  
*Subject: Re: debugging ada with gdb*  
*Newsgroups: comp.lang.ada*

A preliminary version of an Ada aware gdb-5.3 has just been released by ACT, see <http://sources.redhat.com/ml/gdb/2003-02/msg00558.html>

## GNAT Compiler for 68HC11 Targets

*From: Stephane Carrez <stcarrez@nerim.fr>*

*Date: Wed, 15 Jan 2003 21:17:05 +0100*  
*Subject: Re: Are there Ada-compilers for HC11/HC12?*  
*Newsgroups: comp.lang.ada*

gcc 3.2.1 includes the GNAT Ada compiler. I was able to build it for 68HC11/68HC12 and use it on a full example (which fits in 256 bytes!).

The Ada runtime is not compiled and some work is necessary. OTOH I'm not convinced that the Ada runtime as it is is suitable for HC11/HC12. In the example I wrote I took care of not using any runtime support (pragma No\_Run\_Time, pragma Suppress (All\_Checks)). [...]

I'll try to package (tar.gz) my example and give more info when I've time. [...]

*From: Laurent Guerby <guerby@acm.org>*  
*Date: 02 Mar 2003 00:29:36 +0100*  
*Subject: Re: xor*

Stephane Carrez did the port of GCC to 68HC11, he has posted a message entitled "GNAT dream with 3.2.2 pre-release for i686 and m6811-Elf" which you can find at <http://gcc.gnu.org/ml/gcc/2003-01/msg01600.html>.

On his personal web page there is a lot of information: <http://stcarrez.nerim.net/>. I suggest you contact him directly.

*From: Rune Winther <rune.winther@hiof.no>*

*Date: Fri, 28 Mar 2003 09:20:51 +0100*  
*Subject: Ada compiler for 68hc11?*  
*Newsgroups: comp.lang.ada*

We're using the MIT Handyboard (68hc11) in various student projects and I would like to be able to develop Ada programs for it. Any hints to where I might find an Ada compiler for the 68hc11 would be greatly appreciated.

Rune Winther, Ph.D., Ostfold College, Norway

*From: Stephane Carrez <stcarrez@nerim.fr>*

*Date: Fri, 28 Mar 2003 23:22:29 +0100*  
*Subject: Re: Ada compiler for 68hc11?*  
*Newsgroups: comp.lang.ada*

The gcc 3.2.1 compiler can be used but you'll need to use "pragma No\_Run\_Time;" because the GNAT runtime is not ported yet to 68HC11. (I'm also pretty sure it will not fit as is on an embedded system like the HC11). You can't raise an exception unless you port the GNAT exception part.

I wrote an Ada example for 68HC11 some time ago. It uses the 68HC11 bootstrap mode which means it fits in the 256 bytes constraint (actually even less because the stack must be within that 256 bytes region!).

You can have a look at the example: [http://gel.sourceforge.net/ada\\_example.php](http://gel.sourceforge.net/ada_example.php). It shows how you can access the HC11 IO ports in Ada as well as how you can cope without the GNAT runtime. [...]

## GNAT Compiler for AVR Targets

*From: rolf.ebert@gmx.net (Rolf Ebert)*

*Date: 4 Apr 2003 05:01:50 -0800*  
*Subject: Re: Ada for microprocessors ?*  
*Newsgroups: comp.lang.ada*

I started a project some months ago to port gcc/gnat to AVR targets. See <http://tech.ebert-langer.de/avr-gnat.html>. Recently the project was named AVR-Ada and moved to SourceForge at <http://sourceforge.net/projects/avr-ada/>.

The corresponding mailing list has quite low traffic as we are only three right now. Support is always welcome....

## Numerical Integration Libraries

*From: Stephen Leake*

*<Stephe.Leake@nasa.gov>*  
*Date: 22 Apr 2003 16:04:50 -0400*  
*Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)*  
*Subject: Re: Numerical Integration Libraries*  
*Newsgroups: comp.lang.ada*

I have a very simple Runge Kutta implementation licensed under the GMGPL (that's in the "no money" category).

*From: gdemont@hotmail.com (Gautier)*  
*Date: 23 Apr 2003 03:47:09 -0700*  
*Subject: Re: Numerical Integration Libraries*  
*Newsgroups: comp.lang.ada*

If it is on a mesh you can make a Gauss integration using the Finite Element Kernel available in mathpaqs.zip, page below. You may also be interested by the

Numerical Recipes in Pascal and the P2Ada translator - see also there.

<http://www.mysunrise.ch/users/gdm/gsoft.htm>

## TASH - Ada Binding to Tcl/Tk

*From: Terry Westley*

*<westley@yahoo.com>*  
*Date: Tue, 18 Feb 2003 05:01:11 GMT*  
*Subject: announce: releasing Tash 8.4.1a*  
*Newsgroups: comp.lang.ada*

The latest version of Tash (Ada binding to Tcl/Tk) is based on Tcl/Tk 8.4.1, compiled with GNAT 3.15p. You can download a .zip file at [www.adatcl.com](http://www.adatcl.com) which contains all source code and compiled windows binaries for required Tcl/Tk library files. [...]

## VAD 6.1 - Visual Ada Developer

*From: Leonid Dulman*

*<dulman@attglobal.net>*  
*Date: Tue, 15 Apr 2003 22:17:27 +0200*  
*Subject: Announce : Visual Ada Developer VAD 6.1 Leonid Dulman*  
*Newsgroups: comp.lang.ada*

VAD (Visual Ada Developer) is a Tcl/Tk oriented Ada-95(TCL) GUI builder portable to different platforms, such as Windows NT/9x, Unix (Linux), Mac and OS/2. You may use it as IDE for any Ada-95 (C, C++, TCL) project. You may use it to build TCL scripts only. VAD generated Ada sources you may compile and build executables with GNAT on Windows and Unix(Linux) or Aonix ObjectAda 7.2 on Windows.

VAD 6.1 has five realizations: for tcl/tk 8.0.x, 8.2.3, 8.3.5 and 8.4.2 (last version). You need to install and to check tcl/tk before using VAD. From version tcl/tk 8.4.0.1 ActiveState distribution includes many of VAD used packages (Itcl, Img, Tktable, BWidgets, Tkhtml and so on). You may choose the needed version in link time. (I recomend to work with 8.4).

[See also "VAD - Visual Ada Developer" in AUJ 23-3 (Sep 2002), p.134. -- dc]

VAD 6.1 is available in <http://www.websamba.com/ldulman/vad.htm>. You may download sources [...] and binaries [...] (Windows 9x/NT) [...] (Linux).

## ANSI TTY Control

*From: Jeffrey Carter <jrcarter@acm.org>*  
*Date: Sat, 15 Mar 2003 01:41:40 GMT*  
*Subject: Re: Need a gotoXY consol IO procedure for Rational Apex on Sun*  
*Newsgroups: comp.lang.ada*

If you're doing this for Sun then you can probably use ANSI escape sequences. See, for example,

PragmARC.ANSI\_TTY\_Control from the PragmAda Reusable Components.

<http://home.earthlink.net/~jrcarter010/pragmarc.htm>

## Nurses Ada Binding

From: Jeffrey Creem

<[jeff@thecreems.com](mailto:jeff@thecreems.com)>

Date: Fri, 18 Apr 2003 10:55:43 GMT

Subject: Re: GNAT Ada - Clear Screen

Newsgroups: [comp.lang.ada](mailto:comp.lang.ada)

Note that the nurses binding is built in to the later nurses distributions so you dont have to go looking for it:

<ftp://ftp.gnu.org/gnu/nurses>

## Hints for Networking

From: David C. Hoos

<[david.c.hoos.sr@ada95.com](mailto:david.c.hoos.sr@ada95.com)>

Date: Fri, 31 Jan 2003 15:21:34 -0600

Organization: ENST, France

Subject: Re: Hints for networking

Newsgroups: [comp.lang.ada](mailto:comp.lang.ada)

> I'm about to program a networking package for the first time in Ada and am somewhat undecided the way to go. I'd like to hear some pointers for examples or hints. [...]

I have written many many network protocol representations using Ada95 streams. The elegance of Ada Streams, and the simplicity of their use once you have made the effort to properly define your types and their stream attributes really makes the effort worthwhile. Here are some hints and rules needed to do this successfully. These rules are based on the presumption that there is a specific network representation required by your protocol.

1. This rule number 1 is very important. NEVER base any value written to a stream on the standard Ada type Integer. You should always define project-specific types that do not depend on implementations that are subject to variation between compilers or platform types. For example, you can base integer types on those available in the package Interfaces -- Integer\_16, Unsigned\_16, etc. That's what the Interface package is for -- i.e. to permit writing portable code for interfaces.

2. You need to write stream attributes for multi-octet numbers that respect the endiness of your platform. Even when you've been told that "this only has to run on Sparc," sooner or later your code will also need to run on Intel, so I've always taken the trouble to deal with endiness.

3. All objects written to a stream must have a 'Size that is an integer multiple of 8. So, if you have an octet with two four-bit fields, I make them into a record type, and write the stream attributes for that record. I've done things like a record that has a 31-bit field, and a 1-bit field.

Happy streaming!

## HTTP & SMTP

From: [tmoran@acm.org](mailto:tmoran@acm.org)

Date: Thu, 13 Feb 2003 06:57:08 GMT

Subject: ANN: [http & smtp etc packages to adapower](http://www.adapower.com)

Newsgroups: [comp.lang.ada](mailto:comp.lang.ada)

David Botton will soon be posting on [www.adapower.com](http://www.adapower.com) some packages I sent in the hopes they would help someone build the killer Ada app :)

[Direct URL is [www.adapower.com/os/notify.html](http://www.adapower.com/os/notify.html). -- dc]

Http: offers Open/Get/Head/Post routines to access web pages. It includes line (ie, crlf) oriented read with Content-length or transfer-encoding: chunked sizes. It will save/send cookies (default) or ignore them. multitasking safe, of course.

Smtp: handles smtp (e-mail send) protocol.

Sorts: generic List/Merge sort (Knuth Vol 3, alg. 5.2.4L).

WORM\_Str: Write Once, Read Many String store, also includes binary search generic subpackage.

Notify: A 31 line demo of the above, that reads a web page and emails the first line containing the word "title", if there is such a line. [...]

The posted packages are unrestricted in use and themselves use the GMGPL version of Windows Claw, available via [www.adapower.com](http://www.adapower.com)

[And from a later message: -- dc]

They are Ada source files, designed to run in an environment that supplies Claw.Sockets and the Windows winsock API. Feel free to modify them to use a different socket package on a different OS. I note that a different SMTP package, with different requirements and a somewhat different interface, is available at <http://perso.wanadoo.fr/pascal.obry/contrib.html>.

## AWS 1.3 - Ada Web Server Component

From: Pascal Obry <[p.obry@wanadoo.fr](mailto:p.obry@wanadoo.fr)>

Date: 05 Mar 2003 11:54:56 +0100

Subject: ANNOUNCE - AWS 1.3 released

Newsgroups:

[comp.lang.ada](mailto:comp.lang.ada), [comp.lang.ada](mailto:comp.lang.ada)

AWS - Ada Web Server, 1.3 release / SOAP 1.1

Authors: Dmitriy Anisimkov, Pascal Obry

We are happy to announce the availability of the AWS 1.3 release. The API could change slightly at this stage but should be fairly stable now.

AWS stand for Ada Web Server. It is not a real Web Server like Apache. It is a

small yet powerful HTTP component to embed in any applications. It means that you can communicate with your application using a standard Web browser and this without the need for a Web Server. AWS is fully developed in Ada with GNAT.

AWS support SOAP, Server Push, HTTPS/SSL, client HTTP, hotplug modules... We have worked very hard to make this release as stable as possible. Note that Hotplug modules are very nice but have a potentially security hole as it is implemented today. A new secure implementation will be proposed in a future version.

The SOAP implementation has been validated on

<http://validator.software.org/>.

Here are the main changes since AWS 1.2: [see distribution -- dc]

Validation: AWS 1.3 has been compiled and has passed all tests on

- Windows XP, GNAT 3.15a1, 3.16a, 3.17w and 5.01w

- Windows NT 4.0, GNAT 3.15a1

- GNU/Linux x66, GNAT 3.16a and 3.17w

- SPARC Solaris 8, GNAT 3.17w

Other platforms / compiler version combinations have not been tested, it does not mean that it's not working. Previous version of AWS have been build on FreeBSD 4.1 and MacOSX.

Pointers: AWS Home Page (sources and documentation): <http://libre.act-europe.fr/aws>

[See also "AWS 1.1 - Ada Web Server Component" in AUJ 22-4 (Dec 2001), pp.201-202, and further references given there, for pointers to bindings used with AWS, projects using AWS, and much more. -- dc]

## Jabber Instant Messaging Protocol

From: [rd <nospam@spam.no>](mailto:rd@spam.no)

Date: Wed, 16 Apr 2003 11:10:56 -0500

Subject: Re: If anybody wants to make

something in Ada but do not know what

Newsgroups: [comp.lang.ada](mailto:comp.lang.ada)

[In a thread about new e-mail distribution techniques and related protocols to fight spam: -- dc]

Jabber [[www.jabber.org](http://www.jabber.org)] might be just the thing. The best part about Jabber is the real time messaging capability, and if I was designing the second protocol, I would include this functionality.

I don't believe there are any Jabber clients/servers written in Ada, and I'm not sure how well Jabber uses crypto, [...]

From: Pascal Obry <[p.obry@wanadoo.fr](mailto:p.obry@wanadoo.fr)>

Date: 16 Apr 2003 21:16:48 +0200

*Subject: Re: If anybody wants to make something in Ada but do not know what*  
*Newsgroups: comp.lang.ada*

There is one [Ada implementation],  
 Jaffar: <http://picolibre.enst-bretagne.fr/projects/jaffar/>

Pascal Obry, Team-Ada Member, Magny Les Hameaux, France

[Samuel Tardieu <sam@rfc1149.net> also responded: -- dc]

It was developed by students I supervised. If anyone wants to take the developpement from the current sources, create an account on picolibre and I'll add you as a developer.

## AdaSockets Binding

*From: David C. Hoos*  
 <david.c.hoos.sr@ada95.com>

*Date: Wed, 12 Feb 2003 21:41:10 -0600*  
*Subject: Re: TCPIP package or binding?*  
*To: team-ada@acm.org*

> I've been asked to check into the best way to get to TCPIP from Ada.

I would use Ada.Sockets or GNAT.Sockets, both of which provide the necessary interfaces to the OS's TCP/IP stack, as well as UDP/IP.

Ada Sockets for Windows OS  
[http://perso.wanadoo.fr/pascal.obry/archiv\\_e/adasockets-1.0-win32.tar.gz](http://perso.wanadoo.fr/pascal.obry/archiv_e/adasockets-1.0-win32.tar.gz)

Ada Sockets for UNIX OS  
<http://www.rfc1149.net/download/adasockets/adasockets-1.4.1.tar.gz>

## GnuPG Binding

*From: almroth@attglobal.net (Andreas Almroth)*

*Date: 20 Mar 2003 01:39:25 -0800*  
*Subject: Re: GnuPG binding?*  
*Newsgroups: comp.lang.ada*

I have written a binding to GPGME, which is the preferred API for GnuPG (according to the GnuPG team).

[GPGME = the GNU PG ME tool suite for encrypting and signing of date.  
 >From a later message: -- dc]

The license is GPL with the usual exception for generics... adagpgme-0.4.0 is available at  
<http://www.almroth.com/adagpgme-0.4.0.tar.gz>

## Adabot - Real-Time Battle Robot Game

*From: Alfonso Acosta*  
 <alfonso\_acosta\_mail@yahoo.es>  
*Date: Sat, 22 Feb 2003 23:31:08 +0100*  
*Subject: strange execution error*  
*Newsgroups: comp.lang.ada*

I recently started a project in savannah called adabot (<http://savannah.nongnu.org/projects/adabot/>). Its a real-time battle

(<http://realtimebattle.sourceforge.net/>) bot developed completely in Ada. [...]

PS: If anyone is interested in the project I'll be pleased to give him/her a CVS write account (It's obvious that I'm not an experienced Ada programmer and need some help)

## Ada for MUD Game Programming

*From: John R. Stroh*  
 <strohm@airmail.net>  
*Date: Mon, 7 Apr 2003 04:50:06 -0500*  
*Subject: Re: Programming languages*  
*Newsgroups: alt.mud.programming.comp.lang.ada*

kat-Zygfryd <6667@wp.pl> wrote

> I've been writing a mud for some time now in C++ but decided lately to re-write it in Ada. The reason was that although a C++ MUD may work a little faster, reading and extending Ada code is much simpler. Moreover Ada code is more standardized, so it's not a big problem to switch between compilers/archs, where my C++ code compiles well under some compilers and doesn't compile at all on others. And... Ada has got an IDE with very sexy diagram rendering (GRASP).

I've added a follow up to comp.lang.ada, as this will be at least mildly interesting to some of the people there.

When you finish this, please drop a note to comp.lang.ada, ideally some comments about how it went and lessons learned along the way. MUD programming is almost exclusively a C/C++ domain these days; the mere existence of an Ada-based MUD would go a ways toward convincing some people that Ada really is a general-purpose programming language. Also, the fact that you did it in Ada without being coerced by a US DoD contract clause will carry some weight. (For some reason, no one wants to believe Boeing when they say they used, and required all their subcontractors to use, Ada for the 777 because it was a whole lot better than C/C++.)

[See also "Multi-User Dungeon (MUD) Game in Ada" in AUJ 24-1 (Mar 2003), p.15. -- dc]

## P2Ada - Pascal to Ada Translator

*From: Gautier de Montmollin*  
 <gdemont@hotmail.com>  
*Date: Wed, 05 Feb 2003 10:50:27 +0000*  
*Subject: Ann: P2Ada (v.4-Feb-2003)*  
*Newsgroups: fr.comp.lang.ada*

[Translated from French. See also same topic in AUJ 24-1 (Mar 2003), pp.15-16. - dc]

New features of P2Ada, the Pascal to Ada translator:

A module to store types and identifiers allows P2Ada to correctly translate the With instruction, the New allocator, and Read/Write operations to a file.

Import-export of definitions from one translation to another, covering in a flexible way the various kinds of modularity of the Pascal dialects.

64 predefined types, constants, functions, procedures ("classic" Pascal, Delphi, CodeWarrior) with their Ada 95 translation. If the name is (re)defined in the Pascal source, the name is of course used instead of the translation. [...]

[http://www.mysunrise.ch/users/gdm/gsoft\\_fr.htm](http://www.mysunrise.ch/users/gdm/gsoft_fr.htm)

## F2Ada - Fortran to Ada Translator

*From: Jean-Pierre Rosen*  
 <rosen@adalog.fr>  
*Date: Wed, 12 Mar 2003 10:40:28 +0100*  
*Subject: Re: Fortran - Ada*  
*To: <ada-france@ada-france.org>*

> Does somebody know a powerful translator from Fortran 77 or 90 to Ada?

F2Ada: <http://www.informatik.uni-stuttgart.de/ifi/ps/ada/software/html/language-translators.html>

[That page also has a COBOL to Ada translator. -- dc]

## Stream Parsing Library

*From: porton@ex-code.com (Victor Porton)*  
*Date: Wed, 12 Feb 2003 20:36:37 +0500*  
*Organization: Extreme Code Software*  
 (<http://ex-code.com>)  
*Subject: I created a new parser generator*  
*Newsgroups: comp.lang.ada*

I created a library which allows to write parsers of arbitrary streams (e.g. streams of unicode characters or even streams of container objects) entirely in Ada. (You write Ada [code] yourself, not a program [that] would [be] convert[ed] to Ada from a special parser language, there is no such language).

Library is very extensive and customizable (probably more than any other parser toolkit). It is important for reliability that there are no problems with look ahead (as in many other parser generators) for the price of that the library is may be slower.

This library has a deficiency: generally (in the worst case) for each extractor subtype you create (what is sometimes needed to introduce a new syntax construct) you need three things: define the subtype itself, define an instance of it and define a function which returns access to that instance. This is only for the case of complex recursion of syntax constructs. Well, if you are interested I can as well introduce a meta-language to eliminate this problem.

Are you interested despite of this deficiency? If there are the market (i.e. c.l.a will answer yes), I will publish it both as GPL and commercially. Now it is pre-alpha and Extreme Code uses it internally.

[See also "Extreme Code - UniParser" in the "Ada Products" section of this AUJ. -- dc]

## ANTLR Grammar for Ada

*From: Oliver Kellogg <oliver.kellogg@t-online.de>*

*Date: Sun, 2 Mar 2003 20:28:16 +0100*

*Subject: ANTLR grammar for Ada available*  
*Newsgroups: comp.lang.ada*

This is to announce the availability of a first version of an Ada grammar for the ANTLR language recognizer. The grammar is at <http://www.antlr.org/grammars/ada>.

[See also "Parser/lexer generator ANTLR" in AUJ 20-1 (Apr 1999), p.16. - dc]

Currently the main program is just a syntax checker for Ada. The Abstract Syntax Tree (AST) construction is still under development. Anybody wishing to contribute in this area, feel free to contact me.

Oliver M. Kellogg,  
okellogg@users.sourceforge.net

[And from a later message: -- dc]

The grammar has been updated to support AST (Abstract Syntax Tree) construction.

*From: Oliver Kellogg*

*Date: Tue, 22 Apr 2003 23:08:30 +0200*

*Subject: ANTLR Ada grammar update*  
*Newsgroups: comp.lang.ada*

The ANTLR Ada grammar at <http://www.antlr.org/grammars/ada> has been updated. The ANTLR Ada support now includes an Ada tree super grammar.

The tree design has undergone major change, but I believe that it is actually becoming usable. The leaf nodes produced by the tree generator adhere to the RM Annex P grammar as closely as sensible. The first real application, Ada language support for Kdevelop, is currently underway.

[And from a later message: -- dc]

The Ada grammar now has its cvs head at: <http://webcvs.kde.org/cgi-bin/cvsweb.cgi/kdevelop/parts/adasupport>

## OMG CORBA 3.0 IDL Parser

*From: Alexis Muller*

*<Alexis.Muller@lifl.fr>*

*Date: Fri, 11 Apr 2003 10:00:52 +0200*

*Subject: IDL3 Parser*  
*Newsgroups: comp.lang.ada*

I realised an IDL 3 parser in Ada (available here: <http://www.lifl.fr/~mullera/idlcompiler.tar.gz>) Perhaps that can interest somebody...

For now, it does not check semantic rules, but any help would be welcome; -)

Alexis Muller, Laboratoire d'Informatique Fondamentale de Lille (LIFL), Université de Lille 1, France

## MaRTE OS 1.2 - Minimal Real-Time Operating System for Embedded Applications

*From: Mario Aldea Rivas*

*<aldeam@unican.es>*

*Date: Mon, 17 Feb 2003 10:24:36 +0100*

*Subject: MaRTE OS V1.2 released*

MaRTE OS version 1.2 released !

MaRTE OS (Minimal Real-Time Operating System for Embedded Applications) is a real-time kernel for embedded applications that follows the Minimal Real-Time POSIX.13 subset, providing both the C and Ada language POSIX interfaces.

[See also AUJ 23-1 (March 2002), p.18. -- dc]

It allows software cross-development of Ada and C applications using the GNU compilers Gnat and Gcc. Remote debugging of applications is also possible using the GNU debugger gdb. MaRTE OS is available under the GNU General Public License.

In this version, new functionality has been added:

New POSIX functionality included: pthread\_mutex\_timedlock() function, clock\_nanosleep() function, Monotonic clock. I/O Drivers framework. Application-defined scheduling interface improved: Accept and reject threads, Accept and reject mutexes. Port to gnat3.14p (gnat3.13p is no longer used). Boot across Ethernet using "dhcpd" and Etherboot Other minor improvements and bug fixes.

For more extensive documentation and downloading please visit the following URL: <http://marte.unican.es/> [...]

Mario Aldea Rivas, Grupo Computadores y Tiempo Real, Dpto. De Electronica y Computadores, Facultad de Ciencias, Universidad de Cantabria, Santander, Spain.

## Booting an Ada Main Program

*From: sk <sknipe@kct.com>*

*Date: Wed, 26 Feb 2003 17:26:47 -0600*

*Subject: Re: flat binary output with gnatlink*  
*Newsgroups: comp.lang.ada*

> I'd like to get flat binary code in order to boot on it. Can gnatlink or any tool do it?

Your goal is booting an Ada main?

Look at

<http://www.ktc.com/~sknipe/EOSA.html> and if it looks like what you want, directly download

<http://www.ktc.com/~sknipe/EOSA-BOOT-0.0.11-beta.2.tar.gz> [or .tar.bz2 -- dc]

You will have to provide the bits and pieces to make it usable beyond a "Hello" splash screen.

[And from a later message: -- dc]

I put that together with the hope that it would enable people to do interesting things with bootable Ada mains. I would be interested to know of any exciting projects you achieve with it if you find it useful (and also let me know of any bugs etc :-).

## Ada-VHDL Software-Hardware Codesign

*From: Rodrigo Garcia*

*<rodrigo.garcia@epfl.ch>*

*Date: Tue, 01 Apr 2003 16:16:55 +0200*

*Subject: Re: Does anyone know the benefit of codesign using ada and vhdl!*  
*Newsgroups: comp.lang.ada*

> Does anyone know the relevant work about the codesign between Ada/VHDL?

<http://alpha.fdu.edu/~levine/wong/>

*From: volkert@nivoba.de (Volkert)*

*Date: 2 Apr 2003 07:21:53 -0800*

*Subject: Re: Does anyone know the benefit of codesign using ada and vhdl!*  
*Newsgroups: comp.lang.ada*

May be "OOCOSIM" is interesting for you. It is a HW/SW codesign research project focusing on Co-Simulation. It uses Ada/OO-VHDL and as a Top-Level Design Method HRT-Hood/RT-UML.

Have a look at <http://eis.informatik.uni-oldenburg.de/en/research/projects/oocosi m/index.php>

*From: Tony <scr106@york.ac.uk>*

I am doing some work about co-design issues about Ada and VHDL. I found one paper about my research. It is 'Reasoning About Ada'95 As System Level Description Language'. It is written by Yvonne Fritsch, Wolfgang Ecker, Jörg Böttger.

[Available online at

<http://www3.cti.ac.at/ecsi/ecsi/Doc/Other Doc/SLDL/PDF/fritsch.pdf>. -- dc]

## Ada-UML Profile

*From: Simon Wright*

*<simon.j.wright@amsjv.com>*

*Date: 13 Feb 2003 21:08:44 +0000*

*Subject: Re: UML to Ada Mapping*

Newsgroups: *comp.lang.ada*

At the last Ada UK conference, the team decided to work on a profile. [...] The contact point is <http://www.the-training-centre.co.uk/JRA/JRA%20News%20Archive%203.html>

From: Jean-Pierre Rosen

<[rosen@adalog.fr](mailto:rosen@adalog.fr)>

Date: Fri, 14 Feb 2003 09:50:08 +0100

Subject: Re: UML to Ada Mapping

Newsgroups: *comp.lang.ada*

Note that there is a paper on Ada-UML profile at the upcoming Ada-Europe conference. (PLUG, PLUG :-)

From: John Robinson

<[John@JohnRobinsonAndAssociates.com](mailto:John@JohnRobinsonAndAssociates.com)>

Date: Thu, 20 Feb 2003 21:33:53 -0000

Organization: John Robinson And

Associates Ltd

Subject: Re: UML to Ada Mapping

Newsgroups: *comp.lang.ada*

The Ada UK User Group have set up a working group to look at a UML Ada Profile. This is (currently) unrelated to the work being presented at Ada-Europe, although links are being established between the groups. The initial meeting of the Ada UK working group is being held on Monday 24th February 2003, and has attracted some 17 attendees.

Anyone interested in participating in this group, or keeping abreast of developments, should contact myself in the first instance:

[John@JohnRobinsonAndAssociates.com](mailto:John@JohnRobinsonAndAssociates.com)  
Hopefully the group will also be presenting regularly at the two annual Ada UK User Group conferences. We also encourage anyone with similar interests and activities to consider submitting material to these conferences.

Information on the Ada UK User Group, the conferences and other user group activities can be found at [www.AdaUK.org.uk](http://www.AdaUK.org.uk).

## Generating Code from UML Models

From: Simon Wright

<[simon@pushface.org](mailto:simon@pushface.org)>

Date: Thu, 13 Feb 2003 20:28:49 GMT

Subject: Re: Designing for Ada 95?

To: [team-ada@acm.org](mailto:team-ada@acm.org)

> I'm wondering if there's a preferred design template for Ada? UML is pretty much for OO designs and Ada 95 isn't exactly OO.

Not sure it will help all that much, but my view is summarized at <http://www.pushface.org/coldframe/use-cases.html>.

[The URL is for a document entitled "Use Cases for Code Generation".

>From another message: -- dc]

My project is using my open-source code generator ColdFrame (at the moment you

need Rational Rose at the front end, in principle I could interface to eg UML Studio but there are project deadlines!) at <http://www.pushface.org/coldframe/>

[See also "ColdFrame - Ada Framework Code Generator for UML Tools" in AUJ 23-3 (Sep 2002), pp.138-139. -- dc]

The idea is that the UML model is a set of formal domain (subject matter) analysis models. There's little need to model framework concepts (such as where all the current instances of a particular class are kept) because the generated code will do that. Your Ada code appears in separate subprograms and task bodies.

You are not allowed to modify generated specs or package bodies! (and people don't). If you are a really good Ada person you get to work on the framework (aka software architecture); but most of us are modelling, writing code to complete the models, testing it (and engaged in lots and lots of reviews). If you need to change the structure of the code (because you need an additional operation on a class, say) you change the model, regenerate the framework, and carry on. A major reason for this is that it's very hard to relate random changes in code to what they might have meant in the model -- that's easier if your generated code is 1..1 with the model, of course.

We find that in application domains we have about 20% hand-written code (ie, separate bodies), whereas in i/o domains it's the other way round.

From: [martin.dowie@bopenworld.com](mailto:martin.dowie@bopenworld.com)

(Martin Dowie)

Date: 24 Mar 2003 09:54:44 -0800

Subject: Re: case tools

Newsgroups: *comp.lang.ada*

> UML is not well enough defined to fully generate code; you have to add some architecture/design information.

There is a move within the AdaUK and led by Artisan ([www.artisansw.com](http://www.artisansw.com)) to come up with standard UML <-> Ada mappings, perhaps analogous to the mapping between IDL and Ada.

I have looked at their UML -> Ada95 generator and it has matured nicely over the last year. Shame it wasn't available a year ago, or I wouldn't be now using the UML -> C tool ;-)

## Reverse Engineering Tools

From: Marc A. Criley

<[marccriley@earthlink.net](mailto:marccriley@earthlink.net)>

Date: Fri, 14 Feb 2003 06:28:30 -0600

Subject: Re: Designing for Ada 95?

To: [team-ada@acm.org](mailto:team-ada@acm.org)

While it's not cheap [...] Scientific Toolwork's "Understand for Ada" (<http://www.scitools.com/uada.html>) does an admirable job of reverse engineering code and producing lots of diagrams. While not UML, they have nonetheless

proven quite useful when I've had the pleasure of using this tool.

Marc A. Criley, Quadrus Corporation, [www.quadruscorp.com](http://www.quadruscorp.com)

From: [rolf.ebert@gmx.net](mailto:rolf.ebert@gmx.net) (Rolf Ebert)

Date: 14 Feb 2003 04:50:46 -0800

Subject: Re: Reverse engineering of Ada code

Newsgroups: *comp.lang.ada*

It is certainly not freeware/shareware, but the reverse engineering capability of I-Logix's Rhapsody in Ada ([http://www.ilogix.com/products/rhapsody/rhap\\_inada.cfm](http://www.ilogix.com/products/rhapsody/rhap_inada.cfm)) is great.

## On Design Approaches and UML

From: Peter Amey <[peter.amey@praxis-cs.co.uk](mailto:peter.amey@praxis-cs.co.uk)>

Date: Mon, 17 Feb 2003 08:40:18 -0000

Subject: Re: Designing for Ada 95?

To: [team-ada@acm.org](mailto:team-ada@acm.org)

[About UML: -- dc]

> My point really was just that a common nomenclature is better than everyone having their only personal drawing methodology - it saves a lot of time if we are all speaking the same language. [And:] At least with a common set of diagramming techniques and some semantic background you have somewhere to start from.

This is fine but does not represent the current situation. Currently we have a "common set of diagramming techniques" but we do not have much (if any) "semantic background" and, unfortunately, many people seem to regard UML and design to be "finished" rather than "somewhere to start from".

The problem at present, especially when targeting Ada, is that UML, as described in books and as implemented in current tools, drives you down a path of programming by class extension which may not be appropriate for all problem domains. It compounds this problem by not providing standard support for those areas where Ada is more expressive and capable of better abstraction than the design notation. Ada's packages and private types for example. I have even heard UML tool vendors describing Ada as "deficient" because it does not match UML's model of the world properly.

The end result is that real design skills are being supplanted by knowledge of particular tools and mechanical ways of using them. In the end we reach the farcical levels of Shlaer Mellor where the very existence of a software design process is denied (SM alleges that once you have done the OO analysis, you just code the objects and tip them in a box and stir - no software design is required at all).

At Praxis we have a put a lot of effort into a design approach for SPARK (and hence

for Ada) called INFORMED; this places great emphasis on minimising coupling between units by reducing information flow (which is measured by the SPARK Examiner) to a functional minimum. The method used to minimise information flow is careful choice of where "state" is located; e.g. an instance of an abstract data type as a local variable of a subprogram creates less information flow than use of a library-level abstract state machine. We have found it interesting how hard it is to get these concepts across to UML users and tool vendors who tend to see location of state as an implementation detail where for us it is a prime design driver!

PS. I wonder if there is any connection between the exporting of software engineering jobs to India etc. and this trend to see design as a mechanical, tool-driven process? Real insight is rather harder to export.

*From: Wojtek Narczynski  
<wojtek@power.com.pl>  
Date: Mon, 17 Feb 2003 11:47:32 +0100  
Organization: Power Media Sp. z o.o.  
Subject: Re: Designing for Ada 95?  
To: team-ada@acm.org*

I am from "etc." so I may lack proper "insight", but I'd rather blame the (western) "Loose weight without excersises" => "Generate some software from your UML charts in seconds" wave. Nobody is telling (especially young / new) people that software engineering is a hard discipline, despite the ease to get started.

*From: Simon Wright  
<simon@pushface.org>  
Date: Mon, 17 Feb 2003 20:09:59 GMT  
Subject: Re: Designing for Ada 95?  
To: team-ada@acm.org*

I think UML's area of application (I was going to write "strength") is in modelling problem domains. I completely agree that it doesn't handle modelling Ada implementations -- but I would not want to ask it to. And I wouldn't want to model Java or C++ implementations either.

What I mean by that is, often you will find a tool where if you want something in the code it has to be in the model; which is why you get all this elaborative rubbish, where the analysis model is crufted up with all sorts of inessential implementation detail. What you need is a tool which will apply the software architect's rules to the analysis model and generate the support framework required.

I dare say this ["no software design is required at all"] is what used to be sold (it certainly sounds like what some of the practitioners were selling) and it is clearly tosh. And it is not what you would hear from executable UML practitioners nowadays. What they will tell you is that the modelling exercise (in the problem domain) is quite separate from the

implementation exercise (in the software domain) and that it is the proper interest of the software domain to support (automated) translation from problem to software.

And the translation depends heavily on the constructs in the target software domain. My present project is a memory-based VxWorks system with very little persistence. The design of the software architecture is quite different from what would be appropriate if we had a distributed real-time database (as a neighbour project has). But the contention of the executable UML guys is that the application model should not change just because of this; it's the translation engine that changes.

## The Ultimate SI Units Cracker

*From: Christoph Grein  
<christoph.grein@eurocopter.com>  
Date: Wed, 5 Feb 2003 08:03:15 +0100  
Subject: SI Units Checked and Unchecked  
Newsgroups: comp.lang.ada*

The ultimate SI Units checker has been updated. It now handles reading of units like reading of enumeration values. This only affects file reading operations for illegal items, i.e. those which raise Data\_Error. Hours, minutes and liters have been added as units.

[See also same topic in AUJ 23-4 (Dec 2002), p.201. -- dc]

[...] Unit checking can be switched off in production code by just changing a few lines of code. See <http://home.T-Online.de/home/Christ-Usch.Grein/Ada/SI.html> [...]

## AUnit Test Framework Experiences

*From: Stephen Leake  
<Stephen.A.Leake@nasa.gov>  
Date: 11 Feb 2003 12:04:59 -0500  
Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)  
Subject: Re: AUnit test framework  
Newsgroups: comp.lang.ada*

> does anyone have experience with the AUnit test framework for Ada95?

Yes. See <http://savannah.nongnu.org/projects/grace>  
[See also "ACT - AUnit 1.01 - xUnit Test Framework for Ada" in AUJ 23-1 (Mar 2002), p.19. -- dc]

The tests for Grace use Aunit. For now, you have to get them from CVS (possible via a web browser). One of these days I'll do a tarball release. Ping me if you really want that :). There is pretty good documentation with the package. But a real example always helps.

*From: Simon Wright  
<simon@pushface.org>*

*Date: 11 Feb 2003 20:31:26 +0000  
Subject: Re: AUnit test framework  
Newsgroups: comp.lang.ada*

My project is using AUnit. 15 engineers. I think we've found no problems with AUnit (there are some issues with programming standard conformance, warnings with -gnaty).

## Graphical Tools for Ada Package Dependencies

*From: Jeffrey Creem  
<jeff@thecreems.com>  
Date: Sat, 22 Feb 2003 13:16:34 GMT  
Subject: Re: Ada project dependency tree  
Newsgroups: comp.lang.ada*

> Is there an application which generates a dependency tree from a set of Ada files?

GPS was just recently released to supported customers. [...] [http://www.gnat.com/texts/products/gps\\_set.htm](http://www.gnat.com/texts/products/gps_set.htm). I know of no other free graphical dependency tree tool.

Most vendors provide something like what you want with their tools. (Seem to recall Green Hills Ada doing this, and Rational Ada Analyzer.) Also take a look at Understand for Ada (free (as in beer, not speech)) demo available <http://www.scitools.com/uada.html>.

*From: prichtmyer@yahoo.com (Peter Richtmyer)  
Date: 22 Feb 2003 08:35:57 -0800  
Subject: Re: Ada project dependency tree  
Newsgroups: comp.lang.ada*

> By dependency tree I meant a graphical representation of package dependencies in a project.

Understand for Ada can give you this and alot more. It is not free. But they have a very liberal policy for giving out evaluation licenses. IMHO, most Ada projects would save much money and time (especially large re-use projects) if they bought this tool. It has a very easy to use GUI interface, and runs on lots of machines. [URL higher. -- dc]

## Com\_Port - Serial Communication

*From: Jerry Petrey  
<jdpetrey@raytheon.com>  
Date: Fri, 07 Mar 2003 08:17:37 -0700  
Organization: Raytheon Company  
Subject: Re: rs232 communication  
Newsgroups: comp.lang.ada*

> My OS is windows 2000 I work with gnat 3.15p and adagide

I have found Stephen Leake's com\_port packages very useful in that environment. You can find them at: [http://www.toadmail.com/~ada\\_wizard/](http://www.toadmail.com/~ada_wizard/) - dc]



Jerry Petrey, Senior Principal Systems Engineer, Raytheon Missile Systems

*From: Stephen Leake*

*<Stephen.A.Leake@nasa.gov>*

*Date: 07 Mar 2003 10:45:06 -0500*

*Organization: NASA Goddard Space Flight Center*

*Subject: Re: rs232 communication*

*Newsgroups: comp.lang.ada*

[[http://www.toadmail.com/~ada\\_wizard/](http://www.toadmail.com/~ada_wizard/) - dc]

Get com\_ports.zip (yes, this is so old I wasn't using .tar.gz then :).

## Accessing PC Ports

*From: Stéphane Richard*

*<stephane.richard@verizon.net>*

*Date: Mon, 24 Mar 2003 14:37:43 GMT*

*Subject: Re: Ports*

*Newsgroups: fr.comp.lang.ada*

[Extracts translated from French. -- dc]

<http://adasl.sourceforge.net> contains ASL (Ada Structured Library) which has generic extensions for communication with ports on PCs.

[http://www.usafa.af.mil/dfcs/bios/mcc\\_html/ada\\_stuff.html](http://www.usafa.af.mil/dfcs/bios/mcc_html/ada_stuff.html) also contains source code to access serial (serial.zip) and parallel ports (parrallel.zip). You just have to check out the site. (Note that some examples work under the Windows environment, nevertheless this should be useful to learn...)

And concerning Linux/UNIX you can take a look at

<http://home.attbi.com/~minyard/>. Corey has a POSIX compatible implementation. [...]

## Logging API

*From: chris.m.moore@amsjv.com (Chris M. Moore)*

*Date: Thu, 27 Mar 2003 15:00:52*

*Organization: Alenia Marconi Systems*

*Subject: Re: Ada logging utilities?*

*Newsgroups: comp.lang.ada*

> Does anyone know if there exists any good Ada library for logging (in the same fashion as the Java Logging).

I have a logging API. It's not a clone of log4j which is designed for performance. Mine is designed for flexibility (a directed acyclic graph of logging elements). Unfortunately it's alpha software at the moment, although it shouldn't take long to finish a minimal working version.

<http://sourceforge.net/projects/ada-gerbil/>

## Spreadsheet in Ada

*From: Stephane Richard*

*<stephane.richard@verizon.net>*

*Date: Mon, 28 Apr 2003 18:04:39*

*Subject: Re: Tableur En Ada*

*Newsgroups: fr.comp.lang.ada*

[Extracts translated from French. -- dc]

> Does somebody have Ada source code for a spreadsheet (Excel like or much simpler)?

The following site talks about a Spreadsheet application that might interest you. It's in English, but contains the source code of a spreadsheet in Ada.

<http://www.it.bton.ac.uk/staff/je/adacraft/ch18.htm>

## Ada-related Products

### ACT - GnatPRO-Announce Mailinglist

*From: Cyrille Comar <comar@act-europe.fr>*

*Date: Mon, 24 Feb 2003 16:47:35 +0100*

*Subject: Creation of the gnatpro-announce mailinglist*

*To: gnatpro-contacts:@cs.kuleuven.ac.be;*

This message is to let you know that we are creating a new mailing list which is designed to keep you informed of major new releases, product road maps, and other events relevant to the GNAT Pro technology created and maintained by Ada Core Technologies and ACT Europe.

If you are interested in joining this list you can send an email to [gnatpro-announce-request@lists.act-europe.fr](mailto:gnatpro-announce-request@lists.act-europe.fr) with a message body containing exactly the word "subscribe" (simply using the 'Reply' functionality of your email program should be sufficient to write to that address). You can also subscribe online using the web interface available at <http://lists.act-europe.fr/mailman/listinfo/gnatpro-announce> [...]

### ACT - GNAT Programming System IDE (GPS)

*From: Cyrille Comar <comar@act-europe.fr>*

*Date: Tue, 4 Mar 2003 18:39:44 +0100*

*Subject: Announcing the release of the GNAT Programming System IDE*

*To: gnatpro-announce@act-europe.fr*

Announcing the release of the GNAT Programming System IDE

We are pleased to announce the release of the GNAT Programming System, GPS, the Ada-aware and multi-language IDE that streamlines the interaction between developers and their software. GPS integrates the GNAT Pro tool set within a single development environment that makes visual sense of code.

With its intuitive interface GPS is easy to use, simplifying source navigation and highlighting the fundamental ideas in the program. Designed by programmers for programmers, GPS is a new kind of IDE that offers the experience of designing

software in a uniquely comfortable environment.

GPS is bundled with GNAT Pro and is available on the following host platforms: sparc-solaris, x86-linux, x86-windows.

We invite you to visit our website at <http://www.act-europe.fr> or <http://www.gnat.com> where you will find a variety of informative texts and videos introducing GPS and highlighting its key features. [...]

[See also "ACT announces GPS" in AUJ 23-2 (Jun 2002), pp.74-75. -- dc]

### ACT - GNAT Pro 3.16a

*URL:*

*[http://www.gnat.com/texts/news/news\\_gnat316.htm](http://www.gnat.com/texts/news/news_gnat316.htm)*

*Date: Mon, 10 Mar 2003 19:54:36*

*Subject: ACT press releases*

Ada Core Technologies announces the release of GNAT Pro 3.16a

We are happy to announce the immediate availability of the GNAT 3.16a release. GNAT Pro 3.16a is the candidate release of the GNAT 3.16 technology. It provides a number of enhancements to the 3.15 technology [...] and also corrects reported problems [...].

This release marks the introduction of GPS, the GNAT Programming System integrated development environment (IDE), on the following host platforms: sparc-solaris, x86-linux, x86-windows. [...]

[See also "ACT announces GNAT 3.15a" in AUJ 23-2 (Jun 2002), p.75. -- dc]

### ACT - GNAT Pro 5.00a

*URL:*

*[http://www.gnat.com/texts/news/news\\_gnat500.htm](http://www.gnat.com/texts/news/news_gnat500.htm)*

*Date: Wed, 19 Mar 2003 21:35:48*

*Subject: ACT press releases*

Ada Core Technologies announces the release of GNAT Pro 5.00a

We are happy to announce the release of GNAT Pro 5.00a. 5.00a is the first GNAT Pro release based on the new generation of GCC backend (GCC 3.2.3). In all other respects, 5.00a is equivalent to the recently announced 3.16a release (based on GCC 2.8.1).

The GCC 3 backend implements several new optimizations improving execution performance. It offers a new Zero-Cost Exception mechanism (currently available on alpha-openvms, pa-hpux, sparc-solaris) that should boost the performance of applications using exception handlers extensively.

This release is the first step in the GNAT Pro migration to the GCC 3 technology. Other major ports will transition to GCC 3 in the forthcoming GNAT 5.xx releases. New developments & ports will be

performed on this new branch of our technology. The GNAT Pro 3.xx technology will continue to evolve in parallel during the transition period. [...] Contact sales@gnat.com

## Aonix - Grows Into Two New Companies

**URL:**  
[http://www.aonix.com/pr\\_02.28.03.html](http://www.aonix.com/pr_02.28.03.html)  
San Diego, California, February 28, 2003  
Aonix announces the growth of the former Aonix Corporation into two new companies.

The entire former Critical Development Solutions (CDS) division with all the existing products is now a new company keeping the name of Aonix. This company will continue to provide software development solutions for mission-critical and safety-critical applications and will focus on the Space, Avionics, Defense, Transportation and Industrial markets. The former Aonix Divisions, eBIS and Select product divisions, remain intact and have been renamed Select Business Solutions, Inc. and will focus directly on the commercial IT market sector. The growth into two companies allows each new company to focus on their respective market areas.

The former CDS general manager, Nicolas Hadjidakis and the CFO, Christophe Faurere, led a group of brand new investors who purchased the new Aonix. The faith that Aonix employees have in their company and the fact that the CDS division bucked industry trends by generating significant revenue growth in 2002 attracted the new investors. New additional venture capital funding is expected to specifically grow the company around existing products. Aonix current products are: Software Through Pictures (StP), ObjectAda, Raven, TeleUSE, and high performance Java based products.

"We are very pleased to have the opportunity to build upon the fantastic foundations of the CDS division." says Nicolas Hadjidakis, CEO and President of the new Aonix. " We see a growing demand for well engineered 'high-integrity' software for mission-critical applications in all the markets we are targeting.

The new Aonix is retaining the entire CDS management team, as well as its staff, development and support centers for all CDS product lines. All Aonix offices in the US, France, Germany, Sweden, and the UK remain in their current locations with the same contact information.

Press Contacts: Greg Gicca (US), 1-603-429-3415, [gicca@eonix.com](mailto:gicca@eonix.com); Jacques Brygier (Europe), +33 1 41 48 11 05, [brygier@eonix.fr](mailto:brygier@eonix.fr); Additional Product

Information, 1-858-457-2700, [info@eonix.com](mailto:info@eonix.com). [...]

## Aonix - ObjectAda for Windows 7.2.2 Patch Update

*From: owner-intel-objectada*  
*Date: Thu, 20 Mar 2003 08:53:15 -0800*  
*Subject: Intel-OA: New ObjectAda update (1102V722-U8)*  
*To: intel-objectada@eonix.com*

A new patch update to ObjectAda for Windows 7.2.2 (1102) is now available. The update download file and the Release Notes are available at <http://www.aonix.com/content/support/ada/patches/objectada.html>

Please see the Release Notes for more information.

Aonix Ada Support,  
[adasupport@eonix.com](mailto:adasupport@eonix.com), 1-800-972-6649

*From: Steve Hancher*  
*<shancher@eonix.com>*  
*Date: Fri, 11 Apr 2003 06:39:36 -0700*  
*Subject: RE: Intel-OA: Problem with calling Ada DLL from MS C++ 6.0*  
*To: intel-objectada@eonix.com*

OA 7.2.1 corrected 125 problems, implemented a new code generation technology providing two levels of optimization, and upgraded the Intel-OA product to the current ACATS (Ada Conformity Assessment Test Suite) level.

OA 7.2.2 corrected another 36 problems. The latest OA 7.2.2 cumulative patch corrects an additional 29 problems. Good things take time and are usually accomplished incrementally. Aonix remains committed to providing the Ada community with quality Ada products and appreciates your patience.

## Aonix - Software through Pictures 8.3.1 for Linux Available

**URL:**  
[http://www.aonix.com/pr\\_03.31.03.html](http://www.aonix.com/pr_03.31.03.html)

Software through Pictures (StP) Version 8.3.1 for Linux now Available

San Diego, California, March 31, 2003

Aonix, a leading provider of software design tools, development environments, and safety critical solutions, announces the immediate availability of Software through Pictures (StP), version 8.3.1 for Linux.

The new Linux version of the well-established modeling tool, StP, completes Aonix' Unix offerings, strengthening StP's position as a multi-platform modeling tool. Aonix also continues to move consistently toward UML and Model Driven Architecture (MDA). MDA is the newest initiative of the OMG ([www.omg.org/mda](http://www.omg.org/mda)), which will soon be the standard approach for software

projects. By creating a Platform Independent Model (PIM), the level of abstraction in the models is increased and the dependency on the target platform is reduced. The PIM is then transferred into the target environment.

Aonix - an active OMG member since 1992 - has been promoting the transformation of UML models to the target environment for years. Aonix provides a powerful transformation engine for UML models with StP's Architecture Component Development (StP/ACD), already proven in many projects.

StP offers special UML profiles to generate C, C++, Ada 95, Java, CORBA, COM, and EJB. Further industry-specific solutions are available for auto-manufacture, real-time systems and safety-critical applications. This makes StP suitable for a wide range of software projects and systems engineering. [...]

Press Contacts: [see above -- dc]

## DDC-I - TADS-1750A Support for Windows Environments Added

**URL:**  
[http://www.ddci.com/news\\_vol4num2.shtml](http://www.ddci.com/news_vol4num2.shtml)  
*Subject: Embedded News from DDC-I*

[...] February 2003 DDC-I Online News

DDC-I's Trusted TADS-1750A Development System Now Offers Full Support for Windows NT, 2000, and XP Host Environments

Phoenix, AZ and Lyngby, Denmark February 18, 2003. DDC-I today announces the addition of support for Windows based development platforms for the TADS Ada Development System targeting the 1750A (TADS-1750A) product family. In addition to Windows NT, 2000, and XP hosting capabilities, TADS-1750A will continue to support the original Sun SPARC, and DEC VAX/VMS host development environments.

"Updating TADS-1750A to support the latest versions of Windows is a direct response to the needs of our customers, who continue to migrate safety-critical software development for a variety of projects to these increasingly popular platforms," explains Richard Frost, DDC-I Senior Software Engineer and TADS Windows Rehost Project Manager.

TADS-1750A offers a mature solution, combining a highly optimising compiler with selective linking and modular run-time systems to generate the most compact code available. With classical optimizations and performance benefits specific to the Mil-STD-1750A processor architecture, Ada 83 specific compiler optimizations include data packing,

constraint and overflow check elimination and static aggregate initialization.

"TADS remains a valuable development environment for real-time embedded system developers in aerospace, avionics, defense, and many other safety-critical applications where failure is simply not an option. We are dedicated to providing our clients with quality tools as well as customizing solutions as needed for their individual requirements," Frost concludes.

## DDC-I - Windows Migration Package for TADS-1750A Customers

URL:

[http://www.ddci.com/news\\_vol4num2.shtml](http://www.ddci.com/news_vol4num2.shtml)  
Subject: Embedded News from DDC-I

[...] February 2003 DDC-I Online News  
DDC-I Offers TADS-1750A Customers Simple, Cost-Effective Windows Migration Package

Phoenix, AZ and Lyngby, Denmark -- February 21, 2003 -- DDC-I today announced the availability of a new Windows (NT/2000/XP) migration package for existing TADS-1750A users. Designed to streamline the transition from VAX or UNIX-hosted development systems, this limited-time package is fully customizable, offering current customers an affordable migration path to the most popular PC-based network and enterprise computing platform.

"The TADS for Windows program is affordable and flexible, and allows organizations to dictate exactly what tools and support they need, instead of handing them a rigid list of options," explains Bud Blum, DDC-I Product Champion for the TADS product line.

With DDC-I's guidance, customers define package parameters to create a least-cost migration path which includes all necessary license transfers and keys to replace current TADS licenses. Software support from the current license agreement also carries over, to keep recurring costs level, while the customer has complete freedom to select the number of seats they want to rehost and whether to upgrade their software versions during the migration.

According to Blum, the package combines two days of on-site consulting at no additional charge to assist with rescripting, memory and segment set up, tool adaption, related ethernet work and board support upgrades. A final project report with detailed recommendations is also included.

"Our goal is to help our customers get the best from their safety-critical software development tools, and when it comes to managing a platform migration, minimal disruption for the programmers and the

development environment they depend on is top priority," concludes Blum.

## DDC-I - TADS-i960 Support for Windows Environments Added

URL:

[http://www.ddci.com/news\\_vol4num3.shtml](http://www.ddci.com/news_vol4num3.shtml)  
Date: Tue, 3 Jun 2003 14:01:01 +0200  
Subject: Embedded News from DDC-I

[...] March/April 2003 DDC-I Online News

DDC-I's Mature TADS-i960 Software Development System Offers Full Support for Windows NT, 2000, and XP Host Environments

Phoenix, Arizona & Lyngby, Denmark March 31, 2003. DDC-I today announces the addition of support for Windows based development platforms for the TADS Ada Development System targeting the Intel i960 product family. In addition to Windows NT, 2000, and XP host capabilities, TADS-i960 continues to support the original Sun SPARC, and DEC VAX/VMS host development environments.

"We updated TADS-i960 to enable hosting on the latest versions of

Windows to meet the needs of many of our customers, who are steadily migrating safety-critical real-time software development to these increasingly popular enterprise network platforms," explains Harold "Bud" Blum, DDC-I Senior Software Engineer and Product Champion for the TADS-i960 product line.

TADS-i960 offers a mature solution with Ada-specific optimisations like constraint and overflow check elimination, parameter binding, data packing, and static aggregates initialization. With target-specific and classical optimizations tuned for the i960 architecture, five optimization levels permit the proper optimisation strategy at each point in the development cycle. Multiple process management facilities and on-chip memory management help produce the lowest possible run-time memory usage on an i960.

"For real-time embedded system developers in aerospace, avionics, defense, and any other safety-critical industry where application failure is not an option, TADS is a valuable development environment, and DDC-I remains committed to ensuring our customers have quality tools and customized solutions to meet their specific development requirements," concludes Blum.

## Extreme Code Software - UniParser Library

From: [porton@ex-code.com](mailto:porton@ex-code.com) (Victor Porton)

Date: Mon, 10 Mar 2003 21:12:07 +0500

Organization: Extreme Code Software  
(<http://ex-code.com>)

Subject: UniParser 0.1 released.  
Newsgroups: [comp.lang.ada](mailto:comp.lang.ada)

UniParser 0.1alpha released by Extreme Code Software. Alpha - unknown bugs are possible, API may change.

See <http://ex-code.com/uniparser/>. Also (devel page)  
<http://sf.net/projects/uniparser/>. [...]

It is a very flexible, powerful and extensible Ada library for creating grammar parsers and lexers. Works not only with streams of ASCII characters but with any data streams (e.g. Unicode). See <http://sf.net/projects/uniparser/> for more benefits. API documentation included.

Released under both GPL-2 and a commercial license. There is also a tutorial (also as the library itself preliminary, tell me if you don't understand something in the tutorial please), see the Web site for conditions to get the tutorial. If you buy the commercial version among other benefits you receive gratis updates (of both the library and tutorial) for 4 years and technical support.

Please test it and put info about it on the Ada Web sites. I'm also interested in receiving parser examples (written with UniParser) and whether it is enough fast for you.

Deficiency: the source code of parser is bigger than for most other parser toolkits. Maybe will be created a metalanguage over Ada to get over this problem. I like to hear any comments (in c.l.a or by e-mail).

From: [porton@ex-code.com](mailto:porton@ex-code.com) (Victor Porton)  
Date: Fri, 14 Mar 2003 09:21:40 +0500  
Organization: Extreme Code Software  
(<http://ex-code.com>)  
Subject: UniParser 0.21 - ready, changed  
Newsgroups: [comp.lang.ada](mailto:comp.lang.ada)

UniParser 0.21alpha released. It is the first official release. Some new features added.

Because of unexpectedly high interest to UniParser there are changes. Firstly, more features are planned for development as well as more caution to make it more consistent, stable, and reliable. Added option to buy UniParser on CD. The above also means increased price. Also changed licensing policy. [...] These changes are because of unexpected high interest. From now the policy will be more stable.

You can buy UniParser now from <http://www.ex-code.com/uniparser/>. Buying it you will have the right of gratis updates during 4 years.

[And from a later message: -- dc]

I just as encountered so high interest to UniParser realized that I may be flooded by support requests etc. I was just forced

to change the rules. Again I was forced to make changes. And very probably rules will be some day changed again to more free. By the way I also write really free software 100% independent of business. And one more time: for really free non-commercial software UniParser is REALLY also 100% gratis, just request it (see the site). [...]

*From: porton@ex-code.com (Victor Porton)*  
*Date: Sat, 15 Mar 2003 04:29:40 +0500*  
*Organization: Extreme Code Software*  
*(http://ex-code.com)*  
*Subject: UniParser 0.22 - now LGPL :-))*  
*Newsgroups: comp.lang.ada*

OK, now UniParser (<http://ex-code.com/uniparser/>) is Open Source (LGPL-2.1). Customers (?) was correct about licensing, be glad now. I look like a comic, but the product is serious. (It is indeed commercial, online selling with both download and CD should work now.) Well, the tutorial is not freeware.

Now 0.22. The versions 0.21 and 0.22 differ mainly by licensing. [...]

## Green Hills - Ada Tools Lead Green Hills into Profitability

*URL:*  
<http://www.adaic.org/news/ghsprofit.html>

Ada Tools Lead Green Hills Through Two Profitable Quarters

San Francisco (April 23, 2003) Embedded Systems Conference – Ada programming tools sales to defense and aerospace are contributing heavily to Green Hills Software's increasing revenues and sharply rising profits, according to Christopher Smith, Vice President of Marketing.

"We're really out of the doldrums," Smith said in a recent telephone interview. The company reported a profitable last quarter of 2002 and sees the trend continuing in 2003. "We're in a period of growth and hiring people again. We think the recession we've been suffering through is over."

Smith said the growth in fourth quarter defense and aerospace business was due in large part to business involving selling licenses to Ada-based development tools.

While Green Hills does not look under the hood of its customers' software development projects to see which languages they are using, the company does have an excellent way of telling whether or not they are choosing Ada. Green Hills sells a version of its debugger environment called Ada-Multi which customers order if they are using more than C and C++.

In three major avionics design wins for INTEGRITY that Green Hills has announced recently -- Boeing's C-17 jet planes, the Boeing B1B, and the Rockwell

Collins Avionics System for the Sikorsky S-92 -- Ada is evidently very much part of these flight-critical applications.

"They're all major programs for which the developers certainly had a choice and for a variety of reasons chose Ada," Smith said.

Boeing's B-1B Conventional Mission Upgrade Program (CMUP) consists of adding a family of 1760 smart weapons to the Air Force fighter's arsenal. Green Hills' INTEGRITY will be used to enhance the B-1B's onboard avionics flight system and take advantage of CMUP's smart weapons and multiple-target capabilities.

Boeing's C-17 Globemaster III for the USAF will use INTEGRITY-178B as well as GMART (Green Hills Minimal Ada Run-Time) and GSTART (Green Hills Safe-Tasking Ada Run-Time System) to host the jet's navigation, mission planning, and display functions. Boeing is also using Green Hills Software's AdaMULTI Integrated Development Environment (IDE) and G-Cover code coverage tools to develop DO-178B-compliant Ada application software for the jet transport plane.

Rockwell Collins is using a similar toolset, the INTEGRITY-178B RTOS and GSTART Ada run-time environment, for the Sikorsky S-92 helicopter's new highly-integrated avionics package. The new software will run the Avionics Management and Display System, which incorporates the most advanced avionics technology available today for rotorcraft.

See <http://www.adaic.org> for more on Green Hills and on Ada. Or write to Ann Brandon, Communications Director, Ada Resource Association, [ann@onyons.com](mailto:ann@onyons.com).

## PegaSoft - BUSH AdaScript Business Shell

*From: Ken O. Burtch*  
*<kburtch@sympatico.ca>*  
*Date: Fri, 14 Feb 2003 20:22:04 -0500*  
*Subject: ANN: BUSH 0.9.2 released*  
*Newsgroups: comp.lang.ada*

BUSH 0.9.2 has been released. It is available at the PegaSoft website: <http://www.pegasoft.ca/bush.html>

BUSH, the AdaScript Business Shell, is a Linux shell for designing secure, reliable shell scripts that can be later compiled as a fast executable programs. BUSH comes with 10 built-in packages including numerics, string processing, and measurement conversions. (It plays WAV, AU and audio CD's, too.)

We're looking for people to help port BUSH to UNIX.

[See also "PegaSoft - BUSH Business Shell" in AUJ 23-4 (Dec 2002), pp.202-203. -- dc]

Ken O. Burtch, PegaSoft, Jordan Station, ON, Canada

*From: Ken O. Burtch*  
*<kburtch@sympatico.ca>*  
*Date: Tue, 25 Feb 2003 13:37:06 -0500*  
*Subject: BUSH on SourceForge*  
*Newsgroups: comp.lang.ada*

Someone (not me) has set up a BUSH project on SourceForge. You can download the sources using CVS and submit changes and bug fixes. The project link is <http://sourceforge.net/projects/bush/>.

Many hands make light work.

## PolySpace Technologies - PolySpace Developer Edition

*URL: http://www.dedicated-systems.com/VPR/layout/display/pr.asp?PRID=4292*

PolySpace Technologies releases PolySpace Developer Edition

Release Date: Friday, April 11, 2003

Summary: PolySpace Technologies, worldwide leader in the static verification of embedded software, announces the release of PolySpace Developer Edition, the latest solution from PolySpace, that brings the power of unparalleled Abstract Interpretation techniques to the developer's desktop for finding bugs without testing.

With a fully automatic setup, PolySpace Developer Edition is the first solution that detects runtime errors at compile time, without writing or executing any test. Its unique ability to detect errors at the earliest stage of software development process avoids later discovery of annoying software failures during test campaigns, after code freeze or detected by end users. It is a must-have solution for developers overwhelmed by unproductive debugging tasks and who are looking for substantial test costs reduction and high source code reliability improvement. With PolySpace Developer Edition, bugs that used to require hours - or even days - of debug may now be fixed in a matter of minutes thanks to the precise diagnostics given by PolySpace.

"The release of PolySpace Developer Edition will allow us to address the needs of thousands of developers of embedded systems, an activity sector - and market - that grows extremely quickly. PolySpace Developer Edition is definitively a must-have tool to deal with the increase in applications complexity and an ever shortened time-to-market." Said Daniel Pilaud, CEO of PolySpace Technologies.

PolySpace Technologies will be showcasing its latest tools at the Embedded Systems Conference in San Francisco, California, April 23-25, Software Technology Conference, Salt Lake City, Utah, April 29-May 1, and

Embedded Systems Show in London, UK, May 14-15.

PolySpace Developer Edition is available on Windows, Linux and Unix Solaris platforms and analyses codes written in C and Ada development languages.

About PolySpace. In the four years following its creation, PolySpace has confirmed its leadership in static verification of embedded software applications. Relying on a highly innovative mathematical approach, PolySpace provides the first software products capable of performing an exhaustive verification of a program's source code and detecting runtime errors at compilation time. In 2002, PolySpace totalled revenues of over 3 million euros having signed some high-ranking companies in the avionics, aerospace, defence, railways, automotive and energy industrial sectors.

For further information:

<http://www.polyspace.com>. Contact: Eric Pierrel, +33 (0)4 56 38 16 04, [eric.pierrel@polyspace.com](mailto:eric.pierrel@polyspace.com).

## Rational - IBM Completes Acquisition of Rational Software

URL: <http://www-3.ibm.com/software/swnews/swnews.nsf/n/maa5jrkec?OpenDocument&Site=software>

A leading application development tools vendor joins IBM's software business

Armonk, N.Y., February 21, 2003 -- IBM (NYSE:IBM) today announced the completion of its \$2.1 billion acquisition of Rational Software Corp. (NASDAQ:RATL). The acquisition, announced on Dec. 6, 2002, was completed before the market opened today.

Rational provides open, industry standard tools, best practices and services for developing business applications and building software products and systems, including embedded software for devices such as cell phones and medical systems.

"Rational is an important element of our e-business on demand initiative," said Steve Mills, IBM senior vice president and group executive, IBM Software Group. "Rational's complete, open software development platform can improve the speed, quality and predictability of software projects. It's a perfect complement to our existing four brands - WebSphere, DB2, Lotus and Tivoli."

Mike Devlin, who previously was Rational's CEO, is the general manager of the new division reporting to Steve Mills. [...]

Rational shareholders approved the acquisition on Jan. 22, 2003, and the required worldwide government

regulatory approval process has been completed.

Rational, with customers in 89 countries and more than 3,400 employees, estimates that more than 600,000 software developers use its software tools. Rational will be integrated into IBM as the fifth brand of IBM Software Group, a \$13 billion business in 2002. EquiServe Trust Company will be the paying agent for the Rational transaction. [...]

## Rational - Apex Embedded for Tornado 4.2.0 for Sun Solaris to MIPS

From: Eddie Glenn <[cav@Rational.Com](mailto:cav@Rational.Com)>  
Date: Wed, 19 Feb 2003 14:29:23 -0800  
Subject: Rational Apex Embedded for Tornado 4.2.0 for Sun Solaris to MIPS is available by FTP  
To: [apex-announcements@rational.com](mailto:apex-announcements@rational.com)  
<[apex-announcements@Rational.Com](mailto:apex-announcements@Rational.Com)>

[In all Rational URLs below, substitute <rel> by <cross>/releases/sol, <doc> by <cross>/documents/unix/VxWorks, <cross> by <ftp>/apex\_cross/, and <ftp> by <ftp://ftp.rational.com/public/> -- dc]

Product: Rational Apex Embedded for Tornado.  
Version: 4.2.0.  
Platform: Sun Solaris to MIPS.  
URL: <rel>/mips/vw/4.2.0.

This is the Generally Available (GA) release. Follow this link for Rational Apex Embedded for Tornado download and installation instructions. [...]  
<ftp>/standard.msgs/install\_instructions.html

Rational Apex Embedded for Tornado 4.2.0 Release Note for MIPS:  
<doc>/relnote.4.2.dir/vxworks\_relnote\_mips/vxworks\_release\_noteTOC.html  
<doc>/relnote.vxworks.mips.4.2.ps.Z

## Rational - Apex 4.2.0c for Windows NT

From: Greg Bek <[gab@Rational.Com](mailto:gab@Rational.Com)>  
Date: Mon, 14 Apr 2003 13:18:26 +0930  
Subject: Apex NT 4.2.0c Release  
To: Apex Announcements <[apex-announcements@Rational.Com](mailto:apex-announcements@Rational.Com)>

Product: Rational Apex.  
Version: 4.2.0c.  
Platform: Windows NT.  
URL:  
<ftp>/apex/releases/win/apex\_nt.4.2.0c-self\_extract/.

This release has been available for some weeks, but for some reason the announcement wasn't sent. [...]

Greg Bek, Product Manager, Rational Apex Family, Rational Software, IBM Software Group

## RRSoftware - Claw Development

From: Randy Brukardt  
<[randy@rrsoftware.com](mailto:randy@rrsoftware.com)>  
Date: Fri, 21 Mar 2003 14:55:33 -0600  
Subject: Re: gwindows help please  
Newsgroups: comp.lang.ada

The Introductory version of Claw is licensed under the GMGPL. That isn't reflected in the files, but it's true (Adapower has a page devoted to that).

It's true we haven't released a new free version in quite a while, but certainly development is continuing. Recently, we been working mor on Internet stuff (HTTP, FTP) than GUI, but there's quite a bit of new GUI stuff in the pipeline. A new free version should appear in a few months, but I can use a line from RKBD here: it's not a priority over supporting the paying customers.

## Top Graph'X - XInAda Library to be Offered under GPL

From: Jean-Claude Mahieux  
<[jeanclaude.mahieux@topgraphx.com](mailto:jeanclaude.mahieux@topgraphx.com)>  
Date: Fri, 7 Feb 2003 17:50:36 +0100  
Organization: Top Graph'X  
Subject: Re: X11 binding with FSF version of gnat  
Newsgroups: comp.lang.ada

> I am having a bit of a problem with X11 bindings when compiled for OSX (the Intermetrics bindings updated for Linux). [...]

Why not use a full Ada implementation ?  
We plan offering our XInAda Xlib under GPL, but need to allocate some time to do it. The entire XInAda library set includes Toolkit and 1.2 Motif widget set 100% Ada95.

Jean-Claude Mahieux, Top Graph'X Sales Manager, La Norville, France, [sales@topgraphx.com](mailto:sales@topgraphx.com), <http://www.topgraphx.com>, German Representative: [orbriver-de@topgraphx.com](mailto:orbriver-de@topgraphx.com), US Representative: [sroliver@topgraphx.com](mailto:sroliver@topgraphx.com).

[And from a later message: -- dc]

What we plan is to provide XInAda-Xlib as a set of Ada95 library source code. It should be compilable with any Ada95 compiler. We do not plan working in parallel to build specific versions following in particular Gnat versions. The Xlib (including low level required packages) is about 45K lines

From: Jean-Claude Mahieux  
<[jeanclaude.mahieux@topgraphx.com](mailto:jeanclaude.mahieux@topgraphx.com)>  
Date: Mon, 10 Feb 2003 11:06:16 +0100  
Organization: Top Graph'X  
Subject: Re: X11 binding with FSF version of gnat  
Newsgroups: comp.lang.ada

XInAda runs on OS X. Not because it is ported, but XInAda shares part of the IOs with our OrbRiver (CORBA) product. And the orb is supported on OS X because we have customers using it. As the low level packages run fine with the ORB, I do not expect any problem recompiling the rest of the library with Gnat.

## Vector Software - VectorCAST for XGC Compiler

URL:

<http://www.vectors.com/xgcpresrel.htm>

Vector Software Offers Automated Test Tools for Unit Level Testing for Ada, C, and C++ with VectorCAST

Vector Software Announces Integration with XGC Software's XGC Compiler to Provide Automated Software Component Test Capability

North Kingstown, RI - March, 2003 - Vector Software, Inc. announced that it has released VectorCAST for the XGC Compiler. VectorCAST provides an automated module level test capability for users of XGC Software's XGC Compiler, enabling software engineers to test individual software components prior to system level test. VectorCAST for XGC automates building of complete test harnesses for each software component, constructing of test cases, execution and viewing of pass/fail results and code coverage information for each test executed. Further capabilities allow for complete regression testing as components are modified.

XGC Software is a provider of custom and off-the-shelf compilation systems with high performance run-times designed for aerospace applications. [See also "Finland - On-Board Software for GOCE Satellite & Static Analysis Tool" in AUJ 23-3 (Sep 2002), pp.153-154, and "Ada Cross Compilers for 1750 Processor" in AUJ 22-4 (Dec 2001), pp.199-200. -- dc]

Vector Software develops, markets and supports the VectorCAST line of automated component level test tools. The VectorCAST products scan your C, C++, and Ada83/95 source code and automatically generate the test harnesses necessary to test software components in isolation. Utilities are included to construct and execute test cases, generate reports comparing expected and actual results, perform automated regression testing, code coverage, complexity measures and basis path analysis. [See also "Vector Software - VectorCast Automated Test Tools" in AUJ 23-3 (Sep 2002), p.148. -- dc]

## Vector Software - Testing Software Travels with Ada

URL: <http://www.adaic.org/news/test-sw.html>

Date: Fri, 25 Apr 2003 22:38:10 +0200

"Vector Software Easily Writes Fail-Safe Tests in Ada" by Ann S. Brandon

Here's a developer's choice: Easily write software in Language A that doesn't crash, vs. Language C, and easily write software that crashes. Vectors Software chose Language A, which stands for Ada, rather than Language C, which stands for, well, C.

"With C, it's very difficult to put in general purpose guards to protect yourself from unintentional crashes," said the testing software company's director of Engineering, John Paliotta. "In Ada, it's the reverse. It's hard to crash an application at the operating system level, and easy to put in exception handling that gracefully handles the error."

Vector Software, Inc., dedicated the brains and guts of its software, VectorCAST, to Ada in 1993. The software automatically generates and compiles code to test C/C++ and Ada components across a comprehensive list of compilers and environments. Test cases are based on users' requirements and include ultimate values from lowest to highest, as well as fixed increments in between. The tests can be used on the host, a simulator, or an embedded target.

Paliotta says that in the ten years the company has been maintaining and changing its Ada software, they have found the code "almost impossible to crash." VectorCAST's GUI is in C++, which has given the software engineers many opportunities to compare the two languages as tools.

"Ada's exception handlers are better than C++," Paliotta said, explaining the absence of random crashes. "They're general purpose. With C++ they do not by themselves handle anomalous conditions. If at some point I divide by zero -- not intentionally, yet it happens all the time -- no pre-defined mechanisms with C++ can handle it. With Ada, it's built into the language. If you install a small general purpose exception handler it will handle that. Regardless of what bad thing happens it'll be caught and you'll have a defense against an unanticipated anomaly."

Altogether, Paliotta says his department finds the Ada "standard is more uniform and enforced across the vendors and compilers. In the C++ world everyone has their own variant." He acknowledges that C++ has an ANSI standard, but which allows "all the compiler vendors to have their own standard." The basis of Vector Software's choosing Ada can be summed

up, he says, as "The technology is superior."

See <http://www.adaic.org> for more on Ada. Or write to Ann Brandon, Communications Director, Ada Resource Association, [abrandon@sover.net](mailto:abrandon@sover.net).

## Zephyr Basecamp and Monkey Business Enterprise System

URL:

<http://www.redrocketconsortium.com/zbc>  
Date: Tue, 26 Feb 2002 17:34:33 GMT

In June of 2001 I started a project to build an application that would allow small and midsize businesses the same level of inventory, order entry, sales tracking and accounting functionality as would be expected from commercial applications. I wanted it Open Source (GPL, specifically), cross-platform, and international. I wanted to make it possible for the little guy who can't afford the \$60,000 for a commercial application and the yearly annual license fees to bring his business into the modern age.

The working result ended up as Zephyr Basecamp and The Monkey Business database. Zephyr Basecamp and Monkey Business can be downloaded, installed and used at no cost. If a technical support contract is desired, that can be obtained as well. [...]

So, if you have a business and would like to automate, are a programmer or noodler and want to give it a spin, are insanely bored and want something to play with, go ahead to the download area and give it a whirl. Let me know what you think. Naturally, anybody who would like to help me out in this fool's errand is eagerly welcomed to hop on board. We can use documentation (of course!), bug fixes, additions, or anything like that, so there is room for all. The database should work on all PostgreSQL versions above 7.1 and the User Interface works on Windows and Linux. Since it is written in Ada, I would bet that it will work on any platform where GNAT and the various libraries function. [...]

---

## Ada and Linux

### GNAT 3.14p RPM Packages

From: Oleksandr Havva  
<[alex@Lviv.Bank.Gov.UA](mailto:alex@Lviv.Bank.Gov.UA)>

Date: Tue, 04 Mar 2003 09:45:48 +0200

Subject: GNAT 3.14p RPM's

To: GNAT Discussion List

<[gnatlist@lyris.seas.gwu.edu](mailto:gnatlist@lyris.seas.gwu.edu)>

At present time I have built the ALT like RPM's for the GNAT version 3.14p. So I hope it will be interesting for somebody else.

*From: David C. Hoos*  
*<david.c.hoos.sr@ada95.com>*  
*Date: Wed, 5 Mar 2003 14:35:07 -0600*  
*Organization: ENST, France*  
*Subject: ANN: gnat-3.14p rpms available*  
*for Red Hat 7*  
*Newsgroups: comp.lang.ada*

The following files were kindly provided by Oleksandr Havva and are now hosted at ftp.ada95.com/pub/gnat-3.14p/RH-7

asis-3.14p-1.i386.rpm.gz, florist-3.14p-1.i386.rpm.gz,

gnat-3.14p-1.i386.rpm.gz, gnat-3.14p-1.src.rpm.gz,

gnat-3.14p-runtime-1-1.i386.rpm.gz, gnat-glade-3.14p-1.i386.rpm.gz

## Gcc 3 Based GNAT in Linux Distributions

*From: Laurent Guerby <guerby@acm.org>*  
*Date: 04 Mar 2003 12:37:02 +0100*  
*Subject: Re: GNAT 3.14p RPM's*  
*To: GNAT Discussion List*  
*<gnatlist@lyris.seas.gwu.edu>*

Note that Red Hat 8.0 and I assume many other recent distribution have GNAT preinstalled out of box, this is a much welcomed side effect of the ACT CVS GCC GNAT source contribution. I was able to compile and run our company software with it (>500KSLOC) and my GtkAda home software did work too.

It is based on GCC 3.2 and gives slightly better execution times where I tested it (5-20%, I used GCC 3.x tuning flags).

I assume the community will no longer have to package RPMs for the base compiler any more, the add-ons are of course very useful, I hope David Botton will help you set up a public mirror. GNAT users can help speed up the process by checking their favorite distro and asking GNAT to be included if it's not there.

*From: Warren W. Gay VE3WWG*  
*<ve3wwg@cogeco.ca>*  
*Date: Wed, 02 Apr 2003 12:00:27 -0500*  
*Subject: Re: Gnat-Glade 3.2*  
*Newsgroups: comp.lang.ada*

One big plus about the SuSE support of GNAT, is that they have the only publically available binaries for GNAT on the Alpha platform that I know of.

I was able to install and successfully run their GCC-3.1 (with Ada) RPMs on my recent Debian Alpha platform. I hope to make a binary port of GNAT for Debian in the near future, when the GCC-3.x stabilizes a bit more, now that I can bootstrap it. Maybe someone at Debian will do the same.

*From: Warren W. Gay VE3WWG*  
*<ve3wwg@cogeco.ca>*  
*Date: Thu, 03 Apr 2003 12:25:13 -0500*  
*Subject: Re: GNAT for Linux/Alpha*  
*Newsgroups: comp.lang.ada*

> Your hint got me to search for RPM packages for Linux/Alpha with GNAT. I actually found both SuSE and Red Hat packages.

I wasn't aware that Red Hat even supported Alpha. Is this true? I must admit that I didn't look too hard, but I could only find refs to i386 or higher. [...] I don't remember how I found it, but I used some sort of "rpm find" site. [...]

[And from a later message: -- dc]

> I found them by searching for "gnat", "alpha" and "rpm" on Google and noticed that I could use the web site <http://rpmseek.com/> for more precise queries [...]

Yes, rpmseek.com was the one I used. I found that to be very useful here. I'll bookmark it here, before I forget it again ;-) [...]

## RTLGNat - GNAT Port for Real-Time Linux OS

*From: Jorge Real <jorge@disca.upv.es>*  
*Date: Mon, 07 Apr 2003 18:52:29 +0200*  
*Organization: DISCA-UPV*  
*Subject: ANNOUNCE: Ada compiler for*  
*RTLinux*  
*Newsgroups: comp.lang.ada*

RTLGNat version 0.1 is now available from the following web site:  
<http://bernia.disca.upv.es/rtportal/apps/rtl-gnat/>. It works with Gnat 3.14 for Linux and RTLinux 3.1 or later.

>From the README file: "RTLGNAT is a port of the GNAT Ada compiler for the Real-Time Linux operating system. The Ada program is compiled into an RTLinux loadable kernel module, where the program's tasks will run as threads. To start running the Ada program, the generated module must be inserted by means of the Linux service 'insmod'.

As a result, the Ada program is run in the Kernel space with more priority than any other running Linux application."

[And from a later message: -- dc]

> Can RTLGNat also be used to compile Linux kernel modules in general?

No, RTLGNat-generated modules need RTLinux: they use the RTLinux POSIX interface (e.g. for creating threads).

Jorge Real Sáez, Departamento de Informática de Sistemas y Computadores, Universidad Politécnica de Valencia.

---

## Ada and Microsoft

### GWindows 1.1 - Ada 95 Win32 RAD Framework

*From: David Botton <David@Botton.com>*  
*Date: Sun, 9 Mar 2003 23:24:26 -0500*  
*Subject: GWindows 1.1 Released*  
*Newsgroups: comp.lang.ada*

Announcing the release of GWindows 1.1p, the Ada 95 Win32 RAD Framework. March 9, 2003.

[See also "GWindows - Ada 95 Win32 RAD Framework" in AUJ 22-4 (Dec 2001), pp.212-213. -- dc]

GWindows, the Professional Open Source Ada 95 Win32 RAD Framework, introduces for the first time to Ada programming a comprehensive rapid application development framework spanning GUI, Database and Active X integration. It brings AdaPower ;-) to programming domains that up until now are dominated by VB and Delphi.

GWindows includes extensive bindings to the Windows GUI including support for common controls and dialogs, printing, and owner drawn extensions to controls. In addition, GWindows adds a number of new controls, keyboard support, multiple models of event handling, Active X controls, support for creating dialogs and windows from resource files, dynamic garbage collected windows, Window docking, non-GUI bindings, database support, \_database bound controls\_, and much more!

GWindows builds as either ANSI or UNICODE (a first for Ada!) for internationalization and performance boosts on Windows NT, 2000, and XP. GWindows is tightly integrated with GNATCOM, the Ada 95

COM/DCOM/COM+ Development Framework and Tools opening every facet of the Windows platforms to Ada 95 development. Never again will the cries be heard, "but there are no bindings" on the Windows platform!

GWindows is designed to take advantage of Ada's unique combination of features rich typing mechanisms. It is not a thin binding to an underlying C interface, but a complete framework that takes advantage of Ada at every level.

GWindows is being used world wide for production products by governments and large companies to small personal projects.

Version 1.1 of GWindows adds additional stability and improved support for ActiveX controls, a binding to the Scintilla programmers editor control and a Tcl/Tk control.

GWindows is being made available under the GNAT modified GNU GPL used by GNAT's runtime library making it available for use in both GPL and proprietary applications.

For more information on GWindows, to view the on-line documentation, and to download the product, please visit <http://www.adapower.com/gwindows>. Tutorials for GWindows are available at: [http://www.adapower.com/gwindows/user\\_guide.html](http://www.adapower.com/gwindows/user_guide.html) Information on the public version of GNATCOM can be found at

<http://www.adapower.com/gnatcom>. Professional support for GNATCOM is available from Ada Core Technologies, Inc. Please contact [report@gnat.com](mailto:report@gnat.com) for a no-cost GNATCOM evaluation package.

## Bindings to Windows API

*From: Vincent Smeets <No@Spam.org>*  
*Date: Mon, 17 Mar 2003 16:23:51 +0100*  
*Subject: What do I need to write a Windows program with Ada?*  
*Newsgroups: comp.lang.ada*

I know how to write a program with Ada and how to use X-Window (with Motif) for Unix. Now I have a program (small and non-commercial) that I want to port from Unix to a Windows (98 or 2000) environment and give it a Windows-GUI. It will use a menu, an explorer tree and some labels and buttons. The only problem is that I have no idea about programming for the Windows environment. [...]

*From: Stephane Richard*  
*<stephane.richard@verizon.net>*  
*Date: Mon, 17 Mar 2003 15:48:14 GMT*  
*Subject: Re: What do I need to write a Windows program with Ada?*  
*Newsgroups: comp.lang.ada*

I would HAVE to suggest using the Windex Library developed by the one I like to call "The Guru" of Windows Binding, I hereby name Mr. Stephen Leake.

You can get it at [[http://www.toadmail.com/~ada\\_wizard/windex.html](http://www.toadmail.com/~ada_wizard/windex.html) -- dc] It's free and you'll soon notice as you browse the source files that things are very well organized into highly functional groups of objects to make navigation in the library very straight forward and efficient. To me, It's the easiest most natural way to do Windows in Ada and was created by someone who knows what he's talking about :-).

*From: Stephen Leake*  
*<Stephen.A.Leake@nasa.gov>*  
*Date: 17 Mar 2003 12:19:20 -0500*  
*Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)*  
*Subject: Re: What do I need to write a Windows program with Ada?*  
*Newsgroups: comp.lang.ada*

Thanks for the plug. [...] Always nice to be appreciated :). However, Windex does not currently support the tree control. And I've given up developing it, in favor of learning more about / contributing to GtkAda (and supporting my daughter in college, going to anti-war protests, etc :). Windex is GMGPL, so anyone is free to add a tree control to it.

*From: tmoran@acm.org*  
*Date: Mon, 17 Mar 2003 18:31:02 GMT*  
*Subject: Re: What do I need to write a Windows program with Ada?*

*Newsgroups: comp.lang.ada*

There are several [bindings to the Windows API], with different aims and different coverage. I'm one of the authors of Claw - Class Library for Ada on Windows, and suggest you take a look at the TriAda paper available at [www.rrsoftware.com/html/prodinf/triadaper/triada.html](http://www.rrsoftware.com/html/prodinf/triadaper/triada.html)

Claw is a "thick" or "abstract" binding. It's available, as source code, in a GMGPL introductory version and in a supported \$>0 version under active development and with a printed manual as well as Windows Help and html documentation. It's regularly tested under Apex, Gnat, Janus, and ObjectAda compilers. The [www.rrsoftware.com](http://www.rrsoftware.com) site has much more information, including downloads and example programs. [www.adapower.com](http://www.adapower.com) also has a bunch of small programs based on the introductory version of Claw, including a simple web server, SMTP (mail) and HTTP clients, real-time graphics, ActiveX controls by using GWindows together with Claw, a web crawler, and two examples - Minesweeper and Dining Philosophers - of existing programs whose GUI was surgically replaced with Claw while leaving the rest of the program intact. [...]

[Randy Brukardt  
 <[randy@rrsoftware.com](mailto:randy@rrsoftware.com)> added: -- dc]

And, Claw does have a Tree control in the supported version (not in the free version).

*From: David Botton <David@Botton.com>*  
*Date: Mon, 17 Mar 2003 14:23:38 -0500*  
*Subject: Re: What do I need to write a Windows program with Ada?*  
*Newsgroups: comp.lang.ada*

See <http://www.adapower.com/windows> for available Windows bindings, etc.

I recommend GWindows for Win32 - <http://www.adapower.com/gwindows>, see [http://www.adapower.com/gwindows/user\\_guide.html](http://www.adapower.com/gwindows/user_guide.html) for a full tutorial. If you can program Ada, you already know how to write a Windows program :-) [...]

If you are looking to learn the low level APIs you need Petzold's Windows book, ie. a C programming book. I would recommend looking at the GWindows sources as well for best practices in binding to Win32, etc.

To learn GWindows you have a full tutorial in the package and it can be viewed on-line at [http://www.adapower.com/gwindows/user\\_guide.html](http://www.adapower.com/gwindows/user_guide.html) [...]

If you plan on doing Win32 programming long term, you should get a handle on COM (ActiveX) as well by reading InsideCOM and playing with GNATCOM (<http://www.adapower.com/gnatcom>).

## References to Publications

### Endian-Safe Record Representation Clauses

*From: John Harbaugh*  
*<[john.s.harbaugh2@boeing.com](mailto:john.s.harbaugh2@boeing.com)>*  
*Date: Tue, 4 Mar 2003 16:38:33 GMT*  
*Organization: The Boeing Company*  
*Subject: Re: Endianness independence*  
*Newsgroups: comp.lang.ada*

If you are looking for an implementation that is platform and compiler independent, take a look at the December 1999 Ada Letters. Mike Mardis wrote a very good article titled "Endian-Safe Record Representation Clauses for Ada Programs."

### New SPARK Book - Sample Chapters On-Line

*From: rod.chapman@praxis-cs.co.uk (Rod Chapman)*  
*Date: 11 Mar 2003 08:26:53 -0800*  
*Subject: ANN: SPARK Book - Sample Chapters now free on-line*  
*Newsgroups: comp.lang.ada ,*  
*comp.realtime, comp.software-eng,*  
*comp.lang.eiffel, de.comp.lang.misc*

I'm pleased to announce the availability of the "Sample Chapters" of the forthcoming new book "High Integrity Software: The SPARK Approach to Safety and Security" by John Barnes. The sample chapters are freely available in PDF format on [www.sparkada.com](http://www.sparkada.com)

The book is now at the printers. We expect a UK publication date of 21st March. Availability is expected in the USA approximately 4 weeks later. The book can be pre-ordered now at various book-sellers including [www.amazon.com](http://www.amazon.com) and [www.amazon.co.uk](http://www.amazon.co.uk).

All the best, Rod Chapman, SPARK Team, Praxis Critical Systems Limited

### Ada Distilled - On-line Book Updated

*From: Richard Riehle*  
*<[richard@adaworks.com](mailto:richard@adaworks.com)>*  
*Date: Tue, 11 Mar 2003 23:37:42 -0800*  
*Organization: AdaWorks Software Engineering*  
*Subject: Announcement: Ada Distilled*  
*Newsgroups: comp.lang.ada*

The newest revision of Ada Distilled has been sent to <http://www.adaic.org> & <http://www.adapower.com>

This revision has some new programs, particularly for access type examples. Also, there is an accompanying ZIP file with the source code for most of the programs in the book.



As usual, Ada Distilled is free, but I do thank those who have sent a small honorarium for earlier editions.

Richard Riehle, AdaWorks Software Engineering, Salinas, CA, USA  
<http://www.adaworks.com>

## Crosstalk - Call for Articles on Real-Time Programming

From: Richard Riehle

<[richard@adaworks.com](mailto:richard@adaworks.com)>

Date: Sun, 16 Mar 2003 21:23:39 -0800

Organization: AdaWorks Software Engineering

Subject: Crosstalk: Call for Articles on Real-time Programming

Newsgroups: [comp.lang.ada](mailto:comp.lang.ada)

I noticed in the Call for papers section of Crosstalk a request for papers on real-time programming for the November 2003 issue. This is another opportunity for someone to write an article on Ada, preferably a case-study with metrics, etc.

The Crosstalk web site is:  
<http://stsc.hill.af.mil>

## CACM - On Type-Safe Languages

From: Tom Moran <[tmoran@acm.org](mailto:tmoran@acm.org)>

Date: Wed, 2 Apr 2003 04:11:08 GMT

Subject: CACM Risks re Ada

To: [team-ada@acm.org](mailto:team-ada@acm.org)

The Risks column in the current Communications of the ACM is titled "On Sapphire and Type-Safe Languages". It talks about buffer overrun as an abstraction failure and says "Some modern languages (for example, Ada, C#, Common Lisp, Eiffel, Java, Modula 3, Scheme, and Standard ML) prevent this failure of abstraction."

## DDC-I Online News

[Extracts from the table of contents. See elsewhere in this news section for selected items. -- dc]

From: [jc <jcdk@ddci.com>](mailto:jcdk@ddci.com)

Date: Mon, 7 Apr 2003 11:27:25 -0700 (MST)

Subject: Online News - [...]

To: Q9 March\_April2003 Online News DK <[jcdk@ddci.com](mailto:jcdk@ddci.com)>

DDC-I Online News. Real-Time Industry Updates - News from DDC-I. March/April 2003, Volume 4, Number 3 - [[http://www.ddci.com/news\\_vol4num3.shtml](http://www.ddci.com/news_vol4num3.shtml)]. A monthly news update dedicated to DDC-I customers & registered subscribers.

The Safety Solution... Certified. Making its Debut at the Software Technology Conference, April 29th, Salt Lake City, UT BAE SYSTEMS Controls, DDC-I and Enea TekSci Join Forces to Offer A Powerful, Proven Solution for Developers

of Software Where Safety-Critical Behavior is Paramount.

DDC-I Releases Upgraded JOVIAL Programming Tools. Upgrades Aimed at Increasing Project Efficiency.

TADS-i960 Now Available for Windows NT, 2000 and XP Host Environments. A Direct Response to the Needs of Customers Migrating to Popular Platforms.

Third Party Update: Software Certification from Enea TekSci. A Guideline for Certifying Software in Safety Critical Avionics Systems.

Due to technical difficulties, the emailer describing the February issue of the DDC-I Online News did not go out. The following is a brief description of articles available in that issue and the corresponding links.

Full Support for Windows Hosted TADS-1750A Product Line. A Direct Response to the Needs of Our Customers Migrating to Popular Platforms.

Cost-Effective Windows Migration Package for TADS-1750A Customers. Customer Define Package Parameters to Create a Least-Cost Migration Path.

Tech Talk: Verifying PROM'ed Systems by Checksums. Including System Checksum Information Increases System Reliability.

For this months complete newsletter: [http://www.ddci.com/news\\_current\\_issue.shtml](http://www.ddci.com/news_current_issue.shtml). For the complete February issue: [http://www.ddci.com/news\\_vol4num2.shtml](http://www.ddci.com/news_vol4num2.shtml)

## Downloadable HTML Version of Ada Reference Manual Updated

From: Randy Brukardt

<[randy@rrsoftware.com](mailto:randy@rrsoftware.com)>

Date: Wed, 16 Apr 2003 15:07:53 -0500

Subject: Re: Ada language help by reserved keyword

Newsgroups: [comp.lang.ada](mailto:comp.lang.ada)

I just updated the downloadable HTML versions of the ARM to include the search page and the links to it. There also were a few fixes intended to make the ARM validate through the WC3 validator (which I didn't know about that the time that the original HTML was created).

[<http://www.ada-auth.org/~acats/arm.html> -- dc]

## Ada Book Recommendations

From: Alex Gibson <[alxx@ihug.com.au](mailto:alxx@ihug.com.au)>

Date: Sun, 27 Apr 2003 00:30:07 +1000

Subject: Best Ada books?

Newsgroups: [comp.lang.ada](mailto:comp.lang.ada)

Okay this is always a bit of a subjective question asking about programming books but usually worth asking anyway.

If you had a choice of two Ada books which would it be. [...]

[See also same topic in AUJ 22-4 (Dec 2001), pp.214+216. -- dc]

[Georg Bauhaus <[sb463ba@d2-hrz.uni-duisburg.de](mailto:sb463ba@d2-hrz.uni-duisburg.de)> responded:]

Have you been looking at <http://www.adapower.com/books/>? There are some links to reviews and online books as well.

[David C. Hoos

<[david.c.hoos.sr@ada95.com](mailto:david.c.hoos.sr@ada95.com)>:]

Ada As a Second Language, by Norman Cohen, ISBN: 0070116075.

Concurrency in Ada, by Alan Burns and Andy Wellings, ISBN: 052162911X.

[...] Cohen's book is my favorite Ada text book. But note that it is aimed at experienced programmers.

Beginners enjoy John English's book which is online at <http://www.cmis.brighton.ac.uk/~je/adacr aft/>

[[john@assen.demon.co.uk](mailto:john@assen.demon.co.uk) (John McCabe):]

[... Cohen's] is the only Ada 'learning the language' book that I ever bought.

I quite like Michael Feldman's "Software Construction and Data Structures with Ada" (ISBN: 0201887959), but don't follow the coding style :-)

[[tmoran@acm.org](mailto:tmoran@acm.org):]

> Programming In Ada 95 by Barnes, Second edition.

An excellent book, and usually the first one I turn to. Though the Cohen and the Burns and Welling books I also keep within reach.

---

## Ada Inside

### USA - LJK/CDROM, CD Formatting and Verification

From: Kilgallen@SpamCop.net (Larry Kilgallen)

Date: 1 Feb 2003 12:51:51 -0600

Organization: LJK Software

Subject: Re: Bye-bye Ada ?

Newsgroups: [comp.lang.ada](mailto:comp.lang.ada)

> I'm still missing Ada applications at all (in the public). I never heard about a CD-burning app in Ada, nor of an .wav or .bmp manipulation app, nor something else (except GNAT and GVD).

Please don't confuse not having Open Source software for a given domain written in Ada with not having any software for that domain written in Ada.

[http://www.ljk.com/ljk/ljk\\_cdrom.html](http://www.ljk.com/ljk/ljk_cdrom.html)

[From that URL: "This software can be used to format and verify volumes

according to ISO-9660 and related standards." -- dc]

## Germany - CDBurn, CD Writing for RISC OS

*From: news0203@nerdnet.de (Jan-Uwe Finck)*

*Date: Sat, 1 Feb 2003 20:54:55 +0100*

*Subject: Re: Bye-bye Ada ?*

*Newsgroups: comp.lang.ada*

Well, at least a CD-burning software exists. although not under the usual OS, but under Risc OS (<http://www.riscos.com> and [.org](http://www.org)), called CDBurn (<http://www.huber-net.de/cdburn.htm>), the author mentions his use of Ada on his website here (<http://www.huber-net.de/ada.htm>), unfortunately in German.

Yes, people use Ada.

*From: Steffen Huber*

*<steffen.huber@gmx.de>*

*Date: Sun, 02 Feb 2003 16:17:06 +0100*

*Subject: Re: Bye-bye Ada ?*

*Newsgroups: comp.lang.ada*

> I never heard about a CD-burning app in Ada,

You have not looked hard enough ;-) I have written one, called CDBurn. For more information, look at <http://www.huber-net.de/cdburn.htm> However, unless you are lucky enough to run RISC OS, it makes little sense to buy it ;-)

[And from another message: -- dc]

> Fine, will you port it to other OS's?

The whole thing: unlikely. There is very little that CDBurn can do that would make it better than the competition - with the exception of the user interface, but its niceness is heavily bound to RISC OS' way of operation.

The only part that would possibly make sense is the whole ISO/Joliet image creation stuff. Mainly because this would also port the RISC OS filer to Windows, and most of the software available on Windows is a lot less flexible when it comes to filename conventions etc.

Anyway, I am currently happy supporting the few thousand CDBurn users on RISC OS, and it is likely that CDBurn will morph into DVDBurn in the future. If I am ever getting bored, I will think about a port.

## USA - FAA Certifies INTEGRITY RTOS for Sikorsky S-92 Helicopter

*URL:*

*<http://www.ghs.com/news/230210r.html>*

FAA Certifies INTEGRITY RTOS for DO-178B, Level A Use In Sikorsky S-92 Helicopter

Green Hills Certification Package Key to Fast Approval

[See also "USA - Green Hills for New Sikorsky S-92 Helicopter" in AUJ 23-3 (Sep 2002), pp.152-153. -- dc]

Santa Barbara, CA. February 10, 2003 -- Green Hills Software, Inc., the embedded technology leader and second largest RTOS company, today announced FAA acceptance of a DO-178B, Level A, certification package for its INTEGRITY-178B real-time operating system (RTOS). The INTEGRITY certification package was delivered to Rockwell Collins for use in Technical Standards Order certification for a new avionics system aboard the Sikorsky S-92 helicopter.

FAA audit of the certification package was completed at Green Hills Software's offices in November 2002. The audit process verified compliance with DO-178B, Level A requirements, a required step for certification of the S-92 aircraft.

The S-92 is Sikorsky's newest medium-lift commercial helicopter. Featuring a passenger capacity of 19-22, the versatile new helicopter will serve a variety of commercial and international utility needs, including passenger, cargo, aeromedical, search and rescue and resource development support. Production is already underway with final assembly commencing in March 2003.

"The FAA's acceptance of our Level A certification package is very significant," said John Carbone, vice president of marketing for Green Hills Software. "INTEGRITY-178B has been approved for use in the most safety-critical applications sanctioned by the FAA. What's more, our complete certification package for DO-178B, Level A is available today to all developers seeking Level A certification. This gives INTEGRITY a clear advantage over other commercial RTOSes wherever DO-178B, Level A certification is required."

More on DO-178B [see URL above. -- dc]

About Green Hills Software, Inc. [See earlier announcements from Green Hills in the AUJ. -- dc]

For More Information Contact: Green Hills Software, Lynn J. Robinson, [lynnr@ghs.com](mailto:lynnr@ghs.com); Davis-Marrin Communications, Ken Marrin, [kmarrin@davismarrin.com](mailto:kmarrin@davismarrin.com).

## USA - TSim Satellite Terminal Simulator

*From: Vinzent Hoefler*

*<JeLlyFish.software@gmx.net>*

*Date: Fri, 14 Feb 2003 19:23:57 -0500*

*Organization: JeLlyFish software*

*Subject: Re: Ada sockets and gnat 3.13 windows and linux interoperability*

*Newsgroups: comp.lang.ada*

On SIGAda 2002 a paper was presented for a Satellite Terminal Simulator called TSim. They used gnat, Tcl/Tk, TASH, AdaSockets and the Booch Components and in the paper they state the following:

"Throughout Round-3, we managed to retain the source code compatibility between Win32 and Linux. So, we could also build distributed TSim under Linux, and operate exactly the same way. Miraculously, we also found that the Satellite partitions were interoperable under Win32 and Linux. That is, we could run the Satellite partition under Linux, and the Terminal partition under Win32, with TC connected on the same or a different machine. Or, we could run the Satellite partition on Win32, and the Terminal partition under Linux, connected to TC on Win32 - possibly the same machine running the Satellite server. This was truly unexpected, and gave us something technically satisfying."

[...] they used the Distributed Systems Annex, [...]

[David C. Hoos, Sr. <david.c.hoos.sr@ada95.com> also wrote:]

The Distributed Systems Annex goes so far as to provide endianness-independent stream attributes, so that the types will be accurately transferred even between systems with different endianness.

## USA - Astrology Program

*From: Ed Falis <falish@adelphia.net>*

*Date: Sun, 16 Feb 2003 00:19:59 GMT*

*Subject: Re: No Ada in OpenSource?*

*Newsgroups: comp.lang.ada*

Astrology program:

<http://users.adelphia.net/~falish/>

## USA - Latin-to-English Dictionary

*From: Warren W. Gay VE3WWG*

*<ve3wwg@cogeco.ca>*

*Date: Mon, 17 Feb 2003 12:31:44 -0500*

*Subject: Re: No Ada in OpenSource?*

*Newsgroups: comp.lang.ada*

On the weekend I encountered a text mode application that performs "Latin-to-English Dictionary" lookups, and was encouraged to see that it was written in Ada. While this is "free shareware" (with sources available), it is one of many obscure examples of applications throughout the wonderful www. See: <http://users.erols.com/whitaker/words.htm>

[See also "Latin translation program" in AUJ 20-1 (Apr 1999), p.41. -- dc]

If you know where to look, you can find Open Sourced Ada in all corners of the world-wild-web.

## Germany - Controlling Digital Cameras

From: Thomas Preymesser <tp@lehele.de>  
 Date: Tue, 18 Feb 2003 15:58:13 +0100  
 Subject: Re: Ada-Entwicklung, Beteiligung an Projekten  
 To: ada-list@ada-deutschland.de

[Extracts translated from German. -- dc]

I wrote a few smaller things (see <http://www.thopre.de/Ada.html>), which are also still under development and I would be pleased with some participation (e.g. controlling of digital cameras). [...]

## Germany - PL/SQL Developer Moves to Ada

From: Frank Piron  
 <frank.piron@konad.de>  
 Date: Fri, 21 Feb 2003 09:47:06 +0000  
 Organization: KonAd GmbH  
 Subject: New in Ada, written some programs and earned money!  
 Newsgroups: comp.lang.ada

I'm new in Ada, really new, I started to learn the language 3 months ago.

My previous experiences in programming were in pure C and ORACLE PL/SQL 2.3 - much like Ada83 I think. 4 years ago we started development of a complete control system for a police department in switzerland. It is ORACLE-based, contains a workflow engine, document management, GUI in ORACLE-Forms, much PL/SQL-Code inside the Database, much exception handling.

The demands of the customer concerning stability and security were very strict and we expensively used the exception handling mechanism of PL/SQL. But this Language is pretty restricted in respect to modern software engineering methods and last year meanwhile in the own company I encountered Ada95.

I read some books (Barnes, Feldman, ...) and started to play around and I felt good with my PL/SQL-experiences.

3 weeks ago I delivered my first little programs in Ada which I used to write in C earlier, when the capabilities of the poor ORACLE-Client PL/SQL-engine did not suffice.

The programs were simple helpers, one parsing a file written in a proprietary format, one calling some win32 api functions. But I was curious to know more about string handling in Ada and interfacing with C.

Great success! Simple development, no errors till now. Ok, I had my fight with the compiler, but it was productive.

I think the Ada community should try to get the attention of the many many PL/SQL-Developers in the world. Ada is an ideal way to enrich the programming skills and possibilities for them.

## Europe - Eurofighter Crash due to Hardware Failure

From: Peter Amey <peter.amey@praxis-cs.co.uk>  
 Date: Fri, 28 Mar 2003 08:30:22 +0000  
 Subject: Re: Eurofighter DA6 crash  
 Newsgroups: comp.lang.ada

> Last year an Eurofighter crashed in Spain. Both engines failed. It looks like an sw problem. Ada/SPARK involved?

I have no privileged information about the crash itself but can say that the engine control software is not written in SPARK. In any case, speculation before the investigation is complete is usually a waste of time.

From: John Cook  
 <Jwcook@ozemail.com.au>  
 Date: Sat, 29 Mar 2003 15:21:10 +1100  
 Subject: Re: Eurofighter DA6 crash  
 Newsgroups: comp.lang.ada

You may wish to look at <http://www.eurofighter.starstreak.net/forum/viewopic.php?t=585> for all the information publicly available.

The reason was given there was a conflict between the software and the engines fitted in the Development aircraft. (They were earlier versions of the production standard)

Eurofighter Website:  
<http://www.eurofighter-typhoon.co.uk>

From: svaa@ciberpiula.net (svaa)  
 Date: 29 Mar 2003 13:17:12 -0800  
 Subject: Re: Eurofighter DA6 crash  
 Newsgroups: comp.lang.ada

Engine failure.

[<http://www.eurofighter.com/News/Article/default.asp?NewsItemID=116> and 117 - dc]

According to Iblnews (in Spanish): <http://iblnews.com/news/noticia.php3?id=70853>

"Hoax about software failure appeared a few minutes after the crash." "That shows how much people expect software failure. Software failure is an easy escape goat".

## Greece - Objektum to Deliver HOOD courses to Greek Air Force

From: Jackie Bissell  
 <jackie@realitypr.co.uk>  
 Date: Mon, 31 Mar 2003 11:55:29 +0100  
 Subject: PRESS RELEASE - Hierarchical Object-Oriented Design (HOOD) is not all Greek to training and consultancy expert Objektum

Press Release: Hierarchical Object-Oriented Design (HOOD) is not all Greek to training and consultancy expert Objektum

Objektum appointed by defence manufacturer ISI Hellas to deliver a series

of courses designed to develop key defence systems for the Greek Air Force

Date: 31 March 2003, Objektum ([www.objektum.com](http://www.objektum.com)), the training and consultancy expert for the defence industry, has designed an integrated training course on Hierarchical Object-Oriented Design (HOOD), using the STOOD toolset (the design tool for real-time and high quality software developments) to develop a series of major projects for the Greek Air Force.

This specialist course has been commissioned by ISI Hellas, the Greek manufacturer of real-time Command and Control, Data Link Systems and real-time Training Simulators. It will be used in a number of major projects, including the Mission Training System (MTS), for the Greek Air Force's Early Warning and Control Aircraft, and the KDIL project, a digital information link for connecting medium range Surface to Air missile systems.

Leading ISI Hellas developers will attend the 5-day training programme presented by Derek Russell, Objektum's Technical Director and foremost expert on both the HOOD methodology and STOOD toolset.

"The skills gained from the Objektum training would be used to develop new systems designed for the Greek Air Force." Said David Tham ISI Hellas' Technical Director. "It would also be used on training simulators, currently in their final stages of development, to identify common components for re-use in other projects." He added.

"We have designed this very specific series of integrated training courses using STOOD a specialist toolset which fits perfectly with HOOD." Said Derek Russell, Objektum's Technical Director. "ISI Hellas was very clear on its training requirements. In the past we have delivered Ada training to them and, with our defence systems expertise, we are able to fully support its developers in meeting its overall project objectives for the Greek Air Force." He added.

[See URL for more info on Objektum: -- dc] [www.objektum.com](http://www.objektum.com)

For more information or photography please contact: Andy Brown, Reality PR, 020 8663 6111, [andy@realitypr.co.uk](mailto:andy@realitypr.co.uk)

## USA - Trash Finder, Anti-Spam Filter

From: Randy Brukardt  
 <randy@rrsoftware.com>  
 Date: Wed, 16 Apr 2003 15:01:09 -0500  
 Subject: Re: If anybody wants to make something in Ada but do not know what  
 Newsgroups: comp.lang.ada

What kind of spam filter are you talking about? A filter for a server is different in a number of ways than a filter for a mail

client. And a filter for an ISP or large company is different than a filter for a tiny organization.

That said, an anti-spam filter written in Ada already exists: it's called Trash Finder, and it works with the IMS mail server on Windows. I haven't publicized it here precisely because no one here can use it. :-) It is of course 100% in Ada, and it filters for literally dozens of criteria -- after fully decoding and unfolding the message (a significant percentage of spam is encoded). Among other things, it filters on character sets, attachment types, violations of RFCs in the mail format (spammers have a hard time following RFCs), specific HTML features (forms, scripts, graphics, text outside of the markup, etc.), From, To, Subjects, Text (without the HTML markup, which often can be used to hide things), HTML markup, and (most recently) domains in URLs given either in HTML markup or text.

It filters about 98% of the incoming spam on my system. [...]

[And from later messages: -- dc]

[...] The word list for the AdaIC search engine is 270,000 words, and it takes up 6 MB. [...] some of the people running Trash Finder have servers handling 50,000 messages per day with more than 1,000 users. [...]

> Is Trash Finder difficult to set up?

No, just add it to the list of plugins and bounce the services. To be most effective, it needs some filter patterns entered (the domains of persistent spammers), but there is a nice GUI (written in Claw, of course) for doing that. And it will get a lot of junk out-of-the-box.

Of course, Trash Finder has been publicized many times on the IMS-Users mailing list. [...]

## Canada - CMC Electronics Completes Control Display Unit Using ObjectAda/Raven

URL:

[http://www.aonix.com/pr\\_04.21.03.html](http://www.aonix.com/pr_04.21.03.html)

Aonix Announces the Completion of the Control Display Unit Using ObjectAda/Raven by CMC Electronics Inc.

San Diego, California, April 21, 2003

Aonix a leading provider of Ada 95 software development environments, is pleased to announce the successful completion of the CMA-2082D Control Display Unit by CMC Electronics Inc.

CMC Electronics Inc, has completed development and is in the final stages of integration of the CMA-2082D Control Display Unit (CDU) using ObjectAda/Raven for the Canadian

Department of National Defense (DND) CP140 Maritime Patrol Aircraft.

The Canadian Department of National Defense (DND) operates a fleet of eighteen (18) CP140 Aurora aircraft as a long-range maritime patrol platform for surface and undersea surveillance roles. The aircraft design, which entered into service in 1980, is based on the United States Navy (USN) Model P-3C Update II Orion aircraft for the main airframe and engines, with mission avionics mostly based on the USN S-3A Viking mission avionics suite. The CP140 Aurora aircraft is undergoing a mid-life improvement program, referred to as the CP140 Aurora Incremental Modernization Project (AIMP), and includes a series of incremental upgrades over a ten-year period. CMC Electronics is the prime contractor for the Navigation and Flight Instruments Modernization Project (NFIMP) and developer of the CDU, which constitutes the integration and airworthiness certification of new NFI avionics equipment. The upgraded NFI System includes an Avionics Management System (AMS) consisting of four CDUs, two Embedded Global Positioning System (GPS)/Inertial Navigation System (INS) (EGI), Electronic Primary Flight Instruments (EFDS), Autopilot and Flight Director System (AFDS), Radar Altimeter and Altitude Warning System (RAAWS) and Traffic Collision Avoidance System (TCAS).

"CMC Electronics originally selected the ObjectAda/Raven product for use on the CMA-2082D AMS in 2000 following an extensive evaluation of Ada95 tool vendors. Aonix was selected for a variety of reasons including the usability of the tool set and the technical support that they were able to provide," states Kevin Sanford, Software Project Lead, CP140 NFIMP Project for CMC Electronics. "The main discriminator, however, was the safety-critical domain experience of the company and their ability to meet the standards required for incorporation into a DO-178B software system."

The CDU will control the avionics 1553 Data bus and act as a remote terminal on the Mission 1553 Data bus, under the control of a Mission Computer. The CDU will provide the Pilot, Co-pilot, Navigator-Communicator and Tactical-Navigator a means to control and monitor the operation of the two EGIs, provide area and terminal navigation solutions, and provide flight guidance generated by the CDU from an operator defined flight plan and the EGI outputs. The CDU navigation outputs will then be displayed on the EFDS and used as a steering source by the AFDS.

About Aonix & CMC Electronics [see URL above and also "Aonix – Grows Into

Two New Companies" in the "Ada Products" section of this AUJ. -- dc]

Press Contacts: Greg Gicca, 1-603-429-3415, [gicca@aonix.com](mailto:gicca@aonix.com); Additional Product Information: 1-858-457-2700, [info@aonix.com](mailto:info@aonix.com). [...]

## Indirect Information on Ada Usage

[Extracts from and translations of job-ads and other postings illustrating Ada usage around the world. -- dc]

From: Hannes Birnbacher

<[usenet@hannes-birnbacher.de](mailto:usenet@hannes-birnbacher.de)>

Date: Sat, 1 Feb 2003 01:41:26 +0100

Subject: Re: Ada and LFS

Newsgroups: [comp.lang.ada](http://comp.lang.ada)

[...] I am co-operating with an Ada programmer to produce a German open-source tax software, and I am supplying the tax know-how. So, to check out the code, I had to install Ada somehow, without knowing anything about it, after I had my Linux from scratch running for months. I also installed GRASP, which is great for my purposes. [...]

From: David Marceau

<[davidmarceau@sympatico.ca](mailto:davidmarceau@sympatico.ca)>

Date: Sat, 01 Feb 2003 15:13:54 -0500

Subject: Ada job opportunity posted at

Thales in Ottawa Citizen Today

Newsgroups: [comp.lang.ada](http://comp.lang.ada)

The Ottawa Citizen Web site doesn't display it, nor does the Thales Group web site ([www.thalesgroup.com](http://www.thalesgroup.com)) display it but it's in Today's printed edition of the "Ottawa Citizen" Newspaper. [...] the job is in Ottawa it seems. [...]

From: Anders Wirzenius

<[anders.wirzenius@pp.qnet.fi](mailto:anders.wirzenius@pp.qnet.fi)>

Date: Sun, 02 Feb 2003 09:51:13 GMT

Subject: Re: Bye-bye Ada?

Newsgroups: [comp.lang.ada](http://comp.lang.ada)

Just a hint: Join a company where programming is not the core business. If the way to achieve a solution is not an issue, hence, you might be free to choose your own means to get the work done.

I work for a company in the heavy steel business. I don't work full time programming, but almost all my programming is done with Ada. I produce mainly reports from databases and dynamic intranet pages. I keep a rather low profile, I don't overemphasize the means how I achieve my goals, so I have managed to keep my "private" small workshop with Ada. Not a glorious situation, but at least I can do my Ada coding...

From: [volkert@nivoba.de](mailto:volkert@nivoba.de) (Volkert)

Date: 2 Feb 2003 06:58:56 -0800

Subject: Misleading Compiler Warning

Newsgroups: [comp.lang.ada](http://comp.lang.ada)

The snippet is reproduced from a large application we currently bringing from DEC Ada 83 to GNAT (OpenVMS).... the

GNAT 3.16w gives a Compiler-Warning whereas the DEC Compiler keeps silent. [...]

*From: Rick Morneau <ram@eskimo.com>  
Date: 4 Feb 2003 12:03:58 GMT  
Subject: Re: Bye-bye Ada?  
Newsgroups: comp.lang.ada*

I've been working on a project that you might find interesting, and which I'm programming in Ada. Over the past dozen-or-so years, I've been developing an interlingua for use in machine translation. Over the past two years, I've written and tested the first draft of the Ada software that translates from the interlingua to English, with extremely good results. In the process, I learned a lot and I'm now starting a complete redesign of both the interlingua (should be done in about 1-2 weeks) and the software. [...]

Information about the first version of the interlingua along with many examples of machine translations can be found here: <http://www.eskimo.com/~ram/Katanda/> or <http://www.srv.net/~ram/Katanda/>

If you're interested, please feel free to contact me at ram at Eskimo dot com. By the way, this project is a (very serious!) hobby of mine, but I'm retired, so I have lots of time to work on it :-)

*From: Robert C. Leif <rleif@rleif.com>  
Date: Tue, 4 Feb 2003 08:25:56 -0800  
Subject: RE: Bye-bye Ada?  
Newsgroups: comp.lang.ada*

Firstly, Newport Instruments is a user of Ada technology, rather than an Ada company. However as both a small user and formally the corporate fellow of a large company, which created a medical device in Ada, I have my own perspective on the Ada market. [...]

*From: Richard Riehle  
<richard@adaworks.com>  
Date: Thu, 06 Feb 2003 19:33:17 -0800  
Organization: AdaWorks Software Engineering  
Subject: Re: Bye-bye Ada?  
Newsgroups: comp.lang.ada*

If Ada works for you, you might want to keep it a secret. I know at least one company that refuses to let anyone know it is using Ada, even though they are happy with it and it is contributing to their success.

[And on the question "So how does this "stealth" company recruit Ada personnel? And how do Ada people find the company?" -- dc]

I done one seminar for them. All the rest, as far as I can tell, is done within their own organization. I am not allowed to discuss the details.

[AG <ang@xtra.co.nz> responded to the same question: -- dc]

Well, one way that I know of is: The company would advertise for, say, C++ programmers (\*not\* stressing the

language) plus require a wide range of experience in other languages and general programming/SE expertise. During the interview, if you make it that far, you may get asked what other languages you actually know and, if you mention Ada (or any other not too popular language for that matter) asked to prove it ...

*From: Marin David Condic  
<mcondic@acm.org>  
Date: Wed, 12 Feb 2003 07:38:26 -0500  
Subject: Re: [OT] Just taunting us, Marin? :-)  
Newsgroups: comp.lang.ada*

[...] All I can say for sure is that the program is quite alive and well even if their website is not [www.jsf.mil -- dc].

[...] At the moment, I'm working on a variant of the lift system that will allow engine interchangeability between Pratt & GE. The control operates the lift fan, roll posts and the really cool 3BSD nozzle on the back. I've got .mpg and .avi files showing the nozzle in operation and it just gets really amazing that we can swing that big, heavy thing around so quickly and nothing breaks. It \*is\* proof that Ada is capable of doing low-level, embedded, hard-realtime work for those who don't wish to believe it. (How's that for getting back on topic? :-)

*From: james hopper  
<hopperj@macconnect.com>  
Date: Thu, 13 Feb 2003 07:39:52 -0500  
Subject: Re: TCPIP package or binding?  
To: team-ada@acm.org*

[...] What I was doing was passing real time radar images from our sim code on the server to the display app on another computer. The difference was quite dramatic. I could get 60 hz with AdaSockets, but only about 10hz with Gnat sockets. [...]

*From: Bruce Hennessy  
<bhennessy@decilog.com>  
Date: Thu, 13 Feb 2003 15:08:44 -0500  
Subject: Re: UML to Ada Mapping  
Newsgroups: comp.lang.ada*

I am interested in participating in a definition of an Ada profile. I am currently working to present designs for mods to an existing large system written in Ada, which hasn't really got any designs (i.e. cohesive model diagrams and text that depict how the system hangs together). This can be put together with UML and non-standard drawings, but I would very much prefer extensions to UML that can then be standardized - in any case I would look to establish them across this project. [...]

*From: czgri@ocag.ch (Robert Grimm)  
Date: 14 Feb 2003 00:59:57 -0800  
Subject: Rational Apex 4.2 Migration:  
Sudden problems with Interrupts  
Newsgroups: comp.realtime,  
comp.os.vxworks, comp.lang.ada*

All RT software dealing with Interrupts and which were completely tested (positiv) developed by Apex 3.2.0 were migrated now to Apex 4.2.0 (Models). [...] Setup: Rational Apex Embedded 4.2.0 (Ada83, Ada95). Apex Model: power.vw\_ppc.board.ada83.4.2.0.rel, power.vw\_ppc.board.ada95.4.2.0.rel. [...] Target: Windriver Tornado 2.0.2. Target OS: VxWorks 5.4.2. Target HW: Dy4 SVME/DMV-181 und 750er (PowerPC CPU Boards, VMEBus) [...]

*From: tony\_gair@yahoo.co.uk (Tony Gair)  
Date: 14 Feb 2003 07:15:55 -0800  
Subject: Ada sockets and gnat 3.13 windows and linux interoperability  
Newsgroups: comp.lang.ada*

For the adasockets people - I have a client server application written on linux and gnat 3.13 using adasockets in which I send Ada types quite happily with no problems to and from a server to a client. [...]

*From: Frank J. Lhota  
<lhota.adarose@verizon.net>  
Date: Tue, 18 Feb 2003 16:41:28 GMT  
Subject: Ada interface to Winsock 2  
Newsgroups: comp.lang.ada*

I'm doing some Winsock work using ObjectAda. I would like to use some of the functionality added with Winsock 2.0. The Win32Ada interface, however, is based on the Winsock 1.1 API. I can write the a package with the Import pragmas for the additional Winsock 2 functionality, but in order to avoid "reinventing the wheel", I figure I should check my favorite NG to see if someone has done this already. [...]

*From: PlanetRecruit.com  
<mailout@planetrecruit.com>  
Date: Sat, 22 Feb 2003 17:56:43 GMT  
Subject: \*\* 2 NEW Jobs from  
PlanetRecruit.com (22/02/2003) \*\*  
To: Dirk.Craeynest@cs.kuleuven.ac.be*

[...] Ada C++ software engineers

Senior Software Engineer: Shall have five years work experience within the last seven years as a software engineer in software design and development of software programs in Ada, C++ and/or Java; shall have work experience on C3I software development programs; shall have experience with NATO and national defense software development practices and policies (e.g. MIL STD 490A, DoD STD 2167A, MIL STD 1521B); shall have a university degree in engineering...

Europe: Belgium - [...]

*From: John Doolittle  
<john.doolittle@lmco.com>  
Date: Mon, 24 Feb 2003 13:29:14 -0800  
Subject: Intel-OA: Win32 API application does not get Focus in Windows 2000  
To: intel-objectada@aonix.com <intel-objectada@aonix.com>*

We have recently upgraded to OA 7.2.2 with the latest patch. Our application uses

the Win32 API libraries for it's user interface controls. [...]

*From: Wiljan Derks*  
 <Wiljan.Derks@zonnet.nl>  
*Date: Wed, 26 Feb 2003 20:09:02 +0100*  
*Subject: Re: Windows services in GNAT Ada*  
*Newsgroups: comp.lang.ada*

Check out srvany. This tools was in the NT4 resource kit. It allows to run any program as a service. We started with this tool and it worked fine. Finally we needed more functionality and build our own version in Ada95.

*From: Stephen D. B. Wolthusen*  
 <wolt@igd.fhg.de>  
*Date: Fri, 28 Feb 2003 17:38:50 +0100*  
*Subject: Re: GNAT on Windows (Re: Ada using a DLL file*  
*To: team-ada@acm.org*

Is anyone aware of a restricted runtime environment (for Windows NT) [...] Background is that we have code that needs to run as kernel modules/kernel dlls/device drivers under Windows NT/2000/XP. Using pragma no\_run\_time is a bit harsh in its restrictions and leads to code that is less than perfectly readable (e.g. with regard to technically unconstrained arrays). [...]

Stephen Wolthusen, Fraunhofer-IGD, 64283 Darmstadt, Germany

*From: Marin David Condic*  
 <mcondic@acm.org>  
*Date: Fri, 28 Feb 2003 07:48:57 -0500*  
*Subject: Re: the Ada mandate, and why it collapsed and died*  
*Newsgroups: comp.lang.ada*

BTW: Just to follow-up on my own post and demonstrate that I practice what I preach... :-)) I do have Ada projects going on and am actively looking to start new ones if I can sell the ideas to my customers. We are on the lookout for qualified people so if someone is interested in a position in South Florida, you can always send me your resume at mcondic (at) belcan (dot) com [...]

*From: iatz <iatz@infonegocio.com>*  
*Date: Wed, 5 Mar 2003 16:59:12 +0100*  
*Subject: html parser*  
*Newsgroups: comp.lang.ada*

Does anyone know of any html scanner / parser written in Ada? I am currently programming a web browser from scratch as a part of a school project and need some help with the parser issue. [...]

*From: Arthur I Schwarz*  
 <Arthur\_I\_Schwarz@raytheon.com>  
*Date: Thu, 6 Mar 2003 11:50:30 -0800*  
*Subject: Ada Source Code Analyzer*  
*Newsgroups: comp.lang.ada*

We are evaluating toolsets which can provide Ada source code metrics, such as Cyclomatic Complexity, Halstead, etc., on several projects, each project having about 100k SLOC in 300 files. [...]

We need command line tools which can be scripted. We are using Cygwin under WinNT/2k. I am corresponding with some commercial vendors for evaluation copies of their software. Are there some non-commercial products available that I can evaluate?

*From: Wim van Deventer*  
 <wim.van.deventer@fcs-cs.com>  
*Date: Thu, 20 Mar 2003 14:13:23 +0100*  
*Subject: Intel-OA: 7.2.2: Warning message when debugging.*  
*To: intel-objectada@sf.aonix.com <intel-objectada@eonix.com>*

We are currently rewriting some parts our Ada83 software to Ada95 format. When we introduced protected units [...]

FCS Test Systems, The Netherlands, <http://www.fcs-cs.com>

*From: Jean-Philippe Vassilakis*  
 <jpvassilakis@jyhoriba.fr>  
*Date: Fri, 21 Mar 2003 11:29:52 +0100*  
*Subject: Intel-OA: From Task\_Identification to Windows Thread*  
*To: intel-objectada@eonix.com <intel-objectada@eonix.com>*

Using ObjectAda for Windows on W2k and XP, I would like to go down to the Windows Thread ID (and/or its attached MS-event).

Target: have an Ada monitoring task controlling the Windows threads life cycle of the corresponding Ada threads (especially when ending and the immediate time after their termination). We are having a permanent and strong use of those features on uninterrupted PC systems. Is there a way to achieve this?

[And from a later message: -- dc]

There is no idea to try to use Windows Threads at all, I am just speaking of real and good Ada tasks. I am just willing to monitor the Ada tasks by trying to have a view of their internal windows Thread ID and their probable attached Event for read-only observation.

Jobin-Yvon S.A.S. Thin Films Division, Software for Advanced Process Control & JY/Sofie in-situ in-line sensors, Chilly-Mazarin, France

*From: Scott Voyek*  
 <spvoyek@rockwellcollins.com>  
*Date: Wed, 26 Mar 2003 11:20:48 -0600*  
*Subject: Intel-OA: one or more multiply defined symbols found*  
*To: intel-objectada@eonix.com*

I have an Ada application that needs to link in some C code. I used Microsoft Visual C++ 6.0 to create several static libraries to be linked into my Ada application, [...] I am using Aonix 7.2.2 and have installed the latest (U8) patch. [...]

*Date: Sun, 30 Mar 2003 13:12:39 +0200*  
*Subject: Ada job announcement: Aubay Belgium*  
*To: ada-belgium@cs.kuleuven.ac.be*

Ada Software Engineer Real-Time Systems (Belgium)

Aubay is involved in very important IT projects on safety critical systems (embedded systems). We are therefore looking for a highly motivated Ada Software engineer "real time systems".

Function: Analysis, performance, development, testing and implementation of real-time embedded systems.

Profile: More than 5 years experience in Ada analysis and development. Good knowledge of electronics (microprocessors) is an asset. Knowledge of development methods (logistics engineering) is an asset. University level in Software Engineering. Languages: French. Teamwork spirit and good communication skills.

Technical skills: Development languages: Ada, C++. Design methodologies: Object Oriented Analysis and Design - UML; experience with CASE Tools. Good knowledge of Unix and NT platforms. [...]

*From: simon\_the\_softy@hotmail.com*  
 (Simon Apperley)

*Date: 31 Mar 2003 05:18:20 -0800*  
*Subject: Elegant 'abort' of sleeping task*  
*Newsgroups: comp.lang.ada*

I'm looking at the design of a piece of server code which has to handle calls that also pass a timeout value. The target system is aerospace related, and dynamically creating tasks 'on the fly' just is not an option. [...]

Simon Apperley @ General Dynamics UK COM

*URL:*  
<http://jobsearch.webbel.monster.be/>  
*Date: Thu, 3 Apr 2003 09:46:29 +0200*

BE-Henegouwen-Ada programmer (Urgent) [...]

Skills: Ada, Real Time, fluent French.

General function: Large railway company based in Charleroi is looking urgently for an Ada programmer with Real time experience. The candidate must be an expert with Ada and will speak fluent French. [...]

*URL:*  
<http://jobsearch.webbel.monster.be/>  
*Date: Sat, 5 Apr 2003 10:53:26 +0200*

BE-Bruxelles-Permanent: Calling all Ada Developers

We currently have several immediate opportunities for experienced Ada developers. You have at least 2 years of professional experience in Ada development as well as a good knowledge of the national languages. [...]

*From: Patrice Freydiere <frett27@free.fr>*  
*Date: Sat, 5 Apr 2003 16:24:24 +0200*  
*Organization: ENST, France*  
*Subject: AWS ..*  
*Newsgroups: fr.comp.lang.ada*

I am doing a project to build a picture server, with automatic generation of thumbnail pictures in Ada with AWS, [...]

*From: Frank Randolph CIV Beard  
<frank.beard@navy.mil>*

*Date: Fri, 25 Apr 2003 13:21:05 -0400  
Subject: RE: the Ada mandate, and why it collapsed and died  
Newsgroups: comp.lang.ada*

> [...] some of the most strikingly successful projects are considered strategic (whether in government or business) and are therefore kept secret. Once again, the feedback loops are blocked and learning does not take place on any significant scale.

I worked on the Space Shuttle ground based test chambers. The development was totally in Ada. The final product was about 200K lines of code. NASA was very satisfied with the software. We were even given an award by NASA. However, our NASA management told us not to tell anyway about the software, because, as a government agency, we would have to give a copy of the source code to anyone who asked. Among other things, they were afraid we would have to provide support to the outside world, even if it were only occasional telephone, or email, support. I never said anything anywhere that talked about it's success.

## Ada95 for Database-Oriented Business Applications

*From: Mark Wallace  
<Mark.Wallace@verizon.net>  
Date: Sun, 09 Feb 2003 05:58:31 GMT  
Subject: Ada95 in business (database) software  
Newsgroups: comp.lang.ada*

Is anyone using Ada95 for database-oriented business applications? [...] My team is evaluating Ada95 and Eiffel for a new development effort, and we are having a hard time identifying any sort of track record for Ada95 in this category (business/database) of application. All the Ada95 "buzz" seems to be related to device drivers and embedded controller software.

We're willing to swim upstream with regard to non-mainstream languages, because we care deeply about software quality. However, we have neither the expertise nor the funding to (re)create low-level code such as database interface modules.

*From: Poul-Erik Andreassen  
<poulerik@pea.dk>  
Date: Sun, 9 Feb 2003 12:40:07 +0100  
Subject: Re: Ada95 in business (database) software  
Newsgroups: comp.lang.ada*

I don't know if anyone are using this but you can have a look:

<http://www.redrocketconsortium.com/zbc/>. [See also "Zephyr Basecamp and Monkey Business Enterprise System" in the "Ada Products" section of this AUJ. --dc]

*From: Dmitriy Anisimkov  
<anisimkov@yahoo.com>  
Date: Mon, 10 Feb 2003 12:47:54 +0600  
Subject: Re: Ada95 in business (database) software  
Newsgroups: comp.lang.ada*

You could take a look to the Ada 95 to the (Oracle Call Interface). <http://vagul.tripod.com/adaoci.html>. It is also including to the GNADE. <http://gnade.sourceforge.net/> [See also "Oracle Call Interface (OCI) for Ada95" in AUJ 21-1 (Apr 2000), p.18, and "GNAT Ada 95 Database Development Environment" in AUJ 24-1 (Mar 2003), p.14. -- dc]

I'm intensively using it in the business application development. I know some other russian developers using it too. Frankly, there is no good documentation. But there is a good example showing the main functionality, and you can ask me directly.

*From: Peter Amey <peter.amey@praxis-cs.co.uk>  
Date: Mon, 10 Feb 2003 08:32:17 +0000  
Organization: Praxis Critical Systems  
Subject: Re: Ada95 in business (database) software  
Newsgroups: comp.lang.ada*

A significant chunk of a secure system we developed in SPARK and Ada 95 involved SQL links to a database. See "Correctness by Construction - Developing a Commercial Secure System" in IEEE Software Jan/Feb 2002 pp18-25. (Also in PDF on [www.sparkada.com](http://www.sparkada.com) under publications). ISTR we wrote the binding ourselves but that it wasn't hard.

*From: Dmitry A. Kazakov  
<mailbox@dmitry-kazakov.de>  
Date: Mon, 10 Feb 2003 10:37:46 +0100  
Subject: Re: Ada95 in business (database) software  
Newsgroups: comp.lang.ada*

I am using GNADE (its ODBC binding) with GNAT. GNADE is fine (but ODBC and its implementations is a real disaster). AFAIK, GNADE supports Oracle through native interface.

*From: David Botton <David@Botton.com>  
Date: Mon, 10 Feb 2003 19:16:18 -0500  
Subject: Re: Ada95 in business (database) software  
Newsgroups: comp.lang.ada*

I have used Oracle via ADO using GNATCOM working with huge (10s of millions of rows) databases on call center apps very successfully. I've also created a number of small apps using GWindows and its database components for rapidly creating database front ends.

## Ada in Fielded Military Systems

*From: mcq95@earthlink.net (Marc A. Criley)*

*Date: 23 Apr 2003 09:16:17 -0700  
Subject: Re: Ada in Iraq  
Newsgroups: comp.lang.ada*

[...] the best one can hope for is anecdotal bits and pieces. Here's some:

The B2 Weapons System Trainers and Mission Trainers (i.e., flight simulators) are programmed in Ada. The shipboard Tomahawk engagement planning system started transitioning to an upgraded version that was programmed in Ada in the late 90s, so it's a safe bet a very high number of Tomahawk launches were planned with the aid of that system.

Another source of info would be reviewing the "Military Applications" section of "Who's using Ada?" (<http://www.seas.gwu.edu/~mfeldman/ada-project-summary.html>) and determining which of those systems were likely employed in Iraq.

*From: burnsedbw@aol.com (BurnsedBW)  
Date: 24 Apr 2003 02:18:30 GMT  
Subject: Re: Ada in Iraq  
Newsgroups: comp.lang.ada*

At least the following systems, of which I have personal knowledge, are/were written (and fielded) in Ada: the ATACMS missile (part of MLRS family), the MLRS launcher, and (yes!) the PAC-3 missile.

*From: Jerry Petrey  
<jdpetrey@raytheon.com>  
Date: Thu, 24 Apr 2003 08:38:49 -0700  
Subject: Re: Ada in Iraq  
Newsgroups: comp.lang.ada*

It is also used on some of our missiles, the JSOW and Paveway, in particular, as well as on the Apache and Comanche helicopters, Longbow missile, F-16 and F-18 fighters, B1 bomber, C17, and C130J transports to name a few more applications.

Jerry Petrey, Senior Principal Systems Engineer, Raytheon Missile Systems

*From: Richard Riehle  
<richard@adaworks.com>  
Date: Thu, 24 Apr 2003 17:18:54 -0700  
Organization: AdaWorks Software Engineering  
Subject: Re: Ada in Iraq  
Newsgroups: comp.lang.ada*

Hellfire missile. Much of the Bradley. Other armored vehicles. Various software systems for some of the bombers. Other stuff that cannot be mentioned.

*From: John R. Strohm  
<strohm@airmail.net>  
Date: Fri, 25 Apr 2003 00:37:20 -0500  
Subject: Re: Ada in Iraq  
Newsgroups: comp.lang.ada*

TOW ITAS (Improved Target Acquisition System: a sight, autotracker, and guidance computer for the TOW missile) was written in Ada. I worked on it. Development was 1994-1995 at Texas Instruments. I'd be very surprised if it wasn't fielded by now.

---

## Ada in Context

### Feb 17 - It was 20 years ago today...

*From: dirk@cs.kuleuven.ac.be (Dirk Craeynest)*  
*Date: 16 Feb 2003 21:47:52 +0100*  
*Subject: Feb 17 - It was 20 years ago today...*  
*Newsgroups: comp.lang.ada*

Monday February 17, 2003, marks the 20th anniversary of Ada as a standardized language. On this day in 1983 the first Ada standard ANSI/MIL-STD-1815A was published.

We've come a long way since, and it hasn't always been easy. But there's a lot of momentum in the international Ada community now.

While there are signs that global awareness of the importance of reliable software is increasing, at the same time enthusiasm within the Ada world is growing and new activities are springing up to make Ada an even better language and even more suitable for the task of cost effectively building and evolving demanding applications in an ever changing world. I refer among others to the work done world-wide in ISO's WG9 and ARG on the evolution of the Ada standard, the involvement of the Ada-Europe and ACM SIGAda organizations to set up a light-weight infrastructure for managing upcoming "de facto" standards for APIs and libraries (ala C++'s STL), the increased interest in Ada-Europe's and SIGAda's annual conferences (new people and companies attending and exhibiting, more and good quality of submissions, etc.), increasing "on line" activities in newsgroups (comp.lang.ada), mailing lists, and on all kinds of web-sites with e.g. open source projects, and so on...

I would like to invite you all to help that momentum grow and to share a bit of your Ada enthusiasm with friends and colleagues. Contact or form local user groups, write and/or talk about your professional or hobby projects, speak with professors at educational institutes in your neighborhood, etc.

In short: show the world that Ada is very much alive and has a lot to offer!

Dirk Craeynest,

Dirk.Craeynest@cs.kuleuven.ac.be (for Ada-Belgium/Europe e-mail)

[And from a later message: -- dc]

Jeffrey Carter <jrcarter@acm.org> writes:  
 > I don't wish to belittle the importance of the anniversary, but the first Ada standard was MIL-STD 1815, 1980 Dec 10 (Ada was born 1815 Dec 10).

You are correct that the first "official" Ada reference manual was published on December 10, 1980.

My copy of the Ada80 reference manual was printed by Springer-Verlag in Lecture Notes in Computer Science as issue number 106, with copyright 1980 by the United States Government, [...], is entitled "The Programming Language Ada - Reference Manual", with subtitle "Proposed Standard Document - United States Department of Defense".

Note the "Proposed Standard Document" in the subtitle.

This Ada80 language definition was "only" a US military standard, was AFAIK never fully implemented, and has been non-trivially updated after careful international review to finally result in the Ada83 standard mentioned in my previous message. So, exactly 20 years ago today, Ada83 became both a military (MIL-STD) and a non-military (ANSI) standard, which was as we all know later adopted unchanged by ISO as an international standard.

### Happy 25th Anniversary to Blue, Green, Red and Yellow

*From: Ben Brosgol <brosgol@gnat.com>*  
*Date: Mon, 17 Feb 2003 10:56:45 -0500*  
*Subject: Happy 25th Anniversary to Blue, Green, Red and Yellow*  
*To: team-ada@acm.org*

And for the real nostalgia enthusiasts, this past Saturday (Feb 15) marked the 25th anniversary of the delivery of the four competing design specs for (as it was then known) "DoD1": the Blue, Green, Red, and Yellow languages. This was shortly after the "Blizzard of '78" in Boston, so John Goodenough (then at SofTech, and leader of the Blue team) and I (corresponding position at Intermetrics, for the Red language) had an interesting time on the "finalization" aspects of the design and delivery of the documents.

### Advantages of an Ada RTE

*From: John R. Stroh*  
*<strohm@airmail.net>*  
*Date: Tue, 14 Jan 2003 22:03:23 -0600*  
*Subject: Re: Ada Compiler Pricing*  
*Newsgroups: comp.lang.ada*

> Commercial RTE's like VxWorks and LynxOS have been ported to most platforms and are much more popular than any Ada RTE. [...] In the real-time shops I've worked with, proposing Ada for a new project would be considered rather radical.

Back when I was doing Ada, on whatever platform, the runtime was part of the total Ada toolset package. It came already integrated with the rest of the toolkit. The customer normally had to write a board support package to integrate the runtime with the peculiar target hardware, but you have to do that with EVERY executive (unless your hardware is COMPLETELY off-the-shelf - very rare in military embedded systems - and the vendor already supplied you a HIGH-QUALITY BSP to go with your boards - also very rare in embedded systems).

And, for the record, a competently-written Ada runtime fits in about 4K. (4K bytes on a byte machine, 4K words on a word machine, 4K whatever on a whatever machine.) (Ichbiah made that statement in an Ada training video I saw back in about 1990 or 1991. I checked it for the TI 320C30 Ada toolset from Tartan Labs, and it was dead on target.)

I don't know how much VxWorks or LynxOS takes. Anyone have the numbers handy?

*From: Richard Riehle*  
*<richard@adaworks.com>*  
*Date: Wed, 15 Jan 2003 09:43:09 -0800*  
*Organization: AdaWorks Software Engineering*  
*Subject: Re: Ada Compiler Pricing*  
*Newsgroups: comp.lang.ada*

I remember the Ada project at Xerox where the team built the embedded system for a copier using Ada, and did not need to buy a separate RTOS. They used the one supplied by the Ada compiler publisher. It was a highly successful project and the team, along with its manager, realized that Ada was a significant improvement over the way they had built such systems before.

Xerox executives, with their customary skill at "grabbing defeat from the jaws of victory" (Smalltalk, Windows Operating systems, etc.), decreed that Ada would not be allowed to be used for anymore systems. This decree was made even in the face of compelling success with it.

The project manager, in describing the project, remarked on the added value that Ada provided in eliminating the need for purchasing stuff such as VxWorks and related products. Everything they needed was in the Ada.

I see this kind of dumb management a lot in DoD contractor companies. Somehow, these corporate level people are so enamoured of the slick and glossy ads they fail to recognize quality when they see it. Xerox has been particularly guilty of this kind of glitz thinking when it comes to computing. They could own the marketplace now, if there had been a well-functioning brain in the executive levels of the company during the Seventies and Eighties.



## Ada Compilers Do Much of the Work for You

*From: Warren W. Gay VE3WWG*

*<ve3wwg@cogeco.ca>*

*Date: Thu, 06 Feb 2003 13:14:52 -0500*

*Subject: Re: Bye-bye Ada ? (Ada95*

*Wholesale Changes?)*

*Newsgroups: comp.lang.ada*

> I think that any of the regular posters here would be able to program as well in C++ as in Ada. I'm pretty sure that if I had to work in Ada it wouldn't take me very long to get up to expert level.

Well, I can say that for small projects, your statement is probably true.

But I can speak for my own experience, that in largish C++ projects, I always spent an incredible amount of time looking in core files and working with a debugger to find out why some "strange behaviour" was happening in those C++ projects. Sometimes it was related to those automatic type promotions, and at other times, it was related to memory corruption problems of one sort or another (and it need not even be your own code! Sometimes C++ library code does this to you).

On top of all this, once you get your large C++ application to actually work, you then want to "port it" to another platform, with sometimes a different compiler (or even a different/updated gcc compiler). What do you find? Changed behavior in automatic type promotion, leading to changed application behavior; #include file problems; C++ library incompatibilities/limitations.

Sure you can say that this occurs due to poor planning, or design. But the Ada point is that this is not checked, nor enforced up front by your tools. My hobby time is time I hate wasting this way ;-)

In smallish projects, Ada sometimes takes me longer. This is usually due to library issues and bindings.

The payoff is huge though (for me) in largish projects. In the last 4 years (I think) of using Ada (since my conversion ;), I've only encountered memory corruption once (I had freed an object twice due a logic/design error). The rest of my time is "quality time with the compiler", learning how to avoid certain design errors, and learning how to better engineer modules so that the dependencies are pure and correct (C++'s elaboration is pale by comparison). My debugging time is 25% or less of the C++ debugging time. (This gives me a much more pleasant use of my hobby time). When I debug, it is because I am finding out why `_I_` made some logic error. It is not due to corruption, or some faulty elaboration order; not due to external memory references that the linker decided were the same (name collision), not due to #include

file sequence, not due to automatic type promotion surprises etc.

SUMMARY: I don't agree with your statement for large projects. [...]

Ada programmers may wince at this, but here is something that I find really useful about Ada95 for my own Open Source development:

If I look at a spec of a package I want to "clean up", and I don't like the way the data type is presented, and want to make some sort of wholesale change to the package, I'll first make the necessary tweaks or major changes to the specs. This may include: changing constants to enumerated types / vice versa; changing a weak type to a strong type; making types more range specified (subtypes); changing function/procedure API changes; change a record/tagged type (class) to the other kind; etc.

Then I jump into emacs, and start invoking "make" and let the compiler tell me about all of those new errors that will now spew of the compiler. Using emacs I jump to the source of the error (using emacs macros) and make all the necessary changes, until the errors go away. With Ada95 this has never failed me, though I expect some little case may bite me some day, doing changes like this.

THIS HAS NEVER WORKED WELL FOR C++. C++ would never identify all of the necessary changes, because of the number of "automatic conversions" and so on. C++ never tells you that you have a missing case in a switch statement, and its very forgiving mixing enumerated types with ints etc. C++ will not tell you when a related constant or #define needs to be updated.

This Ada95 feature has made it possible for me to very quickly produce clean library interfaces, because I am not reluctant to make wholesale changes when I think it might be beneficial to the client. [...]

I'd be interested if anyone actually uses this type of procedure in more "critical" application development roles. I am sure that others must take advantage of this, if not secretly so? ;-)

Is this type of thing frowned upon by DOD projects, or do they even know this type of thing happens? Just curious.

Warren W. Gay VE3WWG,  
<http://home.cogeco.ca/~ve3wwg>

*From: Robert Spooner <rls19@psu.edu>*

*Date: Thu, 06 Feb 2003 13:51:36 -0500*

*Organization: Penn State University, Center for Academic Computing*

*Subject: Re: Bye-bye Ada ? (Ada95*

*Wholesale Changes?)*

*Newsgroups: comp.lang.ada*

I have done this. It's particularly helpful when I haven't gotten the design right the first time because of a lack of information early-on. I can't imagine doing it in any

other language. Ada allows you to use the computer for things that computers are good for and humans are not - all that consistency checking. When I first was learning Ada I was astonished at the percentage of my programs that did exactly what I wanted once I had a clean compile. My debugger skills atrophied.

*From: Frank Randolph CIV Beard*

*<frank.beard@navy.mil>*

*Date: Thu, 6 Feb 2003 14:12:48 -0500*

*Subject: RE: Bye-bye Ada ? (Ada95*

*Wholesale Changes?)*

*Newsgroups: comp.lang.ada*

I don't know how "critical" you mean, but I've used this feature of Ada on a number of occasions which stemmed multiple projects from running the on ground test chambers for the Space Shuttle group, to the onboard Space Station data storage and retrieval component, to a Naval communications project.

Even though our tools usually give us the capability to see a callers graph, it is still more useful to prototype changes in the spec and have the compiler reliably show me the immediate impacts of the changes. And once the changes made it to fruition, they required very little debugging.

I don't think I've ever openly shared this information even with fellow Ada developers, but it sure does make things easier.

*From: David Botton <David@Botton.com>*

*Date: Thu, 6 Feb 2003 17:16:43 -0500*

*Subject: Re: Bye-bye Ada ? (Ada95*

*Wholesale Changes?)*

*Newsgroups: comp.lang.ada*

Doesn't every one do this?

Richard Riehle once wrote: "But the compiler will catch any potential ambiguities. This is the key point. Ada is designed to detect inconsistencies, irregularities, discontinuities, incongruencies, and unconfirmable constructs earlier in the software development process than any other language."

Also, see his article on Ada Power at: <http://www.adapower.com/articles/capsule.html>

This is THE reason that Ada should be THE choice for almost every project. In particular for the free time hack!

*From: Jerry Petrey*

*<jdpetrey@raytheon.com>*

*Date: Thu, 06 Feb 2003 16:00:48 -0700*

*Subject: Re: Bye-bye Ada ? (Ada95*

*Wholesale Changes?)*

*Newsgroups: comp.lang.ada*

You mean there is another way? :-)

I have always used this method with Ada development.

That is one of the things I love about Ada - you can let the compiler do so much of the work for you. I have often had people new to Ada come to me asking "can I do

this or that in Ada?" - I usually tell them to try it and see - let the compiler tell you if it won't work; you will learn something from the exercise in either case. It is hard to imagine using a language and compiler without this level of checking again.

Jerry Petrey, Senior Principal Systems Engineer, Raytheon Missile Systems

From: Dmitry A. Kazakov

<mailbox@dmitry-kazakov.de>

Date: Fri, 07 Feb 2003 10:22:04 +0100

Subject: Re: Bye-bye Ada ? (Ada95 Wholesale Changes?)

Newsgroups: comp.lang.ada

I am using this technique for both C++ and Ada, mostly because the strategy of our firm and our customers is: "very quick and incredibly dirty". (-:)

It works perfectly in Ada, but has serious problems in C++, especially with MS VC++, which is unable to recompile as necessary, even if precompiled headers are not used (this feature does not work anyway). It is disastrous because a recompile-all lasts for 4-hours to see how thousands compile errors appear on the screen. [...]

Dmitry Kazakov, www.dmitry-kazakov.de

From: Warren W. Gay VE3WWG

<ve3wwg@cogeco.ca>

Date: Mon, 10 Feb 2003 12:16:17 -0500

Subject: Re: Bye-bye Ada ? (Ada95 Wholesale Changes?)

Newsgroups: comp.lang.ada

I appreciate the comments you and others have made to this informal "curiosity poll". This confirms the general usefulness of the Ada compiler in this role. I think this goes a long way to creating good "APIs", since the programmer is much more inclined to make sweeping changes if it is seen as beneficial. [...]

## Programming Language Choice Matters

From: brougham3@yahoo.com

Date: Wed, 26 Feb 2003 20:47:04 -0600

Subject: Re: Ada versus language-X and "getting real work done"

Newsgroups:

comp.lang.java.advocacy,comp.object,comp.lang.ada

> Programming language choice is so far from a factor in interoperability and communications that your comments are laughable.

Why are stack buffer overflow problems such an issue in C when programs receive communications that differ from what the programmer expected? Yes, I'm aware this is a programmer deficiency. Subscribe to something like Bugtraq for any length of time, and you'll see the same programmer problems over and over and over again.

Programming language choice is a factor. If you have a skilled programmer,

language doesn't matter. The problem is that it's far harder to find a truly skilled C++ master than it is a C one. Both are many times harder to find than a skilled Java guru. Not sure about Ada.

From: Steve <steved94@attbi.com>

Date: Thu, 27 Feb 2003 03:35:09 GMT

Subject: Re: Ada versus language-X and "getting real work done"

Newsgroups:

comp.lang.java.advocacy,comp.object,comp.lang.ada

[..], there is a significant evidence the choice of language does in fact matter. A significant amount of work has been done to determine factors that effect the cost of developing software, defect rate, etc.

This information is usually ignored. [...]

From: xanthian@well.com (Kent Paul Dolan)

Date: 26 Feb 2003 23:09:01 -0800

Subject: Re: Ada versus language-X and "getting real work done"

Newsgroups:

comp.lang.java.advocacy,comp.object,comp.lang.ada,misc.misc

> Programming language choice is so far from a factor [...]

Interesting that your experience should be so in contrast to the ten years of research and effort DoD put into studying that very issue in their own shops using their own code and repeated system interoperation experiences to the contrary, in the work leading up to the first Ada language standard. I'm sure your correction to history will be of great value to those who participated in it. [...]

From: Dr Chaos <mbkennel@yahoo.com>

Date: Thu, 27 Feb 2003 22:06:20 +0000

Organization: Univ of Calif San Diego

Subject: Re: Ada versus language-X and "getting real work done"

Newsgroups: comp.lang.java.advocacy,comp.object,comp.lang.ada

Followup-To: comp.lang.java.advocacy

> Programming language choice is a factor. If you have a skilled programmer, language doesn't matter.

This is an assertion that is frequently made, but rather questionable without empirical evidence one way or another. With persistence and skill, programmers can overcome many deficiencies, but deficiencies they still are.

> The problem is that it's far harder to find a truly skilled C++ master than it is a C one.

Why is that? Perhaps there is something intrinsic about the language design there because it's not a lack of exposure.

> Both are many times harder to find than a skilled Java guru. Not sure about Ada.

Why is guru-ness so correlated with language now? My personal opinion is that good languages ought to, and do,

bring up the average quality of the "30% to 98%" skill levels of programming.

My personal opinion is that C++ induces too much "risk" both in programs, where working programs and buggy programs are almost interlocking dense sets (in the mathematical sense) and among programmers. That is there is some non-zero chance that you might end up with a programmer, perhaps good at lots of other things, who is just plain unable to be reliably successful with C++. I think that would be much less likely with other programming languages.

From: JXStern

<JXSternChangeX2R@gte.net>

Date: Thu, 27 Feb 2003 23:49:26 GMT

Subject: Re: Ada versus language-X and "getting real work done"

Newsgroups: comp.object,comp.lang.ada,comp.lang.java.advocacy

[On why it is harder to find a skilled C++ master rather than a mediocre one. -- dc]

Because the language is a mess, and there is no objective measure of who is a "master". Is it someone who can make instant sense out of the worst mess anyone hands him? Then it ain't me. Is it someone who makes rather spare use out of obscure constructs? That would be me, but I would be down-rated because of it in many evaluation contexts. That I do it on purpose, is generally not appreciated. That I got here via the other path, is not understood. [...]

From: John R. Strohm

<strohm@airmail.net>

Date: Fri, 7 Mar 2003 12:59:44 -0600

Subject: Re: Ada versus language-X and "getting real work done"

Newsgroups: comp.lang.java.advocacy,comp.object,comp.lang.ada

What you miss in that statement is that it is nowhere near as necessary to HAVE guru-level programmers when working in Ada. The compiler catches many of the problems that in C or C++ would normally require the guru to meditate long and hard, and perhaps consult the entrails of a freshly-killed junior programmer.

From: Karen

<karen.nussbaum@blueyonder.co.uk>

Date: Fri, 28 Feb 2003 00:13:29 GMT

Subject: Re: Ada versus language-X and "getting real work done"

Newsgroups: comp.lang.java.advocacy,comp.object,comp.lang.ada

[...] Ada is without doubt a solid language. Its the only language I have found where you can teach it through the structure of teaching software engineering. (Teach s/w eng. goals -> principles -> language constructs, so the students know why to do something, not simply the syntax).

The real benefit of other languages is the leverage it can give you to interface / exploit COTS products that provide services in only another language. For

ages the only way to integrate a commercial GUI into a system, was to go via C and later C++. We had to wait so long for native Ada GUI services it was laughable. Only with Ada 95 was the "Ada - I'm the centre of the world and if you can't play Ada, I won't play with you" view changed, and decent mixed language systems could be built - opening the way to exploit commercial tools but still use a decent language for the "business logic (e.g. defence) s/w.

My organisation routinely rolls out multimillion lines systems (soft real-time) for the defence industry - and it works. Increasingly we exploit commercial s/w products, embedded into our systems, but the core is Ada, and we have no intention of changing that.

## Feedback to BBC's Item on Responsible Coding

*From: Carlisle Martin C Dr USAFA/DFCS  
<Martin.Carlisle@usafa.af.mil>  
Date: Wed, 2 Apr 2003 11:03:17 -0700  
Subject: Re: BBC on "responsible coding"  
To: team-ada@acm.org*

[In response to an article entitled "Call for 'responsible' coding" at <http://news.bbc.co.uk/2/hi/technology/2895177.stm> -- dc]

Have submitted the following:

While I agree with Ken Nelson that Java represents a significant improvement over C in this regard, you certainly won't find any Boeing 747's running Java. Note the following quotation from:

<http://java.sun.com/products/jdk/1.2/LICE/NSE>

"4. High Risk Activities. Notwithstanding Section 2, with respect to high risk activities, the following language shall apply: the Software is not designed or intended for use in on-line control of aircraft, air traffic, aircraft navigation or aircraft communications; or in the design, construction, operation or maintenance of any nuclear facility. Sun disclaims any express or implied warranty of fitness for such uses."

In such cases, Boeing and others who must meet FAA certification have shown a clear preference for Ada, whose strong typing facilities promote secure coding. In fact, the SPARK subset of Ada allows programmer to actually have automated proofs that software doesn't contain major categories of errors, well beyond those mentioned in this article.

Martin C. Carlisle, Associate Professor and Advisor-in-Charge, Department of Computer Science, United States Air Force Academy

*From: Ben Brosgol <brosgol@gnat.com>  
Date: Fri, 4 Apr 2003 13:48:17 -0500  
Subject: Re: BBC on "responsible coding"  
To: team-ada@acm.org*

> That restriction was in the Java license because Java is garbage-collected and the garbage collector can run at arbitrary times and take however long it needs. [...]

GC was one (but not the only) reason for the disclaimer. "Java technology" includes the class libraries, and that stuff was definitely not written to comply with any software safety standards.

## On Real-Time Java and Garbage Collection

*From: xanthian@well.com (Kent Paul Dolan)  
Date: 10 Apr 2003 23:10:00 -0700  
Subject: Re: real-time Java  
Newsgroups: comp.lang.java.advocacy,comp.lang.ada*

> I've been hearing a lot lately about real-time Java. I must confess I'm baffled about the motivation behind this endeavor. Would someone please tell me what real-time Java will be able to do that Ada can't do? Or what it will be able to do better than Ada? Thanks.

Sure: bring a popular, thriving, employer-demanded language with a wealth of libraries and a robust built-in GUI into the real time arena.

How the heck you do that with a language subject to garbage collection is going to be an interesting thing to watch happen. Some complaints in [comp.lang.java.programmer](http://comp.lang.java.programmer) mention Java going off to stew for whole minutes at a time.

*From: wojtek@power.com.pl (Wojtek Narczynski)  
Date: 15 Apr 2003 10:59:10 -0700  
Subject: Re: real-time Java  
Newsgroups: comp.lang.java.advocacy,comp.lang.ada*

[...], garbage collector can be deterministic thus realtime. But in practice my experience with Java libraries is that the developers relieved of the necessity to free memory explicitly, tend to forget about memory management completely (i.e. not clear stale references). You're stuck with `OutOfMemoryError` even though there is no "garbage" per se in the running app, and you really cannot do much. You cannot contact the vendor and say - "your library leaks". [...]

*From: Wesley Groleau  
<wesgroleau@despammed.com>  
Date: Tue, 15 Apr 2003 13:47:02 -0500  
Subject: Re: real-time Java  
Newsgroups: comp.lang.java.advocacy,comp.lang.ada*

I was amused at JBuilder one day. I pulled down an "info" menu option. At the bottom of the little widget that popped up was the amount of memory in use. Even though I was doing absolutely nothing, the number kept slowly rising. When it hit

some threshold, GC was triggered and it started over.

*From: Tor Iver Wilhelmsen  
<tor.iver.wilhelmsen@broadpark.no>  
Date: 16 Apr 2003 07:16:54 +0200  
Subject: Re: real-time Java  
Newsgroups: comp.lang.java.advocacy,comp.lang.ada*

Welcome to the wonderful world of non-immutable events leading to tons of events created and discarded in any given Swing application as it runs. Even if you did absolutely nothing, Swing (and possibly JBuilder's "virtual file system") was generating events like asking itself if it needed to repaint etc.

*From: Wesley Groleau  
<wesgroleau@despammed.com>  
Date: Tue, 15 Apr 2003 13:43:29 -0500  
Subject: Re: real-time Java  
Newsgroups: comp.lang.java.advocacy,comp.lang.ada*

[About the JVM as a shared library or resident in the OS, instead of having every Java program in a multi-user environment running its own copy of the JVM. -- dc]

> Unix and Windows will load only one copy of a binary no matter if it's an executable or shared library. So there is only one copy of the garbage collector code. I don't see how collector data could be shared though.

I can't speak for Windows, but this is not true of Solaris. The project I recently left had a problem reporting/tracking tool in Java. Everybody used it. That meant that at any given time, the physical memory and swap space had twenty to thirty copies of the JVM running, and each copy had additional memory allocated to all the same classes.

*From: xanthian@well.com (Kent Paul Dolan)  
Date: 15 Apr 2003 18:07:20 -0700  
Subject: Re: real-time Java  
Newsgroups: comp.lang.java.advocacy,comp.lang.ada*

Hmm, that does raise an interesting point, though. Even if the JVM code space were shared, the classes are `JVM_data_`, not underlying `OS_code_`, and so might not be sharable, unless the clean page/dirty page rules [...] hold for `_any_disk` page images (which might well be the case). Even then, there is next to no way the JIT native code could be shared, but perhaps it is a relatively small item compared to the classes from which it is compiled, since it is only the hot loops of those classes.

## Java vs Ada: the Good, the Bad and the Ugly

*From: Wesley Groleau  
<wesgroleau@despammed.com>  
Date: Thu, 24 Apr 2003 09:03:53 -0500  
Subject: Re: Ada in Iraq  
Newsgroups: comp.lang.ada*

I am just finishing up a college course in Java. Reflecting back on the good, the bad, and the ugly.

Good: All the times (not as many as I was led to expect) where I saved time by calling some existing standard library instead of writing my own.

Bad: All the times (not as many as I was led to expect) where I wasted time repeatedly writing low-level implementations of things that Ada has built in.

Ugly: Line after line of defined integer constants to get the readability benefits of enumerated types with not even a hint of their type safety. Most "roll-your-own" imitations of high-level language features are seriously less readable than built-in versions. (A.multiply(A)).add (B.multiply(B)) instead of  $A * A + B * B$

## On Quality of Tools and Management

From: Richard Riehle

<richard@adaworks.com>

Date: Thu, 20 Feb 2003 07:49:58 -0800

Organization: AdaWorks Software Engineering

Subject: Re: status of PL/I as a viable language

Newsgroups: comp.lang.pl1,comp.lang.ada

> [...] one of the lead stories today on the Aviation Week web site

www.aviationnow.com describes how the F-22 program is now some \$800 million overrun because of delays caused by unreliable software. They can get it to run for about 8 hrs in the lab before it crashes, but in the actual test flights it fails within 3.5 hours forcing the pilot to reboot the system while in mid-flight. If they are using Ada as you contend, then this is a terrible indictment and confirms what I had heard earlier about the language's shortcomings.

The F-22 is one of the most complex systems, software and hardware, even conceived. As we discussed earlier, with regard to other language environments, the quality of the available tools can help only so much.

Ada is designed to maximize the amount of error checking possible as early in the development process as possible. I know of no other language does this as well as Ada. As stated earlier, competent people have used Ada for a wide variety of successful large-scale, safety-critical software systems. On the other hand, less competent people have used Ada, PL/I, C++, C, etc. for a wide variety of unsuccessful systems. At least one Ada failure that I know of by reports from participants rather than from first-hand knowledge was an Air Traffic Control system. The people who were managing the system blamed Ada when they, the

managers, were the real problem. At that time, other ATC systems had already been completed in Ada and were successfully doing their job.

The problem is not the language. As someone else once said, "A fool with a tool is still a fool." From my vantage point, having a fairly broad experience with a lot of programming languages, Ada, once it is understood, continues to be the most appropriate choice for systems such as the F-22. The fact that developers can find a way to screw it up does not detract from the value of the language. If they can make a mess using a language with the rigorous controls built into Ada, imagine the magnitude of the mess they could make with, say, C++. And, no, PL/I would not help with a system this large and complex.

From: Jerry Petrey

<jdpetrey@raytheon.com>

Date: Fri, 21 Feb 2003 15:37:17 -0700

Organization: Raytheon Company

Subject: Re: Donald's F-22 Question

Newsgroups: comp.lang.pl1,comp.lang.ada

I worked for Lockheed during part of the F22 development period although not directly on the F22 program but on another aircraft system (in Ada also). The problem is a common one on large programs – mostly one of management. Managers assume all engineers are alike. There were a few good Ada programmers and a lot of average or below average AdaTran and AdaC programmers ("I can write Fortran (or C) in any language" people). The overall code that resulted, while Ada, was certainly not good Ada. It was a mess to get working and resulted in huge cost overruns, but I would guess it would have been much worse if they had used some other language.

Certainly using Ada does not solve all the problems but it does make the job a little easier of having a wide variety of skill levels involved on a large project. With some good management (rarer than hen's teeth) and letting the minority of top notch software engineers manage and enforce a good process, great results can be achieved even with a lot of lessor skilled people on the team. This may be true in other languages as well, but more so in Ada, and every advantage helps in a large project like this.

Jerry Petrey, Senior Principal Systems Engineer, Raytheon Missile Systems

From: Richard Riehle

<richard@adaworks.com>

Date: Fri, 21 Feb 2003 12:41:23 -0800

Organization: AdaWorks Software Engineering

Subject: Re: status of PL/I as a viable language

Newsgroups: comp.lang.pl1,comp.lang.ada

[...] We need to understand that there are limits to what can be accomplished by depending only on our tools. Ada has a lot

to recommend it for high-integrity software. Other languages have a lot to recommend them. This started as a PL/I thread, and I have learned, more recently, of improvements in PL/I since I last used it. Even so, PL/I, for all its improvements, requires skilled programmers. Ada requires skilled programmers. C++ requires skilled programmers. [...]

Ada is like using a torque wrench. C++ is like using any convenient long-handled wrench. If the mechanic is careful and has a lot of experience, using that long-handled wrench, it will work just fine. In most cases, though, we might find the torque wrench a little more reliable. However, if we have no clue about the appropriate level of torque, cannot beforehand do the required computations, and have no idea what torque is, we are going to twist off the head of the bolt just as easily as the guy with the simple long-handled wrench.

From: Preben Randhol <randhol@pvv.org>

Date: Thu, 20 Feb 2003 17:15:10 +0000

Organization: Norwegian university of science and technology

Subject: Re: status of PL/I as a viable language

Newsgroups: comp.lang.ada

> You're right in that good (really good) programmers can produce successful code not matter what tools (languages) they have to tolerate while poor programmers (or at least poorly trained) will screw it up no matter how good or bulletproof the language. Your ATC example undoubtedly proves the point.

Yes, but if you are a good programmer you would want to use the most appropriate tool. Especially when time is not something you have a lot of in a project.

From: Vinzent Hoefler

<ada.rocks@jlfencey.com>

Date: Thu, 20 Feb 2003 16:20:55 -0500

Subject: Re: status of PL/I as a viable language

Newsgroups: comp.lang.ada

> Which is to say, complex systems can have complex failure modes,

Right.

> and the fact that some programming languages will catch simple errors isn't going to help at all with the complex ones.

NAK. One thing is for sure: The ability to get rid of all the small, stupid, simple errors - and even those have very bad consequences a lot of times - helps to keep concentrated on avoiding the complex ones.

## Building Quality In vs Testing Quality

From: Marin David Condic

<mcondic@acm.org>

Date: Tue, 25 Feb 2003 08:10:23 -0500  
 Subject: Re: Quality (Re: status of PL/I as a viable language)

Newsgroups: comp.lang.pl1,comp.lang.ada

> When you talk about quality, I think a key issue is a discussion about building quality versus testing quality. What means "building quality into something"?

You can never escape the necessity of testing. Whatever you build will ultimately be tested so long as it is actually used. Either you test it or your customer tests it. But that said, testing can't build in quality. It can only demonstrate quality - or the lack thereof. Ideally, you would build a product, test it and find zero errors or defects. That would reflect building the quality in. Going through a cycle of testing it, fixing it, testing it again, fixing it again, ..., is an attempt to test the quality into a product. Using that approach, you might eventually get to zero defects, but it takes longer and costs more than if the test immediately revealed zero defects.

Ada is not the end-all, be-all of software development. It is absolutely not going to guarantee that you get to zero defects. However, one of the things I really like about Ada is that it tends to catch lots of simpler errors up front and never lets them get into the finished product. (Things like type checking, parameter/interface checking, bounds checking, etc.) On my project I know from metrics that this saves me cost & schedule compared against the use of some other languages. (Fortran, C and Assembler, mostly) To the extent that a language eliminates errors up-front, I think that goes to "Building Quality In" and the odd thing is that it costs less to build a better product.

Marin David Condic. My project is:  
<http://www.jsf.mil/>

## To Dream Tomorrow: A Portrait of Ada Lovelace

From: John McCormick  
 <mccormic@cs.uni.edu>

Date: Fri, 25 Apr 2003 15:24:43 -0500  
 Subject: Fwd: New Documentary on Ada Lovelace

To: team-ada@listserv.acm.org

[John W. McCormick, Computer Science Department, University of Northern Iowa, forwarded Deepak Kumar's message that was posted to the ACM SIGCSE list. See also "Jan 21 - U.S. Premiere of New Documentary Film" in AUJ 24-1 (Mar 2003), p.6. A screening of this documentary plus Q&A session with its director John Fuegi has been included in the social events program of the Ada-Europe 2003 conference as well. -- dc]

On behalf of Flare Films  
 (www.flarefilms.org) I would like to announce the completion of a fresh new

documentary on Ada Lovelace. It is called "To Dream Tomorrow: A Portrait of Ada Lovelace" Produced and directed by John Fuegi and Jo Francis. Color, 53 mins (2003).

We did a screening of this documentary two weeks ago as a part of our "Computing History Month" (see [www.cs.brynmawr.edu](http://www.cs.brynmawr.edu))

John and Jo came to do the screening, and led a Q&A session afterwards which was absolutely wonderful, not to mention eye opening. The documentary is well researched, beautifully filmed, and begins to debunk several myths floating around about Ada Lovelace among even the most notable historians.

I think this is an important artifact in the history of computing, especially pertaining to the role and engagement of women in Math & Computer Science. Personally, I think every computer science program should own a copy and do a regular screening. The rights to public screening are very very cheap. Please contact Flare Films at [www.flarefilms.org](http://www.flarefilms.org)

The movie has only been screened at a handful of places so far, it will be formally premiered in Washington DC this summer. Some comments about the film are attached below.

"An engaging, beautiful, well researched film that tells the story of the remarkable woman who fought against the customs of her day to participate in the dawn of the computer age." -- Leonard J. Shustek, Chair, Computer History Museum, Silicon Valley

"'To Dream Tomorrow' is a moving and intelligent documentary on the fascinating life of Ada Byron Lovelace, who had the vision to see the principles of computing 100 years before the appearance of the modern computer." -- Irvin Kershner, Director of "The Empire Strikes Back"

"In their beautiful documentary about Ada Lovelace, 'To Dream Tomorrow,' John Fuegi and Jo Francis have brought to light the life of an extraordinary woman and her contributions to the early history of computing." -- Karen Mathews, Executive Vice President, Computer History Museum

"The film was wonderful and I enjoyed every minute! The questions were very good and your answers were like the film: clean and superb." -- Professor Thomas Bergin, Editor, Annals of the History of Computing

"Just as Ada's father the poet Lord Byron had a genius for writing so Ada had a genius for practical mathematics, her application of which has been as great a contribution to the world of computers as that of her father to the world of literature." -- Geoffrey Bond, Chair, London Byron Society

"First class and very well done." -- Joel Shurkin, Pulitzer Prize Winner and author of Engines of the Mind

Deepak Kumar, Associate Professor of Computer Science, Bryn Mawr College, [dkumar@acm.org](mailto:dkumar@acm.org)

## Ada and Security Issues

From: Shmuel (Seymour J.) Metz  
 <shmuel+news@patriot.net>

Date: Fri, 21 Feb 2003 15:16:36 -0500  
 Subject: Re: status of PL/I as a viable language

Newsgroups: comp.lang.pl1,comp.lang.ada

> The problem is that those errors which are caught are gross errors of coding - buffer overruns, null pointer accesses, arithmetic overflow, etc. But serious errors in complex software aren't caused (only) by these sorts of errors.

True. But a huge proportion of the exploits in Internet software rely on buffer underruns, which wouldn't exist in a language like Ada [...]. And, yes, the checking can be turned off in Ada, but I would hope that such an abomination would never get through code review.

From: Berend de Boer <berend@xsol.com>

Date: Mon, 24 Feb 2003 08:19:01 +1300  
 Subject: Re: status of PL/I as a viable language

Newsgroups: comp.lang.ada

I'm pretty sure that with those checks included we would have zero buffer overflow errors and your credit card info would be a lot safer.

From: Kilgallen@SpamCop.net (Larry Kilgallen)

Date: 24 Feb 2003 06:00:06 -0600  
 Subject: Re: status of PL/I as a viable language

Newsgroups: comp.lang.ada

> Complex software has complex failure modes. Patting yourself on the back for catching buffer overruns is a little premature.

Patting oneself on the back for always avoiding automated tools for catching buffer overflows is also silly, and some C\* programmers do that. (Yes, I group C, C++, C# all together, because many of their advocates do, praising the ability to write C with a C++ compiler.)

## On Run-Time Checks in Operational Code

From: Randy Brukardt  
 <randy@rrsoftware.com>

Date: Fri, 21 Feb 2003 14:51:01 -0600  
 Subject: Re: status of PL/I as a viable language

Newsgroups: comp.lang.ada

[Upon a claim that run-time checks are fine for testing, but are better disabled in released applications in cases where continued operation is important, because

it's more likely that a program which "gets away" with making such an error can keep working, program away. -- dc]

Usually it is better to make the checks and prevent the "wrong answer". This is, after all, the cause of so many of the security holes on the Internet.

The web server for AdaIC is written in Ada. I've left all of the checking on, and provide a global exception handler for each worker thread. Thus, the worst that can happen for a mistaken check is the currently processed operation to be abandoned (the server sends an internal error response to the client). That prevents all sort of security holes from buffer overflows and the like.

It of course does not prevent all errors, but it allowed me to focus on blocking the common security problems that are algorithmic in nature, like traversal errors.

Although the code has had a variety of buffer overflow and other bugs cause individual operations to fail, the server has continued to run and process other operations correctly for the entire 18 months. (And, all failures have been logged so that the cause can be tracked down easily). This has allowed the focus to be on the web site's contents, not on keeping the server running.

There probably are cases where it is better to run with checks off. (We always did that with Janus/Ada for MS-DOS, just to keep the compiler size managable. That's not an issue on Windows.) But I would generally prefer to err on the side of leaving checks on unless it is necessary to do otherwise.

*From: Randy Brukardt*

*<randy@rrsoftware.com>*

*Date: Mon, 24 Feb 2003 14:15:19 -0600*

*Subject: Re: status of PL/I as a viable language*

*Newsgroups: comp.lang.ada*

> Buffer overflow? How?

In order to avoid dynamic allocation of commands, I put the parsed commands into a record type with a bunch of statically bounded strings. These have to be carefully handled for overflow, and there were a couple of obscure cases where I got it wrong. Of course, nothing bad happened, I just got a logged Constraint\_Error and the command was aborted. (Since the problems all came from people trying to break in, I wasn't even particularly upset that they weren't served...)

[And from a later message: -- dc]

> So in reality it wasn't a buffer overflow as Ada caught it, while if it had been in C you would have had a buffer overflow and possibly people breaking in. Did I understand it correctly?

Right. I think of "Constraint\_Error" being raised as a detected buffer overflow,

because it was trying to write outside of a buffer. Of course, its not actually doing any damage.

The point, of course, is that I didn't write any code to detect this failure. Indeed, if I had realized where I'd forgotten to protect against buffer overflow, of course I would have fixed it – adding checking code beyond that would be silly. So the "being a great programmer" would not have helped here - I had already tried to detect all of the overflow cases, but I missed a few. That's called being human.

*From: tmoran@acm.org*

*Date: Fri, 14 Mar 2003 02:30:25 GMT*

*Subject: Re: AdaIC opens the Ada Sites search engine*

*Newsgroups: comp.lang.ada*

> The AdaIC search engine is created with a set of Ada applications, all written in Ada 95. For more on the search engine and how to use it, see <http://www.adaic.com/site/search-info.html>.

[See also "AdaIC Opens Ada Sites Search Engine" in the "Ada-related Resources" section earlier in this AUJ. -- dc]

It's interesting to compare this to the recent:

> DeleGate versions prior to 8.5.0 do not properly handle large robot.txt files, thereby allowing a malicious Web site to execute arbitrary code on the DeleGate system. This vulnerability is confirmed and fixed in version 8.5.0. Source: SecurityFocus Bugtraq <http://archives.neohapsis.com/archives/bugtraq/2003-03/0160.html>

The early indexer didn't handle giant robots.txt files gracefully either. It just stopped and logged the problem however: no "arbitrary code execution" since it was written in Ada and no checks were suppressed.

## Selecting an Appropriate Language

*From: Warren W. Gay VE3WWG*

*<ve3wwg@cogeco.ca>*

*Date: Tue, 04 Feb 2003 12:59:13 -0500*

*Subject: Re: Bye-bye Ada ?*

*Newsgroups: comp.lang.ada*

[...], I'll confess that I don't believe that there are many appropriate "production level" applications that should `_today_` be written in Perl, FORTRAN or assembly language. Sure, device drivers are still appropriate, and for small segments of an operating system where squeezing the last byte out of the code is important (like a boot sector program segment).

Perl is OK for a quick and dirty "hack something together" to perform a one time job. Perl is absolutely the wrong language to use if someone at a later point in time, and especially a `_different_` someone, is stuck with maintaining it (it is IMHO, a

write-once, "hope you don't have to read it again" language). I've also witnessed endless problems with module version/compatibilities problems, on hosts where some people felt Perl was appropriate for production.

FORTRAN IMO, is OK for legacy stuff that is already in that language (tested and trusted), with the usual caveat that it depends on the application (it `_may_` never be good enough for space shuttles, and other critical stuff).

This is a long winded way of simply saying, I don't believe that there are many applications for those inferior tools today. I also believe that better tools exist (Ada95), and that too many people jump on the "efficiency bandwagon" instead (C/C++). [...] OTOH, `_reliability_` becomes increasingly important for general purpose computing as we try to build upon a foundation. It's hard to build on a shakey one. ;-)

*From: Richard Riehle*

*<richard@adaworks.com>*

*Date: Tue, 04 Feb 2003 21:26:18 -0800*

*Organization: AdaWorks Software Engineering*

*Subject: Re: Bye-bye Ada ?*

*Newsgroups: comp.lang.ada*

> I know \*I\* can write reasonably good software in C++. When we have errors, it's due to problems in logic, not problems in the language. But anyway, fine. Don't program in C++, do what you want.

Ah, and there's the issue. [...] The issue is not my preference, it is about choosing the correct tool for the job to be done.

It is also not about whether someone who is an expert using a particular set of tools can do the job better than someone who is not an expert. During a trip to Japan, I watched an expert in Japanese Joinery create the components for a Temple using only hand tools, his keen eye, and his many years of experience. Each component fit together with a perfection few others could achieve, even with more years of experience. This is an example of craftsmanship that few could match.

In my view, C++ is too prone to errors, even when used by experienced craftsmen. Those with the skill of the Japanese Joinery expert cited in the above paragraph are likely to produce defect-free code. We cannot depend on that level of expertise. It is rare. It is unpredictable. It is not easily confirmed. The more C++ code I see, and the more programmers I see trying to use it, in particular, for weapon-systems, the more concerned I have become about the potential for failure due to unpredictable behavior in the resulting code. It is so easy to compile a C++ program that behaves strangely after executing for some unspecified period of time.

My preference for Ada is not made out of ignorance of the other options. It is made because I have discovered that defect-free software is best created when the creational tools assist in preventing defects. At present, it is not a matter of `_can_` we create defect-free software in C++. We certainly can. It is, rather, what is the probability of the code being defect-free in C++ and in Ada. In my view, there is a greater likelihood of producing defect-free software in Ada than in C++.

It is not only a matter of reducing defects. When I look at Ada and then at C++, I realize that Ada provides other substantial benefits. In Ada, used well, I achieve a greater level of traceability than I can achieve in C++. Without resorting to embedded comments, I can code in Ada so the meaning of each construct is intuitive, unambiguous, and absolutely explicit, and do it more effectively than I can with C++.

As I compare the two languages in terms of long-term maintenance, it becomes clear, to me, that Ada supports this important goal far better than C++. It is more readable long after being created. There are far fewer little gotchas than I might expect in corresponding C++ code.

From where I sit, when making a decision about software that needs to live a long time, must be maintained over a long time, must work as predicted for a long time, it is clear that I must choose Ada over C++. It is the responsible thing to do. [...]

## On Mixed Language Development

*From: Frank Randolph CIV Beard  
<frank.beard@navy.mil>  
Date: Tue, 4 Feb 2003 08:29:55 -0500  
Subject: RE: One language environment  
don't have future  
Newsgroups: comp.lang.ada*

[On the claim that successful future systems will be one development environment for many call-compatible languages, a single run-time, single language-neutral set of libraries and OS interface. -- dc]

> Besides how will your VB programmer be able to change a part of a project written in C++, Ada, whatever, unless he also learns this language?

This is an interesting point that I meant to comment on. It leads to multiple points of failure. The project would have to be modularized. But this would only compartmentalize the failure point. If you lost your expert in any area, there would definitely be down time, learning curves, etc.

We have this problem on our current project. It was built with Ada, Delphi, and C++. Another related system used to build messages to send to it, was built using MS

VB. To maintain it, you need someone who knows each language fairly well. Otherwise, you could end up with kludged updates in the future.

*From: Preben Randhol <randhol@pvv.org>  
Date: Tue, 4 Feb 2003 13:34:55 +0000  
Organization: Norwegian university of  
science and technology  
Subject: Re: One language environment  
don't have future  
Newsgroups: comp.lang.ada*

*And if memory serves me well this was the  
(or at least one) initial concern that  
sparked the development of Ada :-)*

## On a Vertical Marketing Approach

*From: Richard Riehle  
<richard@adaworks.com>  
Date: Mon, 03 Feb 2003 09:26:34 -0800  
Organization: AdaWorks Software  
Engineering  
Subject: Re: Bye-bye Ada ?  
Newsgroups: comp.lang.ada*

> Develop something for a narrower industry that has a large software component and some other sort of value-added product (like an integrated cash register or some on-line service - that's where you make the \*real\* money) and you've got something where the big players aren't going to drive you out of the market.

This is called a "vertical marketing" approach. Develop expertise in a particular industry, learn its demands, its needs, and its way of doing business. Then, build a product that is easier to use, more reliable, and more closely mapped to the way that industry does business. Keep the product current as the technology advances. Keep the code portable for variations in operating system availability. Robert Lief was on the right track with his medical equipment in Ada approach. I'm not sure whether he ever got very far with it.

I have some friends in Silicon Valley who specialize in developing software for small medical devices on the I-8051. They are experts at this after many years of doing it. They quietly make money, don't have any intention of going public, love what they do, and have a clientele of satisfied and repeating business.

Ada, although not hosted on the I-8051, could be used to serve a similar market. A lot of lab equipment needs larger processors now. There are robotic assembly lines that need to run on larger processors. When one looks around carefully, it seems there is end to the opportunity for creating new software to upgrade what is already in place, and to sieze the market with reliable software in embedded industrial applications.

The problem is that too many people with Ada expertise are not of an entrepreneurial

bent. Long years in doing software by contract has stunted their ability to do anything except on-spec. The large users of Ada such as Lockheed, CSC, Raytheon, etc., simply don't have the kind of management that understands free-market risk-taking. I recall a conversation with one high-ranking official of one of the large DoD software developers. He asked what kind of commercial opportunities I saw for Ada. I listed several. His reply astonished me. "But who will fund it?" My answer, stunned him. "No one. You need to fund it yourself once you determine there is a market."

[And from a later message: -- dc]

The last thing you want is outside money. Stay away from Venture Capitalists. They will destroy your product, your will, your self-esteem, and everything you loved about what you were doing.

Software has the benefit of being a low resource product. One or two people can build a product in a short time. This is one place where some of the Agile Development ideas can benefit a couple of people with a great idea. If you have a corporate sponsor who will let you share in the harvest, good. But find a sponsor who has the same vision you have of the final product.

A good model to emulate is that of the founder's of Quicken products. Check out how they built their company. Even Bill Gates story includes some positive lessons, if one overlooks the devious methods he sometimes employs to get results.

> Most folks don't have the entrepreneurial spirit. Its not a crime. It just means they aren't likely to head in the direction that will lead them to building a successful business. The trick is to find a few who \*do\* have the right attitude and are willing to persue an idea through to completion.

Correct. Entrepreneurial means willingness to take risks. We take those risks with our own money, our own time, and our own psychological energy. It means a willingness to accept failure or success. If one is afraid of failure, and pursues a business opportunity out of fear, there is little likelihood of success. Each failure simply leads to the next adventure, and the potential for success. Resilience of spirit is the key.

Some of the most successful entrepreneurs I have known were people who lacked the formal education to realize that they could not succeed. Sometimes, we let our own knowledge get in the way of being what we could be. [...]

## On UML and Graphical Languages

*From: AG <ang@xtra.co.nz>  
Date: Sun, 9 Feb 2003 07:38:12 +1300*

*Subject: Re: One language environment don't have future*

*Newsgroups: comp.lang.ada*

> The software development community seems to disagree [that graphical notations are a step forward -- dc]. After many years on the market, graphical programming systems are still not widely used.

And not surprisingly. Compare it with normal languages: Historically, they seem to have migrated from pictorials to hieroglyphics to alphabet and plain text. The reason? Well, more flexibility, generality and more expressive power. Try to come up with a picture for the previous sentence for example ...

For some reason, computerese seems to be going in exactly the opposite way - having started with a sort of literate language it now drifts more and more towards simple (not to say stupid) pictures. [...]

> Well, considering that most of the information in books is currently textual, and has been for many centuries, it seems unlikely that pictorial representations of program logic will prove superior within our lifetimes.

Also, attempts to do that have been made decades ago: Block diagrams anyone? I'm still waiting for a compiler or IDE to draw that diagram and have the code generated which would be equivalent to say a few hundred classes written in text {well, I'm not - sarcasm y'know}

*From: Stephen Leake*

*<Stephen.A.Leake@nasa.gov>*

*Date: 09 Feb 2003 09:25:01 -0500*

*Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)*

*Subject: Re: One language environment don't have future*

*Newsgroups: comp.lang.ada*

> I'm not sure that textual representation of logic is the best there can be.

You may not be sure, but I'm sure Ada 95 is better than UML for "logic". State charts are nice for systems that have lots of states, and use cases are nice for all requirements analysis. The rest of UML I can easily live without.

*From: kcline17@hotmail.com (Kevin Cline)*

*Date: 10 Feb 2003 08:17:23 -0800*

*Subject: Re: One language environment don't have future*

*Newsgroups: comp.lang.ada*

> Modeling tools are able to produce code out of a skillfully crafted model in any language and they'll sooner or later have a Compile button.

Right, but for real-world problems there is an essential complexity that is not readily expressed graphically. The boxes and arrows take up lots of screen space but don't provide much information. The important information is captured by rules

like: "set shipping date two days after order date, skipping weekends and bank holidays" [and] "if the customer lives in a jurisdiction where we have a retail store, then add sales tax for that jurisdiction".

> And you will be able to tweak the generated code as much as you want.

I don't want to. Then I'm stuck working in two languages and worse, have to understand the mapping between them. [...]

## Opportunity for Ada Application

*From: Jerry Petrey*

*<jdpetrey@raytheon.com>*

*Date: Wed, 12 Feb 2003 10:09:47 -0700*

*Organization: Raytheon Company*

*Subject: Great opportunity for Ada application*

*Newsgroups: comp.lang.ada*

DARPA is sponsoring a competition with a one million dollar prize for an autonomous vehicle to transverse a course between L.A. and Las Vegas in 2004. This would be a great opportunity for an Ada controlled entry. See <http://www.darpa.mil/grandchallenge/> for details. Perhaps there is some small company around that would like to undertake this challenge and make a name for themselves and show what Ada can do.

## Ideas for an Ad?

*From: tmoran@acm.org*

*Date: Mon, 10 Mar 2003 03:22:12 GMT*

*Subject: Re: Way OT: Adam Smith and Software Markets*

*Newsgroups: comp.lang.ada*

[Someone joked "C is for Vulcans, Ada is for Humans" and another replied "... and Ferengis use Visual Basic." -- dc]

I can picture the ad: Two pictures with at the top with captions. One shows a dozen Vulcan-looking programmers at clean desks, manuals neatly stored, prettily indented, readable Ada code on their CRTS, caption "The programmers you wish you had". The other picture shows three "hot-shot" stereotype disheveled programmers at messy desks, one under the desk sleeping, another being distracted by a poster of an anatomically improbable female video game figure, and the third with head drooping, hand holding a can of Jolt. All three CRTs showing messy looking code with lots of {[:?( etc. Caption "The programmers you have". Under the two pictures the text says "Let Ada help you get correct, comprehensible code from merely human programmers." Then some testimonials: A citation of the error rates from the recent Spark Crosstalk article, something about productivity from the old Rational study (or Pratt & Whitney?), etc.

## Some Quotes

*From: tmoran@acm.org*

*Date: Wed, 05 Mar 2003 19:11:09 GMT*

*Subject: Re: Way OT: Adam Smith and Software Markets*

*Newsgroups: comp.lang.ada*

Darwin wouldn't have said "There are a lot of ants in the world, therefore ants must be the fittest creatures." In a niche with certain resources and problems, ants do very well. In another niche, in the same world, humans do quite well. If humans want to displace ants in a location, they have to modify the environment in human friendly, ant unfriendly, ways.

If we want Ada to displace C in some market, we need to modify that market - pointing out the dangers of C, encouraging software liability laws, getting government seed money, building convenient Ada libraries, whatever - in Ada friendly, C unfriendly, ways. [...]

*From: owski@hotmail.com (Adam Ruth)*

*Date: 10 Mar 2003 10:43:54 -0800*

*Subject: Re: Advertisement for Ada*

*Newsgroups: comp.lang.ada*

What I've been saying lately to my co-workers [...] is "Ada: Fewer bugs equals more money". It's a line that the non-techies can grasp.

*From: william.f.schreckenstein@lmco.com (Bill Schreckenstein)*

*Date: 25 Mar 2003 14:41:49 -0800*

*Subject: Re: Protest against the war on Iraq*

*Newsgroups: comp.lang.ada*

[...] as an Ada coder you have to appreciate the complexity of these systems. I can't even begin to describe how much easier a time I have had writing code in Ada and debugging it vs. C++. I hope this board keeps up the good work ... I have learned a lot from everyone posting here.

## Historical Note

*From: Laurent Guerby <guerby@acm.org>*

*Date: Sat, 12 Apr 2003 00:45:55 +0200*

*Subject: Historical note*

*To: team-ada@acm.org*

In a quite interesting article by Bob Frankston the author of Visicalc: <http://www.frankston.com/?name=ImplementingVisiCalc>

<<We started programming by using the tools from ECD on Multics. I worked at night when the computer time cost \$1/hour. Honeywell also took advantage of the low fee to use the machine at night to develop the Ada language for the military but those developers worked during their day from France.>>

This was around 1979. :)



## On the Rediscovery of Ada

*From: Marin David Condic  
<mcondic@acm.org>*

*Date: Fri, 28 Feb 2003 07:24:21 -0500*

*Subject: Re: the Ada mandate, and why it collapsed and died*

*Newsgroups: comp.lang.ada*

> [...] new people are discovering [Ada] and recognizing its expressive > power and its virtues over competing technologies. [...] to make a comeback, it will have to be because those who choose to use it are able to create excellent software and the results of their work is discovered by others. This is possible because Ada, in its current form, is still a better technology for software development than the competing technologies, the popularity of those competing technologies notwithstanding.

If we build things in Ada, this starts making it more viable. When you hack tools for your job, use Ada. When you look at new startup projects, use Ada. Go out and dream up a product you might sell to the world and use Ada to develop it.

Successful businesses that use Ada will do a lot more to spur Ada usage than anything else. It would create \*jobs\* that use Ada and people go where the money is. Most of us here believe in Ada as a better way. We need to demonstrate that to the world by using it to build better mousetraps rather than try to convince through academic arguments. I think that the development of some real-world apps with a little "Ada Inside" sticker in the corner would do a lot more to promote Ada than anything else. Is anybody out there building products with Ada to sell to people and generate jobs? [...]

*From: Richard Riehle  
<richard@adaworks.com>*

*Date: Fri, 28 Feb 2003 18:56:28 -0800*

*Organization: AdaWorks Software Engineering*

*Subject: Re: the Ada mandate, and why it collapsed and died*

*Newsgroups: comp.lang.ada*

[...] The fact that Ada continues to be a better technology for a large class of problems means that it is still there to be discovered. Indeed, we see more and more people rediscovering it even as some [US DoD] contractors, ignorant of the gold that lies at their feet, abandon it in favor of less worthy technologies. [...]

*From: Marin David Condic  
<mcondic@acm.org>*

*Date: Sat, 1 Mar 2003 08:13:54 -0500*

*Subject: Re: the Ada mandate, and why it collapsed and died*

*Newsgroups: comp.lang.ada*

[...] What I would like to see is an army of small-time developers building better products and shamelessly using Ada to do it. (Put that little "Ada Inside" sticker on

it! :-)) One could think big, like an Ada-OS or an Ada version of Star Office or an Ada version of Quicken (Is there a good open-source checkbook/accounting program out there?) One could think small, like tools for education (tutorial software for languages, math & statistics packages for academic uses, etc.) or hacker tools within the organization. One could think about the new products the company is planning and see how to get Ada to fit in there. Whatever angle it goes, the objective is to build really solid, reliable software and put that "Ada Inside" sticker on it.

The beauty of it is that if this army of small-timers goes out and makes reliable products and Ada starts gaining some visibility & reputation as a result, you'd have the DoD contractors looking at the "New Wave" and wondering how they missed the boat the first time around. The irony of it would be delicious, wouldn't it? :-)

*From: John R. Strohm*

*<strohm@airmail.net>*

*Date: Sun, 2 Mar 2003 10:27:05 -0600*

*Subject: Re: the Ada mandate, and why it collapsed and died*

*Newsgroups: comp.lang.ada*

Actually, there's a flip side.

If Ada really is as good as we all believe it is, if it really does offer the productivity gains we all believe it does, then a company that took Ada seriously for product development would quickly carve out a price, performance, profitability niche that no one else could touch. That company would, out of enlightened self-interest, KEEP QUIET about the fact that they were using Ada, treating it as competition-sensitive proprietary information. [...]

*From: Robert A Duff*

*<bobduff@shell01.TheWorld.com>*

*Date: 02 Mar 2003 15:03:18 -0500*

*Subject: Re: the Ada mandate, and why it collapsed and died*

*Newsgroups: comp.lang.ada*

I don't agree with that kind of strategy. The problem is that there is a great value in standards. If you're the last person on earth using Ada, and Ada is a huge benefit over whatever language is the popular, that benefit will be lost, because you're going to be providing all the funding for Ada compiler development, and all the training in Ada programming. Much better to get lots of folks (even your competitors) to use Ada, so you can share the cost of Ada compilers and so forth. [...]

*From: Marin David Condic*

*<mcondic@acm.org>*

*Date: Mon, 3 Mar 2003 13:04:38 -0500*

*Subject: Re: the Ada mandate, and why it collapsed and died*

*Newsgroups: comp.lang.ada*

[...] Ultimately, you can't really cover up that you use Ada – someone who works

for you will likely spill the beans somewhere along the line. And so what? Ada and its benefits aren't a secret, so you might as well use it as a kind of "Product Distinction". You build a better product by more than just the language, but you get the publicity and prestige of using Ada as you build name recognition with other companies doing similar.

## Advertisement for Ada

*From: Marin David Condic*

*<mcondic@acm.org>*

*Date: Sat, 8 Mar 2003 11:02:04 -0500*

*Subject: Re: Advertisement for Ada*

*Newsgroups: comp.lang.ada*

Since Ada isn't exactly sponsored by some large organization with an interest in promoting it, advertising gets to be problematic. I'd suggest that the best thing to do is put a little "Ada Inside" graphic in the corner of the main screen of the apps you build with Ada. If there are useful apps circulating around and they are well-built, reliable products, the fact that it is mentioned they were built with Ada will help to promote it.

It might not hurt to put some advertising into the "About" screens you create for your apps. An Ada graphic, some words describing why Ada was chosen to build this app, perhaps a link to the Adapower website... Anything that gives the potential user some insight into the notion of "Quality Software Through Ada". Of course, one had better build good, solid, reliable apps or the whole thing is counterproductive. Ada isn't going to stop anyone from building bad code. :-)

# Conference Calendar

This is a list of European and large, worldwide events that may be of interest to the Ada community. More information on items marked ♦ is available in the Forthcoming Events section of the Journal. Items in larger font denote events with specific Ada focus. Items marked with ☉ denote events with close relation to Ada.

The information in this section is extracted from the on-line *Conference announcements for the international Ada community* at: <http://www.cs.kuleuven.ac.be/~dirk/ada-belgium/events/list.html> on the Ada-Belgium Web site. These pages contain full announcements, calls for papers, calls for participation, programmes, URLs, etc. and are updated regularly.

---

## 2003

- July 13-16                    **22nd Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC'2003)** Boston, Massachusetts, USA. Topics include: design, specification, implementation, application and theory of distributed systems
- ☉ July 14-17                **OMG Annual Workshop on Real-Time and Embedded Distributed Object Computing**, Washington, DC, USA. Topics include: Applying CORBA in any real-time or embedded system; Integrating CORBA with legacy real-time and embedded systems; Evaluation of real-time, high confidence, and embedded middleware; Advanced scheduling techniques and high-level real-time programming models; Fault-tolerance issues in real-time and embedded systems; etc.
- ☉ July 21-25                **17th European Conference on Object-Oriented Programming (ECOOP'2003)** Darmstadt, Germany. Includes a.o. the following events:
- ☉ July 21-25            **ECOOP2003 - Workshop on Exception Handling in Object Oriented Systems (EHOOS'2003)**. Topics include: formalisation, distributed and concurrent systems, practical experience, design patterns and frameworks, practical languages (Java, Ada 95, Smalltalk, Beta), etc.
- ☉ July 21                **ECOOP2003 - 7th Workshop on Pedagogies and Tools for Learning Object-Oriented Concepts**. Topics include: frameworks/toolkits/libraries for learning support; approaches and tools for teaching design early; experiences with innovative CS1 curricula; etc.
- ☉ July 21                **ECOOP2003 - 13th PhD Students Workshop in Object-Oriented Systems (PhDOOS'2003)**. Topics include: Concurrent, real-time, parallel systems; Patterns; Distributed and mobile object systems; Language design and implementation; Programming environments; Software components; etc.
- July 21                 **ECOOP2003 - 8th International Workshop on Component-Oriented Programming (WCOP'2003)**.
- July 21                 **ECOOP2003 - 4th International Workshop on Object-Oriented Re-engineering (WOOR'2003)**.
- ☉ July 22                **ECOOP2003 - Workshop on Communication Abstractions for Distributed Systems (CADS'2003)**. Topics include: Communication abstraction in programming languages; Middleware services; Design Patterns for communication and distribution; Communication components; Authentication, authorization, privacy; Group-oriented communication; Tolerance of Partial Failures; etc.
- ☉ August 13-15            **16th International Conference on Parallel and Distributed Computing Systems (PDCS'2003)** Reno, Nevada, USA. Topics include: all areas of Parallel and Distributed Computing Systems, their modeling and simulation, design, use and performance, etc.
- ☉ August 24-27            **6th Joint Modular Languages Conference (JMLC'2003)** Klagenfurt, Austria. Topics include: concepts of well structured modular software; teaching good design and programming style; construction of large and distributed software systems; etc. Includes on 25-26 August, some common tutorials, keynote talks and a common panel with EuroPar'2003. Deadline for early registration: July 15, 2003

- © August 26-29      **9th International Conference on Parallel and Distributed Computing** (Euro-Par'2003) Klagenfurt, Austria. Topics include: all aspects of parallel and distributed computing. Includes on 25-26 August, some common tutorials, keynote talks and a common panel with JMLC'2003. Deadline for early registration: July 15, 2003
- © August 27-29      **4th International Conference on Parallel and Distributed Computing, Applications, and Techniques** (PDCAT'2003) Chengdu, PR China. Topics include: all areas of parallel and distributed computing, including Parallelizing compilers, Component-based and OO Technology, Programming languages and software tools, etc.
- September 01-05      **Joint 9th European Software Engineering Conference (ESEC) and 11th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE-11)** Helsinki, Finland. Includes a.o. the following event:  
                                  September 01-02 **ESEC/FSE-11 - International Workshop on Principles of Software Evolution** (IWPSE'2003). Topics include: methodology for evolutionary design and development, validation and verification of evolution, experience and lessons learned from evolutionary software systems, etc.
- September 02-05      **9th International Conference on Object-Oriented Information Systems (OOIS'2003)** Geneva, Switzerland
- September 03-05      **29th EUROMICRO Conference** (EUROMICRO'2003) Antalya, Turkey. Includes track on: Component-based Software Engineering (Component design, implementation, testing; Development environment and tools; Case studies and experience reports; Components for real-time and embedded systems; etc.)
- September 08-12      **11th IEEE International Requirements Engineering Conference (RE'03)** Monterey Bay, California, USA.
- September 08-14      **12th International Formal Methods Europe Symposium (FME'2003)** Pisa, Italy. Topics include: concerns and risks for potential adopters of formal methods; cost-benefit analysis; reports on practical use and case studies (reporting positive or negative experiences); tool support and software engineering; etc.
- September 15-19      **12th International Real-Time Ada Workshop (IRTAW-2003)** Viana do Castelo, Portugal. Participation on invitation only.
- September 16-19      **7th International IEEE Enterprise Distributed Object Computing Conference (EDOC'2003)** Brisbane, Australia. Topics include: Use and enhancement of middleware platforms; Practical experiences with enterprise distributed object computing; etc.
- September 22-25      **2nd International Conference on Generative Programming and Component Engineering (GPCE'2003)** Erfurt, Germany. In Cooperation with ACM SIGPLAN and SIGSOFT. Deadline for poster submissions: July 13, 2003
- September 22-26      **IEEE International Conference on Software Maintenance (ICSM'2003)** Amsterdam, The Netherlands. Includes a.o. the following event:  
                                  © September 23 **ICSM2003 - International Workshop on Evolution of Large-scale Industrial Software Applications (ELISA)**. Topics include: empirical studies of evolving software, scalability of technological solutions, evolution of open source and COTS software, etc.
- © September 23-26      **22nd International Conference on Computer Safety, Reliability and Security (Safecom'2003)** Edinburgh, Scotland, United Kingdom. Topics include: state-of-the-art, experience and new trends in the areas of computer safety, reliability and security regarding dependable applications of computer systems.
- September 25-26      **3rd International Conference on Quality Software (QSIC'2003)** Beijing, China. Topics include: Debugging; Economics of software quality and testing; Formal methods; Quality evaluation of software products and components; Reliability; Software quality education; Static and dynamic analysis; Testability; Testing of object-oriented systems; Testing of concurrent and real-time systems; Testing strategies, tools, processes, and standards; Tool support for improving software

quality; Validation and verification; Application areas such as component-based systems, distributed systems, embedded systems, enterprise applications, ...; etc.

- ☉ September 27-Oct. 01 12th **International Conference on Parallel Architectures and Compilation Techniques (PACT'2003)** New Orleans, LA, USA. Topics include: Programming languages for parallel scientific and object-oriented applications, etc.
- ☉ September 29-Oct. 02 **GI-Jahrestagung Informatik 2003 - Teiltagung "Sicherheit - Schutz und Zuverlässigkeit"** Frankfurt/Main, Germany. Topics include (in German): Software in sicherheitskritischen Anwendungen, Zuverlässigkeit und Sicherheit softwarebasierter Systeme, etc.
- September 30-Oct. 01 **IEEE International Symposium on Empirical Software Engineering (ISESE'2003)** Frascati, Italy. Topics include: strengths and weaknesses of technology in use and new technologies, etc.
- ☉ October 01-03 9th IEEE Computer Society's **International Workshop on Object-oriented Real-Time Dependable Systems (WORDS'2003)** Capri Island, Italy.
- ☉ October 06-08 **22nd Symposium on Reliable Distributed Systems (SRDS'2003)** Florence, Italy. Topics include: Distributed systems with properties such as reliability, availability, security, safety, and/or real time; Security and high-confidence systems; Analytical or experimental evaluations of dependable distributed systems; etc.
- October 06-09 **13th International Conference on Software Quality (ICSQ'2003)** Dallas, Texas, USA
- October 20-24 **6th International Conference on UML - Modeling Languages and Applications (UML'2003)** San Francisco, USA. Topics include: tool support for any aspect of modeling or model use; models in the development and maintenance process; domain-specific and concern-oriented modeling; etc.
- ☉ October 26-30 **18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'2003)** Anaheim, California, USA.
- ☉ October 27-31 **22nd Digital Avionics Systems Conference (DASC'2003)** Indianapolis, Indiana, USA. Software Engineering Track Topics include: Formal Specification and Verification; Software Reliability - Measurement and Techniques; Software Safety; Software Architecture - Evolution/Product Families; etc.
- October 28-31 **IEEE Symposia on Human-Centric Computing Languages and Environments (HCC'03)** Auckland, New Zealand. Topics include: tools that enable humans, using textual languages, visual languages or any other appropriate technologies, to accomplish their tasks more effectively; etc.
- ☉ October 30-Nov. 01 **International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES'2003)** San Jose, California, USA. Topics include: Compilation techniques that focus on embedded architectures; Design, specification, and analysis of embedded systems; Managed runtime environments for embedded systems; Memory management and compiler controlled memories; Software design for multiprocessor embedded systems; etc.
- ☉ November 03-07 **5th International Symposium on Distributed Objects and Applications (DOA'2003)** Sicily, Italy. Topics include: Applications of distributed-object technology; Design patterns for object-based components and applications; Interoperability between object systems and complementary technology; Real-time solutions for distributed objects; Scalability for distributed objects and object middleware; Security for distributed-object systems; Software engineering for distributed object-based applications; Technologies for reliable and fault-tolerant distributed objects; etc.
- November 17-20 **14th IEEE International Symposium on Software Reliability Engineering (ISSRE'2003)** Denver, CO, USA. Deadline for student paper and fast abstract submissions: August 1, 2003
- November 19-21 **6th IFIP International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS'2003)** Paris, France.
- November 19-21 **4th International IFIP Working Conference on Distributed Applications and Interoperable Systems (DAIS'2003)** Paris, France. In conjunction with FMOODS'2003.
- November 24-26 **2nd International Conference on Software for Embedded Systems (ICSTEST-E)** Bilbao, Spain. Topics include: Transportation and Safety-Critical Systems, Industry real experiences, Verification and Validation, Techniques for real time systems, Static and Dynamic analysis, etc.

- December 07-11      2003 **ACM SIGAda Annual International Conference** (SIGAda'2003) San Diego, California, USA. Sponsored by ACM SIGAda (ACM approval pending). In Cooperation With Ada-Europe and ACM SIGAPP, SIGCAS, SIGCSE, SIGPLAN, and SIGSOFT (Cooperation approvals pending). Topics include: Reliability needs and styles; Safety and high integrity issues; Improvements and additions to Ada to be included in Ada 200Y; Use of the Ada Distributed Systems Annex; Process and quality metrics; Testing and validation; Standards; Use of ASIS for new Ada tool development; Mixed-language development; Ada in XML environments; Ada in .Net environments; Quality Assurance; Ada & software engineering education; Commercial Ada applications: what Ada means to the bottom line; Use of SPARK and Ravenscar profile for high reliability software; Use of Real-Time CORBA; Real-time networking/quality of service guarantees; Fault tolerance and recovery; Distributed system load balancing; Static and dynamic code analysis; Performance analysis; Debugging complex systems; Integrating COTS software components; System Architecture & Design; Information Assurance in the age of terrorism.
- December 10      Birthday of Lady Ada Lovelace, born in 1815. Happy Programmers' Day!
- December 10-12      **European Software Process Improvement Conference** (EuroSPI'2003) Warwick, UK.
- December 10-12      10th **Asia-Pacific Software Engineering Conference** (APSEC'2003) Chiang Mai, Thailand. Topics include: Object-Oriented Technology and Patterns, Software Components, Programming Languages and Software Engineering, Reliability and Security, Formal Methods, Software Development Environments, Distributed and Parallel Systems, etc.

---

## 2004

- January 01/05-08      Software Technology Track of the 37th **Hawaii International Conference on System Sciences** (HICSS-37) Big Island of Hawaii, USA. Includes mini-tracks on: Distributed Object and Component-based Software Systems (Design Patterns for Distributed Systems, Middleware, Programming Languages and Environments for Distributed Object and Component Systems, ...); etc.
- January 03/29-April 02      **European Joint Conferences on Theory and Practice of Software** (ETAPS'2003) Barcelona, Spain. Event includes: conferences from 29 March to 2 April, 2004, affiliated workshops on 27-28 March and 3-4 April, 2004
- January April 05-07      4th **International Conference on Integrated Formal Methods 2004** (IFM'2004) Canterbury, Kent, England. Deadline for submissions: September 15, 2003

# Object-Oriented Programming Enhancements in Ada 200Y

*S Tucker Taft*

*SofCheck, Inc. 11 Cypress Drive, Burlington, MA 01803; USA.; Tel: +1 781 750 8068; email: stt@sofcheck.com*

## Abstract

*This article provides an overview of four proposed amendments to the Ada standard for possible inclusion in the revision planned for late 2005 or early 2006. Together, these four amendments can be seen as "finishing" the job of integrating object-oriented programming features into Ada.*

*Keywords. Ada, Object-Oriented Programming, Amendment*

## 1 Introduction

A new revision of the Ada programming language standard is being prepared, with a scheduled completion date of late 2005 or early 2006. As part of this revision, the Ada Rapporteur Group (ARG), a part of the ISO Working Group 9 (WG9), is developing proposed amendments to the standard. Several of these amendments relate to object-oriented programming (OOP). This paper will describe some of these amendments, and the background and rationale for their development.

When Ada 95 was being designed, there was still a fair amount of controversy whether object-oriented programming features should be included in the language at all, because of their generally dynamic nature, and because of concern about whether some of their perceived negative aspects (difficult to test and verify, "weaker" typing model, etc.) might outweigh their claimed positive aspects.

Over the past decade, object-oriented programming has become the dominant programming paradigm, so much so that it is now simply assumed, and debates have moved on to other language and methodology issues (e.g. aspect-oriented programming, extreme programming, highly scalable programming, etc.). Two major new object-oriented programming languages have appeared on the scene, Java and C#. And most colleges and high schools are now teaching an object-oriented programming language in their introductory programming courses.

Hence, there is no longer any significant debate whether adding object-oriented programming to Ada 95 was a good idea. The question that remains is whether the object-oriented programming features of Ada 95 are as usable, effective, and understandable as they should be.

## 2 Differences Between Ada 95 and Other OOP Languages

Before attempting to answer this question, it is useful to first identify what makes Ada 95's object-oriented

programming features different from those of most other OOP languages, both in a positive and a negative sense. There are several important differences:

a) Ada 95 makes a significant and explicit distinction between class-wide types and specific types. This distinction implicitly exists in essentially all OOP languages, but there is rarely a way to talk about it in the source language itself. Instead, depending on context, a type or class name in such a language might represent a single type in the hierarchy (what Ada 95 calls a "specific" type), or it might represent a type and all types derived directly or indirectly from it (what Ada 95 calls a "derivation class of types").

Only when dealing with class-wide types in Ada 95 is there any possibility of dynamic binding. In most other OOP languages, dynamic binding is the default, and static binding requires additional effort, or is simply not available. This makes it more likely in such languages that dynamic binding will be used in places where static binding would have been preferred, and would have produced a faster, more verifiable, and more maintainable system.

In Ada 95, because static binding is the default, there will generally be significantly reduced coupling between a derived type and its parent type, allowing the parent operations to be treated more like black boxes. In most other OOP languages, you really need to see the source code for all parent operations to know whether it is safe to inherit any one of them rather than override it in a derived type.

b) Ada 95 has no direct linguistic support for type hierarchies involving multiple inheritance. Although there are several other language features (such as "with" and "use" clauses, generic packages, private extensions, and access discriminants), that allow programmers to solve problems in Ada 95 for which other languages might rely on their linguistic multiple inheritance capabilities, there are still some situations where the lack of linguistic support does restrict the ease of solving an important problem.

c) Except for synchronizing operations (such as a task entry call or a protected operation), all operands to an operation in Ada 95 are treated symmetrically in the syntax. That is, they are all passed as parameters "inside" the parentheses, independent of whether the operand might control dynamic binding.

This symmetry makes object-oriented abstract data types be a natural generalization of "normal" abstract data types, and makes user-defined binary operators work in a natural way

with such types, without any special treatment. The controlling operand of a binary operator could be the right operand or the left operand, depending on what is appropriate. The controlling "operand" can even be provided by context, in the case of a call on a parameterless function like "Empty\_Set" which will result in the invocation of the "appropriate" overriding of Empty\_Set, depending on the underlying run-time "tag" of the "receiver" of the result of the call.

Unfortunately, this symmetric approach can result in extra verbiage and possible confusion when used with a multi-package type hierarchy. Some "operations" in such a hierarchy might be so-called "class-wide" operations, which are generally declared in the package where the root type of the hierarchy is declared, while others will be "dispatching" operations which are inherited down the hierarchy, and are implicitly declared within each package where a derived type is declared. To call an operation, one has to either have "use" clauses for all packages where it might have been declared, or determine the correct package and put a prefix on the operation name that identifies the relevant package. Although this does not at first glance seem onerous, when working with relatively large type hierarchies, always identifying the package or "use"ing all the relevant packages can make the code less rather than more readable.

With OOP languages that use the "asymmetric" approach, where the (one and only) controlling operand precedes the name of the operation, and the other operands appear inside the parentheses, there is rarely a need to identify the module where an operation is declared, since it is determined by the type of the controlling operand. In C++, the module name is used generally only when overriding the default dynamic binding, and requesting static binding to an operation in a particular class/namespace.

There are some languages, in particular Modula-3, which allow either notation to be used, with the asymmetric "prefix" notation being a short-hand (syntactic "sugaring") for the symmetric notation.

**d)** Ada separates declaration from implementation, and requires that all types and operations be declared before they are referenced. In some OOP languages, in particular Eiffel and Java, declaration and implementation are not separated in the definition of a class. Furthermore, in these languages, in part because all objects are referenced via pointers and hence are of known "size," there is no need to declare a class before it is referenced.

Because Ada requires declaration before reference, extra work is required to create collections of types that are mutually dependent. In general, an incomplete type declaration is required to allow for such cyclic type structures. However, an incomplete type must be completed within the same package in which it is declared. This precludes such cyclic type structures from crossing multiple packages, and tends to lead to larger-than-ideal packages simply to accommodate such a cycle. The child library unit feature was added to Ada 95 in part to allow

packages to remain smaller, with hierarchies (subsystems) of packages being used to represent large multi-type abstractions.

C++ retains the separation between declaration and implementation, while allowing cyclic type structures to cross multiple "namespaces." This is possible because namespaces may be defined in several separate textual pieces, and an incomplete type declaration in C++ may be in one piece of the namespace, while a separate piece contains the full type declaration. In Ada, packages have only two textually separable pieces, namely the package declaration ("spec") and the package implementation ("body"). But putting a full type declaration in the package body is not a solution to the multi-package cyclic type structure problem, because the declarations within the package body are not visible outside the package. By contrast, all the "pieces" of a C++ namespace can contain "visible" declarations.

**e)** Ada 95 supports 3 levels of visibility for operations and components of a type: fully public, visible to child units, and visible only within the defining package. Most other OOP languages provide special visibility of operations to derived types (subclasses). In C++ and Java this is called "protected" visibility.

An important advantage of the Ada 95 approach to "partial" visibility is that it is provided only to modules whose position within the naming hierarchy implies their special visibility. This creates a strong boundary around the set of units that might be affected by changes to partially visible operations or components. In most other OOP languages, this special visibility is unrelated to the module structure, and a derived type/subclass which might be affected by changes to partially visible operations or components could be in any module, anywhere in the system.

The net effect is that encapsulation and information hiding in Ada 95 is linked more closely to the naming hierarchy, making maintenance of Ada object-oriented systems easier to perform, even when the systems grow large and involve large hierarchies of types.

**f)** Ada 95 supports both object-oriented programming and multi-threaded programming, but does not directly integrate these two. Tasks and protected objects can be components of an object-oriented "type," or vice-versa, but neither tasks or protected objects can themselves be directly extended. By contrast, in Java, which is one of the very few other languages that have linguistic support for both object-oriented programming and multi-threading, synchronizing operations can be added in subclasses, and the types used to represent threads can also similarly be extended using the normal inheritance mechanisms.

An important advantage of Ada's tasking model is that all operations of a protected type or a task synchronize properly with one another, while in Java, it is possible to have both synchronizing and non-synchronizing operations on the same type, which is an obvious avenue for subtle race conditions to enter a system. Furthermore, because Ada's protected and task types do not allow piecemeal

inheritance, all operations that synchronize with one another are defined in the same module, preserving the original advantages of the "monitor" concept introduced many years ago -- analysis and verification of proper synchronization conditions can be performed without having to chase down all critical sections that might be scattered about the system.

Given the above important differences between Ada 95 and most OOP languages, it is appropriate to evaluate these differences, and see whether they represent strengths or weaknesses in Ada's support for object-oriented programming. In some cases, the differences have both positive and negative aspects. Arguably one overall negative aspect of such differences is that they may put Ada 95 out of the mainstream of object-oriented programming, given that more and more programmers are being introduced to OOP, or even programming as a whole, through languages like Java and C#. On the other hand, Ada 95 has an important role in the development of complex, critical systems, and some of the differences are specifically designed to assist in the development of safe, robust, and verifiable systems, while still providing the flexibility and extensibility of object-oriented programming.

### 3 Areas of Strength, Areas for Enhancement

The challenge for this upcoming revision of Ada is then to preserve Ada's great strengths in its support for the construction of safe, verifiable systems, while enhancing its object-oriented features to take advantage of what has been learned about object-oriented programming features over the past ten years. Areas that have been identified for possible enhancement are support for multi-package cyclic type structures, support for multiple-inheritance type hierarchies, support for the "asymmetric" notation for invoking operations, and support for some kind of extension for protected and task types.

On the other hand, Ada's clear distinction between specific and class-wide types, its default of static binding with dynamic binding only where necessary, and its strong boundary around modules that have visibility on "partially" visible operations and components, are seen as clear advantages to Ada's approach to object-oriented programming, with no need for significant alteration. Furthermore, any changes that are proposed must not compromise Ada's strengths, and if anything, should extend Ada's unique position as the safe and verifiable, real-time object-oriented programming language.

### 4 Cyclic Type Structures

One item identified as very important for enhancement has to do with allowing cyclic type structures to cross package boundaries. In Ada 95, it is possible to use a combination of class-wide types, type extension, and "downward" type conversion, to overcome the basic Ada limitation to single-package cyclic type structures. However, this approach introduces additional complexity and some degree of run-

time overhead and possible sources of run-time errors. Hence, there has been a concerted effort to provide a natural way for cyclic type structures to be safely and securely extended across packages.

Several alternative proposals have been developed and evaluated. Unfortunately, no one proposal has emerged as clearly the best solution in every dimension. The original proposal introduced a new kind of "with" clause called the "with type" clause. This allowed a package to refer to a type that would eventually be declared in some other package, but without requiring that that other package be compiled first. A version of this proposal was actually implemented in the GNAT Ada Compiler from AdaCore Technologies, but was ultimately dropped from consideration by the ARG because of difficulties discovered while working out the lower level details.

Three proposals remain under consideration: one involving type "stubs" (analogous to program unit "stubs", identified by the "is separate" syntax), a second involving a generalization of incomplete type declarations to allow a parent package to declare an incomplete type that will be completed in a child or nested package, and a third proposal involving a new kind of "limited" with clause, allowing one package to gain visibility on the types and nested packages of another package, without requiring "full" compilation of the other package.

Here are examples of the three proposals. They all are based on the Employee/Department problem, where there is a type that represents employees, and a type that represents departments, and employees are members of a department, while a department has a manager who is an employee. The challenge is to define the employee type in one package, and the department type in a separate package, but accommodate the desire to have references to both employees and departments in both packages.

The first example is the "type stub" proposal:

```

limited with Employees;
-- Allow type stubs to refer to this package

package Departments is
  type Employee is separate Employees.Employee;
  -- Type stub
  type Employee_Ref is access Employee;
  type Department is private;

  procedure Set_Manager(Dept: in out Department; Mgr:
Employee_Ref);

  function Manager(Dep: Department) return Employee_Ref;
  ...
private
  type Department is record
    Mgr: Employee_Ref;
    ...
  end record;

```



```

end Departments;

limited with Departments;
-- Allow type stubs to refer to this package
package Employees is
  type Department is separate Departments.Department;
  -- Type stub
  type Department_Ref is access Department;
  type Employee is private;
  procedure Set_Department(Emp: in out Employee;
Dept: Department_Ref);
  function Department(Emp: Employee) return
Department_Ref;
  ...
private
  type Employee is record
    Dept: Department_Ref;
    ...
  end record;
end Employees;

```

The second example uses the generalized incomplete type declaration:

```

package Office is
  type Employees.Employee;
  -- Incomplete type completed in child
  type Employee_Ref is access Employees.Employee;
  type Departments.Department;
  -- Incomplete type completed in child
  type Department_Ref is
access Departments.Department;
end Office;

  package Office.Departments is
  type Department is private;
  procedure Set_Manager(Dept: in out Department;
Mgr: Employee_Ref);
  function Manager(Dep: Department)
return Employee_Ref;
  ...
private
  type Department is record
    Mgr: Employee_Ref;
    ...
  end record;
end Office.Departments;

```

```

package Office.Employees is
  type Employee is private;
  procedure Set_Department(Emp: in out Employee;
Dept: Department_Ref);
  function Department(Emp: Employee) return
Department_Ref;
  ...
private
  type Employee is record
    Dept: Department_Ref;
    ...
  end record;
end Office.Employees;

```

The third example uses the "limited with" clause:

```

limited with Employees;
-- Gives visibility on types as incomplete types
package Departments is
  type Employee_Ref is access Employees.Employee;
  type Department is private;
  procedure Set_Manager(Dept: in out Department;
Mgr: Employee_Ref);
  function Manager(Dep: Department)
return Employee_Ref;
  ...
private
  type Department is record
    Mgr: Employee_Ref;
    ...
  end record;
end Departments;

limited with Departments;
-- Gives visibility on types as incomplete types
package Employees is
  type Department_Ref is
access Departments.Department;
  type Employee is private;
  procedure Set_Department(Emp: in out Employee;
Dept: Department_Ref);
  function Department(Emp: Employee)
return Department_Ref;
  ...
private
  type Employee is record
    Dept: Department_Ref;

```

```
...
end record;
end Employees;
```

All three proposals allow a type defined in one package to be treated as an incomplete type in some other package, without the second package "depending" semantically on the first package. This is the critical capability, because it allows a cyclic type structure to be established without contradicting the partial ordering implied by "normal" semantic dependence relationships. All of the solutions involve a "weaker" kind of dependence, where one package knows that another package "exists" without having full semantic dependence on it. The "limited" with clause proposal approaches this problem by introducing a "limited" dependence on another package. Limited dependences are allowed to be cyclic. They imply some kind of pre-scan of a package to determine the names of the types (and the subpackages) of the package, without doing a full semantic analysis of the package.

The type stub proposal also requires a similar kind of limited dependence, but limits it even further to specific types identified by type stubs. Further, it does not require any kind of pre-scan of the package, because post-compilation checks can be performed to verify that type stubs refer only to types that actually exist in the package.

The incomplete-type-completed-in-a-child proposal introduces a "weak" dependence between a parent package and one of its child packages, requiring that a child package exist and that it declare a type that matches one identified in a generalized form of incomplete type declaration present in the parent's specification.

At this point there is consensus that a solution to this problem will exist in the Ada 200Y standard, and that the form of the solution will be based on one of these three proposals, but the particular approach has not yet been chosen. It is anticipated that the final choice will be made at the ARG meeting immediately following the AdaEurope 2003 conference.

## 5 Multiple-Inheritance Type Hierarchies

When Ada 95 was designed, a significant amount of energy was expended in evaluating the possibility of including direct syntactic support for multiple inheritance. At the time, some OOP languages included full multiple inheritance (C++, Eiffel), while others chose single inheritance (Modula-3, Smalltalk). Full multiple inheritance introduces a number of language complexities as well as a somewhat more complicated and/or less efficient run-time model for dispatching calls. Ultimately, we decided to stick with the simplicity of single inheritance for Ada 95, but provide various "building blocks" that could be used to solve problems that in other languages might require multiple inheritance.

Since the Ada 95 design was finalized, a middle ground in the spectrum of inheritance models has become popular that provides multiple inheritance of interfaces (i.e.

contracts), but with actual implementation "code" and data components inherited from only a single "primary" parent type. This approach, as exemplified in Java, C#, and to some extent CORBA IDL, eliminates much of the complexity of "full" multiple inheritance, because data components can continue to use the straightforward linear extension approach of single inheritance, and because conflicts due to inheriting code from multiple parent types cannot occur.

The current proposal for adding multiple inheritance of interfaces adds a new kind of type to Ada called an "interface". An interface type is in most respects equivalent to a type declared as "type T is abstract tagged null record;" though the syntax is shortened to be simply "type T is interface;". However, in addition to being usable in all contexts where such an abstract type may be used, the type may also be used as a "secondary" parent type in the declaration of a type extension. Secondary parents ("interface parents") are identified by appearing second or later in a list of the parent types in a record extension. The parent type names are separated from one another by "and", as in:

```
type NT is new Primary and Secondary_1 and
Secondary_2 and ... with ...;
```

Note that the Primary parent may also be an interface type, since an interface type may be used anywhere an abstract tagged type may be used.

Interfaces may also be used as "parents" of other interfaces, using the following form:

```
type NI is interface with Int_Parent1 and Int_Parent2;
```

As implied above, no code or components are inherited from interfaces, only the specification of operations that must be implemented by the type that has the interface as a parent. If an interface has other interfaces as parents, then the union of all the operations of the parents combined with the operations defined on the new interface must be implemented by all (non-abstract) types derived from the new interface.

Here is a larger example which uses interfaces:

```
package MVC is
-- Set of interfaces that define a model-view-controller
-- structure.
type Observer is interface;
-- "interface" is roughly equivalent to
-- "abstract tagged null record"
type Observer_Ref is access all Observer'Class;
-- An observer waits for changes to a model
type Model is interface;
type Model_Ref is access all Model'Access;
-- A model represents some data structure
-- that is being viewed and/or manipulated
```

```

procedure Notify(
  Obs: access Observer;
  M: Model_Ref) is abstract;
-- Notify observer that model it was observing
-- has changed
type View is interface with Observer;
type View_Ref is access all View'Class;
-- A view is a visual display of some model
procedure Display_View(
  V: access View;
  M: Model_Ref) is abstract;
-- Display view of associated model
type Controller is interface with Observer;
type Controller_Ref is access all Controller'Class;
-- A controller supports input device(s) for
-- manipulating/updating an underlying model
procedure Start_Controller(
  Ctr: access Controller;
  M: Model_Ref) is abstract;
-- Initiate controller for associated model
procedure Register_View(
  M: access Model;
  V: View_Ref) is abstract;
-- Register view for given model.
procedure Register_Controller(
  M: access Model;
  Ctr: Controller_Ref) is abstract;
-- Register controller on given model.
end MVC;

with MVC;
with Devices;
package Inputs is
  type Mouse is new Devices.Device with private;
  type Mouse_Controller is new
    Devices.Device and MVC.Controller with private;
-- Primary parent type, if any, must be listed first
-- All other parent types must be interfaces.
procedure Handle_Input(
  MC: in out Mouse_Controller);
-- Optionally override operations of parent type
-- (or may inherit those with appropriate defaults)
procedure Notify(
  MC: access Mouse_Controller;
  M: Model_Ref);
procedure Start_Controller(
  MC: access Mouse_Controller;
  M: Model_Ref);
-- Required to define all abstract operations declared
-- for Observer and Controller
type Two_Button_Mouse_Controller is new
  Mouse_Controller with private;
procedure Start_Controller(
  TMC: access Two_Button_Mouse_Controller;
  M: Model_Ref);

```

```

-- May inherit or override operations inherited from
-- parent type including those that are needed for
-- interfaces Observer and Controller
procedure Register_And_Start(
  MC: access Mouse_Controller'Class;
  M: Model_Ref);
-- Class-wide operation to register the mouse
-- controller on given model, and then start the
-- controller going.
private
  ...
end Inputs;

```

Although not illustrated in the above example, the proposal for interface types includes a proposal for "declared-null" procedures. A declared-null procedure is one whose specification ends with "is null," rather than ";" or "is abstract;". No separate body is permitted for such a procedure. The implicit null body has no effect when executed.

Rather than requiring that all primitive operations of an interface type be abstract, this proposal also allows the primitive operations to be declared null. Such a procedure need not be overridden in a type derived from this interface. If not overridden, its implementation is null. If at least one interface ancestor of a type declares a given operation as null, the type need not provide an explicit overriding of the operation. If a non-interface ancestor type provides a non-null implementation of the operation, that is inherited rather than the null procedure.

Declared-null procedures are useful in that they allow a number of optional capabilities to be supported in an interface, without every derived type having to explicitly define the capability. In addition, if an abstract or interface type with one or more declared-null primitives is used as the ancestor in a generic formal type extension, the formal type is presumed to have non-abstract implementations of these operations. This can be useful when overriding the operations, since it is often desirable to call the parent's operation from an overriding, particular in the case of initializing or finalizing operations.

## 6 Using Object.Operation Notation

When doing object-oriented programming in Ada 95, the programmer must identify the package in which an operation is declared, along with the various operands. Because dispatching operations are often implicitly declared, identifying the package where they are declared can sometimes be confusing. In addition to dispatching operations, class-wide operations are important in many object-oriented systems. However in Ada, class-wide operations, unlike their equivalent in many other OOP languages, are not inherited along with the dispatching operations. Instead, they are only declared in the original package where they appear.

This distinction in inheritance between dispatching operations and class-wide operations means that it can be a burden to identify the package where an operation of interest is declared, particularly when the choice between making an operation a dispatching operation versus a class-wide operation might be more of an implementation detail than an essential part of the semantics of the operation from a user's point of view. The distinction is generally important when deriving from a type, but may be irrelevant when using the type.

Programmers familiar with other OOP languages that use an "object.operation(...)" syntax rather than Ada's "package.operation(object, ...)" syntax find this added burden an entry barrier to using Ada for OOP systems, which tends to make the language feel less object-oriented than it truly is. The alternative of inserting "use" clauses for every possible package where an operation might be declared has other negative ramifications.

Given these considerations, a proposal has been developed to allow the use of an "object.operation(...)" syntax as a syntactic shorthand for "package.operation(object, ...)". Originally it was proposed that this syntactic shorthand be available to all kinds of types, whether or not the type is tagged. However, supporting this for both access types and tagged types adds to the complexity of the proposal in certain ways due to the desire to allow implicit dereference (implicit ".all") of the "object" if it is designated by an access value. Implicit dereference is provided in all other places where "." is allowed in the syntax, and it would be inconsistent not to allow it here. Furthermore, this notation is specifically intended to simplify object-oriented programming where there may be multiple relevant packages. When using non-tagged types, the object.operation syntax would not provide as much benefit.

The basic idea of this restricted proposal is that any dispatching operation, or any class-wide operation declared in a package where the corresponding specific type is declared, is eligible for calling via this shorthand, so long as the first formal parameter is a controlling parameter, or is of the class-wide type. When the object.operation syntax is used, the "operation" is looked up first as a component, and then as though the packages where the type and any of its ancestors are declared had been made use-visible. If the object were of an access-to-tagged type, an interpretation using an implicit dereference would also be considered. If there are possible interpretations of "operation" among these packages, it is checked to see if any of them are subprograms where "object", "object.all", or "object'access" could be passed as the first parameter, and any actual parameters given in parentheses after "operation" correspond to the remaining formals.

Here are some examples of use of this shorthand:

Given the MVC and Inputs packages given above:

```
M : MVC.Model_Ref;
V : MVC.View_Ref;
```

```
C : MVC.Controller_Ref;
MC : aliased Inputs.Mouse_Controller;
begin
  V.Display_View(M);
  -- equivalent to MVC.Display_View(V, M);
  MC.Start_Controller(M);
  -- equivalent to Inputs.Start_Controller(MC'Access, M);
  MC.Handle_Input;
  -- equivalent to Inputs.Handle_Input(MC);
  MC.Register_And_Start(M);
  -- equivalent to
  -- Inputs.Register_And_Start(MC'Access, M);
  -- (this is a call on a class-wide op)
```

## 7 Inheritance of Interfaces for Protected and Task Types

During the Ada 95 design process, it was recognized that type extension might be useful for protected types (and possibly task types) as well as for record types. However, at the time, both type extension and protected types were somewhat controversial, and expending energy on a combination of these two controversial features was not practical.

Since the design, however, this lack of extension of protected types has been identified as a possible target for future enhancements. In particular, a concrete proposal appeared in the May 2000 issue of ACM Transactions on Programming Languages in Systems (ACM TOPLAS[1]), and this has formed the basis for a language amendment (AI-00250).

However, in ARG discussions, the complexity of this proposal has been of concern, and more recently a simpler suggestion was made that rather than supporting any kind of implementation inheritance, interfaces for tasks and protected types might be defined, and then concrete implementations of these interfaces could be provided. Class-wide types for these interfaces would be defined, and calls on the operations (protected subprograms and entries) defined for these interfaces could be performed given only a class-wide reference to the task or protected object.

An important advantage of eliminating inheritance of any code or data for tasks and protected types is that the "monitor"-like benefits of these constructs are preserved. All of the synchronizing operations are implemented in a single module, simplifying analysis and avoiding any inheritance "anomalies" that have been associated in the literature with combining inheritance with synchronization.

The detailed syntax for protected and task interfaces has not been proposed. Here is one possibility:

```
protected interface Queue is
  -- Interface for a protected queue
  entry Enqueue(Elem : in Element_Type) is abstract;
  entry Dequeue(Elem : out Element_Type) is abstract;
```

```

function Length return Natural is abstract;
end Queue;
type Queue_Ref is access all Queue'Class;

protected type Bounded_Queue(Max: Natural) is new
  Queue with
-- Implementation of a bounded, protected queue
  entry Enqueue(Elem : in Element_Type);
  entry Dequeue(Elem : out Element_Type);
  function Length return Natural;
private
  Data: Elem_Array(1..Max);
  In_Index: Positive := 1;
  Out_Index: Positive := 1;
  Num_Elems: Natural := 0;
end My_Queue;

task interface Worker is
  -- Interface for a worker task
  entry Queue_To_Service(Q : Queue_Ref)
  is abstract;
end Worker;
type Worker_Ref is access all Worker'Class;

task type Cyclic_Worker is new Worker with
  -- Implementation of a cyclic worker task
  entry Queue_To_Service(Q : Queue_Ref);
end Cyclic_Worker;

task Worker_Manager is
  -- Task that manages servers and queues.
  entry Add_Worker_Task(W : Worker_Ref);
  entry Add_Queue_To_Be_Serviced(Q : Queue_Ref);
end Worker_Manager;

task body Worker_Manager is
  Worker_Array : array(1..100) of Worker_Ref;
  Queue_Array : array(1..10) of Queue_Ref;
  Num_Workers : Natural := 0;
  Next_Worker : Integer := Worker_Array'First;
  Num_Queues : Natural := 0;
  Next_Queue : Integer := Queue_Array'First;
begin
  loop
  select
    accept Add_Worker_Task(W : Worker_Ref) do
      Num_Workers := Num_Workers + 1;
      Worker_Array(Num_Workers) :=
        Worker_Ref(W);
    end accept;
    or
    accept Add_Queue_To_Be_Serviced(
      Q : Queue_Ref) do
      if Num_Queues > 0 then
        -- Assign next queue to this worker
        Worker_Array(Num_Workers).
          Assign_Queue_To_Service(
            Queue_Array(Next_Queue));
        -- Dynamically bound entry call
        -- Advance to next queue
        Next_Queue := Next_Queue mod
          Num_Queues + 1;
      end if;
    or
    accept Add_Queue_To_Be_Serviced(
      Q : Queue_Ref);
      Num_Queues := Num_Queues + 1;
      Queue_Array(Num_Queues) := Queue_Ref(Q);
    end accept;
  end select;
  end loop;
  -- Assign queue to worker if enough workers
  if Num_Workers >= Num_Queues then
    -- This queue should be given one
    -- or more workers
    declare
      Offset : Natural := Num_Queues-1;
    begin
      while Offset < Num_Workers loop
        -- (re) assign queue to worker
        Worker_Array((Next_Worker + Offset -
          Num_Queues) mod Num_Workers + 1).
          Assign_Queue_To_Service(
            Queue_Array(Num_Queues));
        -- Dynamically bound call
        Offset := Offset + Num_Queues;
      end loop;
      -- Advance to next worker
      Next_Worker := Next_Worker mod
        Num_Workers + 1;
    end;
  end if;
end Worker_Manager;

  My_Queue : aliased Bounded_Queue(Max => 10);
  My_Server : aliased Cyclic_Server;
begin
  Worker_Manager.Add_Worker_Task(
    My_Server'Access);
  Worker_Manager.Add_Queue_To_Be_Serviced(

```

```
My_Queue'Access);  
...  
end;
```

## 8 Summary

The four proposed amendments to the Ada standard discussed above are in some sense an attempt to "finish" the job of integrating object-oriented programming into Ada which was started during the Ada 95 revision process. Although the existing OOP features in Ada 95 are both powerful and flexible, eight years of use and ongoing developments in the object-oriented programming language community have suggested opportunities for enhancement.

Although it is likely that some of these amendments will be approved for addition to the standard, it is quite possible that some will not, or that the proposals will be further refined in minor or major ways. Hence it is essential to keep in mind that this is a snapshot of an ongoing revision process, and by no means the final story. For those interested in tracking the progress of these amendments, the website of the Ada Conformance Assessment Authority (ACAA) provides ready access to all of the amendments, as well as minutes of ARG meetings. The URL for this website is:

<http://www.ada-auth.org/>

## References

- [1] Wellings, A.J.; Johnson, B.; Sanden, B.; Kienzle, J., Wolf, Th., and Michell, S.: "Integrating Object-Oriented Programming and Protected Objects in Ada 95", ACM TOPLAS 22 (3), May 2000; pp. 506-539.

# Ada-Europe 2003 Sponsors

## **ACT Europe**

Contact: *Zépur Blot*

8 Rue de Milan, F-75009 Paris, France

Tel: +33-1-49-70-67-16

Email: [sales@act-europe.fr](mailto:sales@act-europe.fr)

Fax: +33-1-49-70-05-52

URL: [www.act-europe.fr](http://www.act-europe.fr)

## **Aonix**

Contact: *Marrienne Tournery*

66/68, Avenue Pierre Brossolette, 92247 Malakoff, France

Tel:

Email : [info@aonix.fr](mailto:info@aonix.fr)

Fax: +33-1-41-48-10-20

URL : [www.aonix.fr](http://www.aonix.fr)

## **Green Hills Software Ltd**

Contact: *Christopher Smith*

Dolphin House, St Peter Street, Winchester, Hampshire, SO23 8BW, UK

Tel: +44-1962-829820

Email :

Fax: +44-1962-890300

URL : [www.ghs.com](http://www.ghs.com)

## **Praxis Critical Systems Ltd**

Contact: *Rod Chapman*

20 Manvers Street, Bath, BA1 1PX, UK

Tel: +44-1225-823763

Email : [sparkinfo@praxis-cs.co.uk](mailto:sparkinfo@praxis-cs.co.uk)

Fax: +44-1225-469006

URL : [www.sparkada.com](http://www.sparkada.com)

## **I-Logix**

Contact:

1 Cornbrash Park, Bumpers Way, Chippenham, Wiltshire, SN14 6RA, UK

Tel: +44-1249-467-600

Email : [info\\_euro@ilogix.com](mailto:info_euro@ilogix.com)

Fax: +44-1249-467-610

URL : [www.ilogix.com](http://www.ilogix.com)

## **Rational SARL**

Contact: *Nasser Kettani*

Immeuble de la Gare, 1, Place Charles de Gaulle, 78180 - Montigny le Bretonneux, France

Tel: ++33-1-30-12-09-50

Email: [info@rational.com](mailto:info@rational.com)

Fax: ++33-1-30-12-09-66

URL: [www.rational.com](http://www.rational.com)

## **TNI Europe Limited**

Contact: *Pam Flood*

Triad House, Mountbatten Court, Worrall Street, Congleton, Cheshire CW12 1DT, UK

Tel: +44-1260-29-14-49

Email: [info@tni-europe.com](mailto:info@tni-europe.com)

Fax: +44-1260-29-14-49

URL: [www.tni-europe.com](http://www.tni-europe.com)