

ADA USER JOURNAL

Volume 24
Number 3
September 2003

Contents

	<i>Page</i>
Editorial Policy for <i>Ada User Journal</i>	130
Editorial	131
News	132
Conference Calendar	173
Forthcoming Events	179
Articles	
John Barnes	
“ <i>Additional Numeric Packages for Ada 2005</i> ”	185
Ada-Europe 2003 Sponsors	192
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Unless the authors state explicitly otherwise, submission of an article is taken to imply that it represents original, unpublished work, not under consideration for publication elsewhere.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

The Ada community continues to live its intense 2003. Its *fil rouge* certainly is language revision process, which we have been and will be watching closely, and which makes the yearly Ada conferences extremely important moments for the revision process to touch base with its ultimate users.

The Ada User Journal takes great pleasure in being a vehicle for the dissemination of highlights from the language revision process. We plan to continue this coverage throughout the whole of next year, especially reporting on the consolidation of the various proposals under consideration.

In this issue of the Ada User Journal our technical article is offered by John Barnes, a distinguished member of the ARG and long acquaintance of this journal, who presents a number of packages for addition to the Numerics Annex of Ada 2005.

A rich News section, which continues to offer very rich information on the lively and healthy activity of the Ada community at large, constitutes the other major part of this issue, along with the usual Calendar of events and announcements.

We hope you will enjoy this issue and look forward to the future.

Tullio Vardanega
Padova
September 2003
Email: tullio.vardanega@math.unipd.it

News

Dirk Craeynest (ed)

Offis nv/sa and K.U.Leuven. Email Dirk.Craeynest@cs.kuleuven.be

Contents

	page
Ada-related Events	132
Ada and Education	134
Ada-related Resources	135
Ada-related Tools	137
Ada-related Products	146
Ada and Linux	150
Ada and Microsoft	150
References to Publications	151
Ada Inside	153
Ada in Context	156

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]

May 8 - DC/Baltimore SIGAda Meeting on Optical Character Recognition (OCR) of Cryptographic Source Code

From: David Emery <demery@cox.net>
Date: Fri, 02 May 2003 22:51:06
Subject: Next MD/DC Local SIGAda meeting (McLean, VA, 8 May)
Newsgroups: comp.lang.ada

Next Meeting: Thursday, 8 May 2003, 7:30 P.M. [...] 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" by Karl Nyberg.

Abstract:

This talk 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. [...]

Slides From John Warther's and Gordon Uchenick's Presentation Available:

At the Baltimore SIGAda meeting on 8 April 2003, John Warther and Gordon Uchenick gave an excellent presentation titled: High Reliability, Safety, and Secure Solutions for Embedded Developers. [See also "Apr 8" topic in AUJ 24.2 (Jun 2003), p.68. -- dc] Slides from his presentation are available online as a Powerpoint Presentation at http://www.acm.org/sigada/locals/dc/2003_04_Gordon_Uchenick.ppt (2.097 MB). The much requested jpg image of the X-38 Space Station Lifeboat with caption "When Ctrl-Alt-Delete is Not an Option" is available at: <http://www.acm.org/sigada/locals/dc/X38.jpg> (159 KB). [...]

Jeff Castellow, Chair, DC SIGAda

May 20 - Aonix Technology Seminar

From: Aonix <info@aonix.fr>
Date: Fri, 09 May 2003 20:14:42
Subject: [ada-france] Seminaire technique gratuit
To: ada-france@ada-france.org

[Translated from French: -- dc]

Free Technology Seminar

Tuesday May 20, 2003, Hôtel Lutetia, [...], Paris, France

Aonix, major actor in the field of innovating technologies adapted to complex and strategic industrial applications, proposes you to give a progress report on a number of promising solutions for the development of advanced software, in particular for systems known as "critical" or "certifiable". [...]

Program (in French unless mentioned otherwise)

9.00-9.30 Introduction - Jacques Brygier, Directeur Marketing International

Session Modeling and MDA (Model Driven Architecture)

9.30-10.00 Modeling techniques* - Michael Benkel, Chef de Produits, Aonix. News from the OMG, UML 2.0, MDA, the real-time profile(s), automatic generation of code, ...

10.00-10.30 An example of the use of MDA techniques for a "mission critical"

application - Jean-Noël Meunier, Expert Technologies Objet, Aonix. The European project Hidoors, the definition of a real-time UML profile, the ACD technology, ...

Session Objects and Critical Real-Time 10.45-11.00 Deployment of object technologies in critical software - Marc Richard-Foy, Responsable Projets Exploratoires, Aonix. [...]

11.00-11.45 UML and reliability - Gilles Motet, Professeur à l'INSA Toulouse. History, current work, UML and certification, the ISO High Integrity working group, ...

11.45-12.15 Java for embedded/real-time with strong constraints* - Kelvin Nilsen, Ph.D, Founder of Newmonics. [...]

Session Test and Certification

13.30-14.00 Key stages in the implementation of test phases - Pierre Morere, Responsable Ada and Reliability Center, Aonix. [...]

14.00-14.30 From code audit to test audit - Romuald Reculet, Consultant Aonix. [...]

14.30-14.45 An example of a test session - Patricia Langle, Consultant Avant-Vente, Aonix. The use of a commercial tool adaptable to a predefined test process, ...

15.00-15.30 Radioscopy of a certification - Pierre Morere, Responsable Ada and Reliability Center, Aonix.

15.30-16.30 Customer experience feedback - Auxitrol, Thales, ...

* Presentations in English.

Aonix, 66-68, av. Pierre Brossolette, 92247 Malakoff Cedex, info@aonix.fr, www.aonix.com

Jun 16-20 - Ada-Europe 2003 Conference

From: dirk@ada.cs.kuleuven.ac.be (Dirk Craeynest)

Date: 18 May 2003 21:05:21

Organization: Ada-Europe, c/o Dept. of Computer Science, K.U.Leuven

Subject: Ada-Europe'2003 early registration deadline approaching

Newsgroups:

comp.lang.ada,fr.comp.lang.ada

2nd Call for Participation - Updated 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.

4th invited speaker announced: Patrick Farail speaks on the software development process for Airbus A380.

Opening event on Monday evening being prepared.

[For more details see also same topic in AUJ 24.2 (Jun 2003), p.70. -- dc]

Papers: 3 invited papers and 29 technical 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 15 countries: Australia, Austria, Belgium, Canada, Finland, France, Germany, Hong-Kong, India, Israel, Italy, Portugal, Spain, United Kingdom, and USA.

Exhibition: 10 exhibitors [...]: 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. [...]

From: dirk@ada.cs.kuleuven.ac.be (Dirk Craeynest)

Date: 5 Jun 2003 22:12:24 +0200

Organization: Ada-Europe, c/o Dept. of Computer Science, K.U.Leuven

Subject: Press Release - Reliable Software Technologies, Ada-Europe'2003

Newsgroups:

comp.lang.ada.fr.comp.lang.ada

[...] Latest updates:

The opening session on Monday evening will see the European premiere of a new film documentary on the life and work of Ada Byron. More info below and via the conference web site.

The proceedings published by Springer are ready and will be distributed at the conference. More info is available at http://www.springer.de/cgi/svcat/search_book.pl?isbn=3-540-40376-0. Abstracts can be checked out via the conference web site.

[...] Opening session:

The organizing committee of the Ada-Europe 2003 conference in Toulouse is preparing an attractive opening session on Monday evening, June 16, 2003. We gladly acknowledge the generous sponsoring of this session by ACT Europe <http://www.act-europe.fr/>.

A documentary film was made recently on Ada Byron Lovelace, her work with Charles Babbage, and their contributions to computing. The film is entitled "To Dream Tomorrow" and has received very good critiques at screenings in the U.S.

The formal premiere will be in Washington DC this summer; the screening at our opening session is its European avant-premiere.

More info is available at the conference web site <http://www.ada-europe.org/conference2003.html>, see "What's New?". [...]

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

Sep 15-19 - 12th International Real-Time Ada Workshop

From: Clyde Roby <roby@ida.org>

Date: Fri, 1 Aug 2003 07:49:02

Subject: SIGAda and Conferences reminder

To: SIGAda-announce@acm.org

Don't forget about our upcoming Ada-related conferences: [...]

International Real-Time Ada Workshop. Pousada Monte de Sta. Luzia, Viana do Castelo, Portugal. 15-19 September 2003. <http://www.hurray.isep.ipp.pt/irtaw2003/>

[...] Clyde Roby, SIGAda Secretary

Sep 29 - Oct 2 - Ada-Germany 2003 Conference at GI Conference

From: keller <keller@iai.fzk.de>

Date: Mon, 23 Jun 2003 17:23:13

Subject: Ada Deutschland Tagung 2003 auf der GI Tagung im Rahmen der Teiltagung Sicherheit

To: Ada, Liste <ada-list@ada-deutschland.de>

[Extracts translated from German. -- dc]

For Info.

Track on "Security - Protection and Reliability" at the GI Annual Conference Informatik 2003: 29 September - 2 October in Frankfurt/Main.

The two communities "Safety" and "Security" have united in a new GI-Fachbereich under the name "Security - Protection and Reliability": Safety, the protection of humans and goods from failures of technical systems, and Security as protection of information and data processing against intelligent aggressors. The amount of sensitive applications, for instance the use of computers in vehicles and airplanes, wireless communication or electronic payments, as well as making systems accessible that were at first dedicated and isolated and the discussion around appropriate interfaces show the strong synergy between these topics, which are however still to a large extent researched independently despite general efforts.

In order to clarify these topics, to further develop them, and to bring experts and interested people together both from the scientific and the application world, the

Fachbereich organizes a track on "Security - Protection and Reliability" at the GI Annual Conference "Informatik 2003" from 29/9 to 2/10/2003 at the Johann Wolfgang Goethe university in Frankfurt/Main.

The track will cover the entire range of Security and Safety topics with reviewed papers and workshops, [...] Tutorials on data replication, Ada dataflow analysis, methods for the development of safe systems, software architecture evaluation and security in the Internet environment complete the program.

All further information, among others the detailed program, is on the web pages of the Security Fachbereich (www.gi-fb-sicherheit.de) and of "Informatik 2003" (www.informatik2003.de). [...]

Dr. Hubert B. Keller, Forschungszentrum Karlsruhe in der Helmholtz-Gemeinschaft, Institut für Angewandte Informatik, Postfach 3640, 76021 Karlsruhe

From: keller <keller@iai.fzk.de>

Date: Tue, 24 Jun 2003 09:42:27 +0200

Subject: Re: Ada Byron Lovelace: To

Dream Tomorrow

cc: ada-list@ada-deutschland.de

christoph.grein@eurocopter.com wrote:

> I saw this film ["To Dream Tomorrow" -- dc] at Ada-Europe 2003 in Toulouse. It is really extremely worth seeing. Length just under 1 hour. The film is only about Ada's life, not a word about Ada as a language (that was also not the goal of the film). What would this women have reached, if she would have been allowed to study? (Women were at that time not allowed at universities. It was believed that mathematics harmed their brains!)

Ada-Germany has bought this film and will show it gladly at the 2003 Ada conference. The Director of the film, a Swiss, also can respond the German comments.

From: keller <keller@iai.fzk.de>

Date: Mon, 04 Aug 2003 09:17:14 +0200

Subject: [Ada] Ada auf der GI-Jahrestagung 2003 und Ada Europe 2004

To: ada@mail.gi-fb-sicherheit.de

The up-to-date overview of the GI annual conference 2003 with the sessions, Workshops and Tutorials of the specialized Ada group can be found under <http://www.gi-fb-sicherheit.de/vg/informatik2003/index.html>

[...] The new "Ada Tour CD 2004" is in preparation. It will be available mid September. [...]

Ada mailing list

Ada@mail.gi-fb-sicherheit.de

<http://mail.gi-fb-sicherheit.de/mailman/listinfo/ada>

Dec 7-11 - ACM SIGAda 2003 Conference

From: ricky.sward@ix.netcom.com (Ricky E. Sward)
Date: 12 Jun 2003 21:43:35 -0700
Subject: SIGAda 2003 Submission Deadline Extended
Newsgroups: comp.lang.ada

Submission Deadline Extended to July 11, 2003!

Call for Participation - SIGAda 2003, 7-11 December 2003, San Diego, California, USA. Sponsored by ACM SIGAda (Approval pending by ACM).

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 or the Program Chair, Ricky E. Sward rsward@usafa.edu.

Jun 14-18 - Ada-Europe 2004 Conference

From: dirk@ada.cs.kuleuven.ac.be (Dirk Craeynest)
Date: 24 Jun 2003 22:02:19
Organization: Ada-Europe, c/o Dept. of Computer Science, K.U.Leuven
Subject: CfPapers, Reliable Software Technologies, Ada-Europe 2004
Newsgroups: comp.lang.ada.fr.comp.lang.ada

9th International Conference on Reliable Software Technologies - Ada-Europe 2004, 14 - 18 June 2004, Palma de Mallorca, Spain

<http://www.ada-europe.org/conference2004.html>

The 9th International Conference on Reliable Software Technologies

(Ada-Europe 2004) will take place in 2004 in Palma de Mallorca, Spain. The full conference will comprise a three-day technical program and exhibition from Tuesday to Thursday, and parallel workshops and tutorials on Monday and Friday.

The conference will provide an international forum for researchers, developers and users of reliable software technologies. Presentations and discussions will cover applied and theoretical work currently conducted to support the development and maintenance of software systems. Participants will include practitioners and researchers from industry, academia and government. [...]

The proceedings will be published in the Lecture Notes in Computer Science (LNCS) series by Springer Verlag, and will be available at the start of the conference. [...]

There will be honorary awards for the best paper and the best presentation. See the Ada-Europe Prizes page for previous winners:

<http://www.ada-europe.org/prizes.html>.

The conference will be accompanied by a three-day commercial exhibition on June 15, 16 and 17. Vendors of software products and services should contact the Exhibition Chair, Peter Dencker (dencker@aonix.de) at their earliest convenience for further information and to ensure their inclusion.

For full contact information and the Program Committee: see the online CfP at the conference Web site. [...]

Dirk.Craeynest@cs.kuleuven.ac.be, Ada-Europe2004 Publicity Chair

Ada and Education

Ada in the Computing Curriculum

From: Chad R. Meiners
<crmeiners@hotmail.com>
Date: Sat, 3 May 2003 12:22:19
Organization: Michigan State University
Subject: Re: employment with ada
Newsgroups: comp.lang.ada

[About a claim made at a conference: -- dc]

> No college in this country [= USA] is teaching Ada. There may be some qualifiers on that that I don't remember, such as "as a major portion of their program", or something like that, but in short there isn't a source of new Ada programmers, nor is there likely to be.

Now that just isn't true! Truman State University's (www.truman.edu) computer science program is taught almost entirely in Ada. In fact Boeing sends representatives to the university to give

job hiring presentations outside of the normal career expo.

[See also the "Ada and Education" news section in almost every previous Ada User Journal issue. -- dc]

From: Jeffrey Carter <jrcater@acm.org>
Date: Sat, 03 May 2003 20:30:48
Subject: Re: employment with ada
Newsgroups: comp.lang.ada

> Glad to hear it. Any others? Someone should tell [the people making the above claim -- dc].

<http://www.seas.gwu.edu/~mfeldman/ada-foundation.html> lists 117 "Colleges and Universities Introducing Ada as the First Language Taught in a Computing Curriculum". 56 are not in the US, leaving "only" 61 in the US. 33 more (25 in the US) introduce Ada in the 2nd or 3rd course in the curriculum.

Given this level of FUD on their 1st point, I have to suspect that they were rationalizing their prejudices, not examining the evidence and reaching a valid conclusion.

From: Alex Gibson <alxx@ihug.com.au>
Date: Sun, 4 May 2003 23:45:33
Organization: Ihug Limited
Subject: Re: employment with ada
Newsgroups: comp.lang.ada

Okay I'm not in the US but the university I'm at still teaches Ada. Mostly to graduate students.

http://www.eng.uts.edu.au/CurrentStudents/Postgraduate/postgraduate_subject_descriptions/49212.html and [49234.html](http://www.uts.edu.au/div/publications/subjects/49234.html)

Ada Education in France

From: Tarroux
<philippe.tarroux@limsi.fr>
Date: Sat, 10 May 2003 11:10:07 +0200
Subject: Re: [ada-france] enseignement de Ada
To: <ada-france@ada-france.org>

[Extracts translated from French. See also same topic in AUJ 24.2 (Jun 2003), p.71. -- dc]

I've taught Ada in various circumstances (at the ENS Ulm in particular) and this year I tried the experiment to set up an introduction to informatics with Ada and Java for the non-informatics students of the "DEA de Sciences Cognitives" of Paris-Sud.

The experiment is instructive concerning the "lightness" of Java and the rigour of Ada. Certain students have thus discovered Ada "by contact" and I had very positive feedback in particular on the way concurrency is managed.

Since a few years, we give training courses where Ada is the supporting

language and the theses in progress in my group relate to developments in Ada.

Philippe Tarroux, Responsable du Département Communication Homme Machine, LIMSI-CNRS, BP 133, F-91403 Orsay cedex

From: Daniel Feneuille <feneuille@univ-aix.fr>
Date: Mon, 26 May 2003 15:42:39 +0200
Organization: IUT Aix
Subject: [ada-france] enseignement de Ada (suite!)
To: Ada France <ada-france@ada-france.org>

Some time ago I proposed to write a paper reviewing ten years of Ada teaching. Many answers incited me to persevere. But progressively (and taking into account the observations received) I think that it would be wise to make a more collective work of it! [...]

My experience relates to teaching Ada to "D.U.T." i.e. students receiving 900 hours of information after the "Bac" (over two years) with a concern of being directly integrable in industry. [...]

I thus wish to ask for participation either in the form of writing specific "chapters" or in the form of criticisms and complements to parts already written. That risks to be a little long but I think that it would be more interesting and can be more exhaustive!

I await on this list your remarks [...].

From: Etienne Baudin <pfox@free.fr>
Date: Fri, 13 Jun 2003 11:53:54
Subject: Etablissements Toulousains
Newsgroups: fr.comp.lang.ada

While speaking about the INSA of Toulouse, I believe that there exists somewhere [...] a list of the establishments where Ada courses are given. In my memory there was no Toulouse establishment in the list.

Thus if the persons concerned with the updates reads this, they can add there: University Paul Sabatier (Toulouse) in DEUG MIAS [DEUG = "Diplôme d'Etudes Universitaires Générales", MIAS = Mathematics, Informatics and Applied Science -- dc]; INSA of Toulouse in the first cycle and in the third year at the department of electric and informatics engineering.

From: Lionel Draghi <fr.thalesgroup.com>
Date: Fri, 13 Jun 2003 12:15:45
Subject: Re: Etablissements Toulousains
Newsgroups: fr.comp.lang.ada

Such a list exists at <http://www.ada-france.org>. See the section "Communauté".

But no-one in particular is responsible for the updates. It is a collaborative site on which everyone is invited to submit short messages and articles. Similar for descriptions of the establishments using

Ada. They must be written by those who know, you for example.

Such an article is easy and rapidly written, and there are several examples that have been published recently. [...]

From: valdesl@wanadoo.fr (Laurent Valdes)
Date: Fri, 20 Jun 2003 19:10:10
Organization: ARUR
Subject: Re: Etablissements Toulousains
Newsgroups: fr.comp.lang.ada
 > University Paul Sabatier (Toulouse) in DEUG MIAS

Yes, I followed the courses of this DEUG. They took Ada instead of Pascal for teaching imperative programming with invariants and program proofs. I must say that I was very impressed by the compiler. Especially when I returned to C after I understood what Ada gave me. [...]

From: Sébastien Martin <Sebastien.Martin2@free.fr>
Date: Sat, 21 Jun 2003 15:51:21 +0200
Subject: Re: Etablissements Toulousains
Newsgroups: fr.comp.lang.ada

> Moreover I believe that Ada is also taught at Orsay in Paris.

Ada is taught in the first year of Fiifo (Engineers at the Faculty of Orsay) where I am a student; [...]

Tutorials on Distributed Programming in Ada

From: Samuel Tardieu <sam@rfc1149.net>
Date: Wed, 23 Jul 2003 23:56:01
Subject: Re: Distributed Programming Tutorials?
Newsgroups: comp.lang.ada

> Is there any online tutorial available, that covers distributed (and reliable) programming in Ada95?

See <http://tinyurl.com/huj6>, there is a small tutorial in it.

[This URL is for an on-line version of the User Guide for GLADE, the GNAT implementation of the Ada95 Distributed Systems Annex. See also "Getting Started with Distributed Ada" in AUJ 23.4 (Dec 2002), p.191. -- dc]

From: Bobby D. Bryant <bdbryant@mail.utexas.edu>
Date: Wed, 23 Jul 2003 22:33:22 -0600
Subject: Re: Distributed Programming Tutorials?
Newsgroups: comp.lang.ada

If you're a GNAT user and install the GLADE implementation of the DSA (aka gnat-glade, due to a name clash with another GLADE), you should get a handful of example programs that will be helpful in getting started, plus some PostScript documentation.

As far as "reliable" goes, there are some non-trivial theoretical issues you can get in to if you want to take the plunge. I'm

sure there are lots of different approaches, but you can find J. Misra's work at <http://www.cs.utexas.edu/users/psp/>

Bobby Bryant, Austin, Texas

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: Jean-François Peyre <peyre@cnam.fr>
Date: Wed, 09 Jul 2003 16:24:10
Organization: Cedric, CNAM
Subject: [ada-france] Re: formation ada, programmation
To: ada-france@ada-france.org

> Due to my professional transfer in September, I'm looking for training in the area of Bordeaux on the Ada language or on programming in general [...]

At the CNAM of course!

You will find various programming courses (some in Ada) in the center associated with the CNAM of the Aquitaine region. Here are their coordinates: CNAM Aquitaine, 16 Cours de la Marne, 33000 Bordeaux, Tél: 05 57 57 30 50, Fax: 05 57 57 30 70.

And also www.cnam.fr or deptinfo.cnam.fr where you will find many programming resources (especially in Ada).

Jean-François Peyre, Conservatoire National des Arts et Métiers, Paris 03

From: colbert@abssw.com (Ed Colbert)
Date: 27 Jul 2003 09:57:16 -0700
Subject: [Announcing] Public Ada 95 Course, 18-22 August 2003 in Carlsbad CA
Newsgroups: comp.lang.ada

Absolute Software will be holding a public Ada 95 course during the week of 18 August 2003 in Carlsbad, CA. You can find a full description and registration form on our web-site, www.abssw.com. Click the Public Courses button in the left margin. (We also offer courses on object-oriented methods and other object-oriented languages.) [...]

Ada-related Resources

French Resources

From: Brian Fraval <brian@fraval.org>
Date: Wed, 7 May 2003 23:59:31
Subject: Re: site web pour Ada?
Newsgroups: fr.comp.lang.ada
 [Translated from French: -- dc]

The FAQ of this newsgroup: <http://www.rfc1149.net/fcla/>

You will find some links on this page:
<http://brian.fraval.org/cnam/ada.html>

Ada-Germany News Items

From: keller <keller@iai.fzk.de>
Date: Wed, 14 May 2003 08:57:42 +0200
Subject: Ada Liste
To: Ada, Liste <ada-list@ada-deutschland.de>

[Some news items from the Ada-Germany web site; more information on selected items elsewhere in this AUJ issue. Extracts translated from German: -- dc]

Ada on the Web:
<http://www.ada-deutschland.de/>

News

Rod Chapman, Praxis Critical Systems: "MISRA C at SIL4? Perspectives and Alternatives".
 <<http://www.sparkada.com/downloads/misracatsil4reader.pdf>>.

Article on UML and Ada code generation. Hubert Keller et al.: "Customizing UML for the development of distributed reactive systems and code generation to Ada 95".

<http://www.ada-deutschland.de/papers/inject_paper.pdf>. [...]

GI Jahrestagung 2003 / Call For Papers. Informatik 2003, Teiltagung Sicherheit <<http://www.gi-fb-sicherheit.de/vg/informatik2003/cfp.html>> Frankfurt, 29 Sep to 2 Oct 2003. [...]

Ada Article in Softwaretechnik-Trends, Nov 2002. Christoph Grein: "Physikalische Dimensionen in Ada". [...]

Ada and Safety. In several international standards for the development of safety-related systems (e.g. IEC 61508, EN 50128 and DO-178B) Ada is proposed explicitly as "highly recommended"!

GI Jahrestagung 2003. The Security and Safety track
 <<http://www.informatik2003.de/teiltagung/teiltagung.htm>> will be an important part of the GI Jahrestagung 2003 in Frankfurt

<<http://www.informatik2003.de/>> (29 Sep to 2 Oct 2003). The Ada "Fachgruppe" will co-organize this track. [...]

The GI Fachbereich "Security - Protection and Reliability"
 <<http://www.gi-fb-sicherheit.de/>> was created with cooperation of Ada Germany (the Ada group in the Security "Fachbereich").
 <<http://www.gi-ev.de/fachbereiche/sicherheit/ada/index.html>> [...]

The proceedings of the Ada Deutschland Conference 2002 are available. Jürgen F. H. Winkler, Peter Dencker, Hubert B. Keller, Michael Tonndorf (Eds.)

New Interviews on Ada-France Web Site

From: Samuel Tardieu <sam@ada-france.org>
Date: Mon, 19 May 2003 13:52:06 +0200
Organization: Ada addicts (see <http://www.ada-france.org/>)
Subject: [ada-france] Entretien avec Gautier de Montmollin
To: ada-france@ada-france.org

[Translated from French: -- dc]

After Tristan Gingold, it's Gautier de Montmollin who granted Lionel Draghi an interview for the Ada-France site <<http://www.ada-france.org/>> [Gautier speaks about TeXCAD; see also "TeXCAD - Drawing and Retouching Pictures in LaTeX" in the Ada Tools news section. -- dc].

From: Lionel Draghi
 <lionel.draghi@free.fr>
Date: Thu, 05 Jun 2003 02:03:09
Subject: [ada-france] Troisième interview sur www.ada-france.Eorg
To: ada-france@ada-france.org

After Gautier de Montmollin two weeks ago, it's now Pascal Obry who speaks about AWS. [...] Don't hesitate to propose us topics for other interviews.

From: Lionel Draghi
 <Lionel.Draghi@Ada-France.org>
Date: Thu, 05 Jun 2003 02:18:49
Subject: interview de Pascal Obry sur www.Ada-France.org
Newsgroups: fr.comp.lang.ada

The third interview published on <http://www.ada-france.org/>, that of Pascal Obry on AWS, is for me the occasion to recall that it's useful to regularly take a look at <http://www.ada-france.org/>.

Things happen all the time in the Ada world. To realize this, it's enough to look at the publication dates of the articles and announcements.

And to finish, a reminder: everyone can propose articles, as long as the subject is related to Ada, of course.

Updated Ada Conformity Assessment Test Suite

From: abandon@sover.net
Date: Wed, 28 May 2003 10:30:44 -0400 (EDT)
Subject: [AdaIC] Ada News Updates: Conventions, revised book
To: announce@adaic.com

[...] Finally, we've posted the Ada Conformity Assessment Test Suite (ACATS) Modification List 2.5F and associated test files. Please visit <http://www.adaic.org/> for more information [...]

Ann Brandon, AdaIC Editorial Webmaster, ann@onyons.com

AIs for Ada Extensions

From: Robert I. Eachus
 <rieachus@attbi.com>
Date: Wed, 11 Jun 2003 06:31:07
Subject: AIs for Ada extensions
Newsgroups: comp.lang.ada

[On how to get language extensions and new packages included in the Ada standard. -- dc]

The right thing to do here is to get busy and catch up on the proposed new packages in Ada0Y. They can be found in the AIs at <http://www.ada-auth.org/ais.html>. I made up a list of the extension AI's I am aware of -- Bob and Randy should let me know if I missed any. Of course, in many cases whether an AI is a "normal" AI fixing something that is broken or a suggested extension is hard to characterize. Also some of the AI's listed as "other than packages" may end up as extension packages and vice-versa. AI-234 for example may end up being resolved by having an explicit package that defines a "true" unsigned integer type. (Or such a type may end up in package System, System.Address often is such a type.)

Some things to keep in mind if you want to read these AIs and comment.

- 1) Not all of these AIs will end up in Ada0Y or whatever you want to call it. AI-323 is a good example of something that won't make it.
- 2) Read the AIs thoroughly before sending comments to the ARG. If you are not sure whether to post here or to the ARG or the Ada-Comment list, post here. There are several ARG members who read this list regularly. If your post should go to the ARG, we will either suggest you send it, or forward it.
- 3) There will be a meeting of the ARG in just over a week. Fifteen extension AIs are on the preliminary agenda. So some of these AIs will change drastically in a couple of weeks.
- 4) If you do have constructive comments, be very specific. The more detail in your comments, the more they will be listened to. I think there should be a package for ZZZ is not useful. I think this package (see listing below with explanation) should be made part of the standard, is more likely to be listened to -- and result in lots of other comments.
- 5) If you want to submit a new package proposal, go ahead. But have a package you use for a rough proposal at least. "Wouldn't it be nice to have..." doesn't belong in the standard. (If you do have a package that your group and several other groups uses, those are good candidates for inclusion.)

If you have thin skin or a big ego think carefully before sticking your head up. People don't get attacked at ARG

meetings or on the ARG list, but ideas get sliced into shreds -- often by the original author. (Think of submitting ideas to the ARG as akin to throwing a chunk of meat to a well-trained sled dog team. The dogs may be nice to each other, but the meat is going to get torn to shreds and thrown around a lot.) Eventually, of course, all of the potential issues have been dealt with, and the final proposal may look nothing like the original. At that point the ARG will move on to the next AI.

This is necessary. You don't want half-baked or changes that were not thoroughly examined added to the language. But it also means that many people are best off submitting ideas to the ARG through filters (like this newsgroup). If half a dozen people agree on a particular package specification, it might not get torn up too badly by the ARG. But more important, if you are not the only author, you won't take the criticism so personally.

So if you want to participate read the e-mail comments on some of these AIs. You may then want to arrange to be invited to an ARG meeting. If you survive the meeting, you may even want to join. Feeling mentally wrung out at the end of a meeting is nothing to worry about. It means you were actually following the discussions. Some of these AIs take more than a day to fully understand, and the ARG often deals with more than a dozen per day. Of course, some AIs go through several meetings before they are finished. Some issues are best dealt with in a meeting, others are better handled by e-mail. So a typical resolution is for the ARG to accept a set of revisions "in principle," delegate revising the AI to someone. This will be discussed by e-mail and then either an e-mail vote is taken, or it goes on the agenda for the next meeting (again).

Proposed new packages:

AI-248 Directory Operations.

AI-282 Ada unit information symbols.

AI-292 Sockets Operations (has a general discussion about standardizing new packages).

AI-296 Vector and matrix operations.

AI-297 Timing Events.

AI-301 Missing operations in Ada.Strings.Unbounded.

AI-302 Data structure components for Ada.

AI-307 Execution-Time Clocks.

AI-324 Physical Units Checking.

Language Extensions for Ada0Y other than packages:

AI-217 Handling Mutually Dependent Type Definitions that Cross Packages (see also AI-10217, AI-20217, AI-30217, AI-40217 and AI 50217).

AI-218 Accidental overloading when overriding (see also AI-10218).

AI-222 Feature control.

AI-224 pragma Unsuppress.

AI-230 Generalized use of anonymous access types.

AI-231 Access-to-constant parameters and null-excluding subtypes.

AI-234 Unsigned integer types.

AI-249 Ravenscar profile for high-integrity systems.

AI-250 Protected Types, Extensible, Tagged, Abstract.

AI-251 Abstract Interfaces to provide Multiple Inheritance.

AI-252 Object.Operation Notation.

AI-254 Downward closures for access to subprogram types.

AI-257 Restrictions for implementation-defined entities.

AI-260 How to control the tag representation in a stream.

AI-261 Extending enumeration types.

AI-262 Access to private units in the private part.

AI-264 Exceptions as Types.

AI-265 Partition Elaboration Policy for High-Integrity Systems.

AI-266 Task Termination procedure (See also AI-10266).

AI-267 Fast float-to-integer conversions.

AI-270 Stream item size control.

AI-274 Requiring complete record representation clauses.

AI-281 Representation of enumeration type image attribute.

AI-284 Nonreserved keywords.

AI-285 Support for 16-bit and 32-bit characters.

AI-286 Assert pragma.

AI-287 Limited Aggregates Allowed.

AI-288 Pre/Postconditions and Invariants.

AI-290 Declaring functions Pure.

AI-294 Built-in hash function.

AI-298 Non-Preemptive Dispatching.

AI-299 Defaults for generic formal parameters.

AI-300 The standard storage pool.

AI-305 Data structure components for Ada.

AI-310 Ignore abstract nondispatching subprograms during overloading.

AI-314 Standardize Discard_Names.

AI-315 Full support for IEEE 754.

AI-317 Partial Parameter Lists for Formal Packages.

AI-318 Object_Size attribute.

AI-322 User-defined operator symbols.

AI-323 Allow in out parameters for functions (status No Action, and that is highly unlikely to change).

AI-325 Anonymous access types as function result types (depends on AI-230).

AI-326 Tagged incomplete types.

AI-327 Dynamic Ceiling Priorities.

More on Real-Time AIs

From: Rodrigo_Garcia

<rodrigo.Garcia@epfl.ch>

Date: Tue, 24 Jun 2003 15:58:48

Organization: EPFL

Subject: Re: Task execution time (writing a scheduler)

Newsgroups: comp.lang.ada

Last minute information about this and other proposals (from Alan Burns):

"There was a WG9 and an ARG meeting last week as part of Ada Europe. I thought it might be interesting to you all to get an update on the progress of real-time AIs.

Ravenscar AIs (249 and 305) - have already passed through WG9.

Partition Elaboration Policy (265) now agreed by WG9.

Execution Timers (budget timing) (307) agreed by ARG, under editorial review - will pass to WG9.

Non-Preemptive Dispatching (298) agreed by ARG, under editorial review - will pass to WG9.

Rewording of D.2 (321) agreed by ARG, under editorial review - will pass to WG9.

Timing Event (297) agreed by ARG, under editorial review - will pass to WG9.

Task Termination Procedure (266) still alive but best way to meet varying demand still not certain.

Dynamic Ceiling (327), well received, work to be done but looks likely to succeed.

All in all very good progress is being made - lets get more together at the forthcoming workshop in September.

Professor Alan Burns, Head of Department Department of Computer Science, University of York, Heslington York, UK YO10 5DD"

Ada-related Tools

Reusable Software from CNES's "Nacelles Pointées" Division

From: Laurens Andre

<Andre.Laurens@cnes.fr>

Date: Wed, 14 May 2003 17:09:26 +0200

Subject: [ada-france] réutilisables

To: 'Ada-France, Liste' <ada-france@ada-france.org>

[Translated from French. -- dc]

A number of Ada packages and software are made available on the web-site of the "Nacelles Pointées" ["Stabilized Gondolas" -- dc] (an activity of the "Ballons" division of the CNES - cf URL below) under the heading "Les gens, les écrits -> Logiciels". This section will be updated as developments are realised.

The software is either directly relevant for the "Nacelles Pointées" application domain or for general fields used by this activity, but also generally useful programs, reusable in very different contexts. In general, it is portable software, tested under Linux and Windows. [...]

André Laurens, Nacelles Pointées, Centre National d'Etudes Spatiales, 31401 Toulouse,
http://ballons.cnes.fr/nacelles_pointees

Ada0Y.Directories - AI-248 Implementation

From: Martin Dowie
 <martin.dowie@bopenworld.com>
 Date: Sun, 1 Jun 2003 11:02:52
 Subject: [Announcement]
 Ada0Y.Directories - alpha release
 Newsgroups: comp.lang.ada

Under the premise of 'release early, release often' I'm announcing an alpha release of "Ada0Y.Directories". This is an implementation of AI-248 (<http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-00248.TXT>). Also included are AdaBrowse generated html pages containing the AI text as comments to each subprogram and some test programs.

It isn't 100% complete but it can currently: set/get current directory, create/destroy directories, create paths, search for files/directories (including wildcards).

The URL is:
<http://www.martin.dowie.btinternet.co.uk/>

It has been implemented and tested on WinNT/WinXP using GNAT 3.15p.

If you can confirm other configurations please send me details via the 'feedback' links and I'll include a table. If you find any bugs please also send 'feedback'. [See also "Towards a Standard Ada OS Binding" in AUJ 24.1 (Mar 2003), p.13 and "OS File System Bindings" in AUJ 23.1 (Mar 2002), pp.13-14. -- dc]

From: Martin Dowie
 <martin.dowie@bopenworld.com>
 Date: Fri, 13 Jun 2003 16:50:08
 Subject: Ada0Y.Directories - updated
 Newsgroups: comp.lang.ada

I've uploaded a new version of Ada0Y.Directories at:
<http://www.martin.dowie.btinternet.co.uk/>

New features in this version are: copy and rename files; added support for Simple, Base and Full Names; added support for file extensions; added support for modification time*; simple directory name simplification; added exception messages; updated the tests; added more info from the AI as comments to the package spec; added more AdaBrowse tags; added "pragma License" information; created HTML using GNATHTML. (* On Win32 this is and not the value seen in Explorer.)

Still not 100% compliant but probably good enough for applications.

Thanks to those who have already sent feedback, the more the better! I'm particularly interested to hear from people using something other than WinNT/WinXP.

Charles - Container Library

From: mheaney@on2.com (Matthew Heaney)
 Date: 12 Jun 2003 10:33:53 -0700
 Subject: Re: C bug of the day
 Newsgroups: comp.lang.ada

> Ada has no equivalent to the C++ STL. Ada is fine as long as you are happy to program using only arrays. Beyond that, you are stuck searching the web for some public domain components and hoping they have been debugged enough to be reliable, or writing your own and hoping that you don't somehow screw up new/unchecked deallocation.

Yes, it's true that there is no standard container and algorithms library a la the STL in Ada95. However, the Charles library is more or less a port of the STL to Ada95, so in practice you don't need to resort to low-level arrays or linked lists.

I also think that Charles has "been debugged enough to be reliable," but ultimately it's up to the user to decide whether that's really the case.

The Charles library is available at my webpage.
<http://home.earthlink.net/~matthewjheaney/charles/>

[See also same topic in AUJ 24.2 (Jun 2003), p.75 and AUJ 23.4 (Dec 2002), p.192. From another message: -- dc]

The statement that "STL is only possible in C++" is false. The Charles library constitutes a proof-by-counterexample. [...] The components in the Charles library use the GMGPL.

From: mheaney@on2.com (Matthew Heaney)
 Date: 27 Jun 2003 17:54:45 -0700
 Subject: Re: Dynamic Ada task creation?
 Newsgroups: comp.lang.ada

> Yes, most container libraries refuse to work with limited data objects, but not all.

The new release of the Charles library does indeed work with element types that are limited. The basic idea is that when you insert an element into the container, the element is allocated with its default value -- whatever that means for the element type. For example, for an element type that is controlled, controlled initialization will occur. [...]

Not all of the containers for limited elements are there yet -- I still have to do the multisets and multimaps. I'll post them next week (30 June 2003).

[Files at URL above:] index.html & charles-20030627.zip

I also posted the slides from my paper presentation about Charles at the Ada-Europe 2003 conference which took place last week in Toulouse, France.

[Files at URL above:] charles.ppt, charlesppt.pdf, charlesppt.htm

From: mheaney@on2.com (Matthew Heaney)

Date: 11 Jul 2003 14:58:04 -0700
 Subject: Re: Computer Language Shootout
 Newsgroups: comp.lang.ada

[...] The latest release (20030710) of Charles has better hash table support. There are also containers for storing elements whose type is limited. [...]

I must say, nothing beats having a predefined container library. Problems that would otherwise take hours to solve now take just minutes.

From: Matthew Heaney
 <matthewjheaney@earthlink.net>

Date: Mon, 28 Jul 2003 23:33:21
 Subject: Re: sorting large numbers of large records
 Newsgroups: comp.lang.ada

> Does anyone know of a good way to do this kind of sort?

The list containers in the Charles library have a (quick)sort operation. You could read the records into a singly-linked list container, and then sort that. Lists allow the sort to be done without copying elements, by exchanging pointers.

Alternatively, you could use a vector or array, and then use the generic sort algorithm (also part of the library) to sort that. A vector is more space efficient, but a sort requires copying of elements.

Another option is to read the records into a set, which automatically sorts elements as they are inserted into the container. (Sets and maps are implemented using a balanced tree.) A set is probably overkill, though, because you don't need do lookups or anything. If all you want to do is sort, then try using a list.

The Charles library is available from my website. [See above -- dc]

Look for a new release of Charles in the next few days. I am currently synchronizing the interfaces of the hashed vs. sorted associative containers. Send me email if you have any specific library questions.

[And from another message: -- dc]

> The [original poster] might just want to move to a real database, where problems like this have already been dealt with.

The indexed I/O components I wrote will probably meet his needs.

I'm not sure he really needs a full-blown database.

http://www.adapower.com/reuse/indexed_io.html

See also the Charles library for internal sorting methods: [...]

From: Matthew Heaney

<matthewjheaney@earthlink.net>

Date: Tue, 29 Jul 2003 03:38:23

Subject: Re: sorting large numbers of large records

Newsgroups: comp.lang.ada

Eventually, when I get around to implementing the full suite of generic algorithms, the Charles library should meet most sorting needs. However, the library is useable as is right now, and you can indeed sort every kind of container in the library.

I have been busy tweaking the containers. There will be a new release either this week or early next week. I will then be devoting the month of August (2003) to writing a proposal in response to AI-302, for a standard container library.

Cross-Compiler for H8300-hms Targets

From: Eduardo Ruhland

<ruhland@inf.ufsc.br>

Date: Thu, 29 May 2003 09:44:38

Subject: Ada cross-compiler Host = i386 Target = H8300-hms

To: GNAT Discussion List

<gnatlist@lyris.seas.gwu.edu>

I'm trying to build an Ada compiler (gcc) to program the Lego Mindstorms RCX. I need some directions on how to do it. I have build binutils for the H8300-hms and the binaries of this build are in my path. [...]

Eduardo Ruhland, Computer Science
B.Sc. Student, UFSC,
<http://www.inf.ufsc.br/~ruhland>

From: Michael Feldman

<mfeldman@seas.gwu.edu>

Date: Thu, 29 May 2003 11:06:51

Subject: Re: Ada cross-compiler Host = i386 Target = H8300-hms

To: GNAT Discussion List

<gnatlist@lyris.seas.gwu.edu>

A project at the US Air Force Academy was interested in using Ada to program the RCX. Instead of doing a full GNAT port, they decided to build on "Not-Quite-C" - a C dialect that's been used in a number of schools for RCX programming - and so developed a GNAT-based Ada -> NQC translator. Both NQC and the USAFA translator should be readily available.

Martin Carlisle, a USAFA professor who was involved in this project, is a gnatlist reader; I hope he will respond here. In any case, you can look Martin up at http://www.usafa.af.mil/dfcs/faculty/bio/DFCS_BIO_CARLISLE.HTM

From: Eduardo Ruhland

<ruhland@inf.ufsc.br>

Date: Thu, 29 May 2003 12:24:00

Subject: Re: Ada cross-compiler Host = i386 Target = H8300-hms

To: GNAT Discussion List

<gnatlist@lyris.seas.gwu.edu>

I think I can't use Ada/Mindstorms from USAFA because I want to control the RCX by setting register bits directly, i.e., I want to make a motor go forward by manipulating bits at the 0xf000 address for example.

From: Laurent Guerby <guerby@acm.org>

Date: 29 May 2003 18:39:45

Subject: Re: Ada cross-compiler Host = i386 Target = H8300-hms

To: GNAT Discussion List

<gnatlist@lyris.seas.gwu.edu>

[...] I suggest you try first to set up a C cross compiler, then you should try with Ada. GNAT has a "no run time" mode that can be quite useful for limited targets.

Someone built an Ada to 68HC11 compiler, may be you can find some useful hints there:
<http://gel.sourceforge.net/ada_example.php>. [See also "GNAT Compiler for 68HC11 Targets" in AUJ 24.2 (Jun 2003), p.76. -- dc]

RTEMS is an embedded run time, and supports a more complete Ada runtime (with tasking) on some targets, you might want to check

<<http://www.oarcorp.com/rtems/maillistArchives/rtems-users/>>

<http://www.oarcorp.com/RTEMS/RTEMS_Project/Mail_Lists/mail_lists.html>

[See also "OAR - RTEMS Operating System" in AUJ 22.4 (Dec 2001), pp.209-210. -- dc]

From: Joel Sherrill

<joel.sherrill@OARcorp.com>

Date: Fri, 30 May 2003 11:35:40

Organization: OAR Corporation

Subject: Re: Ada cross-compiler Host = i386 Target = H8300-hms

To: GNAT Discussion List

<gnatlist@lyris.seas.gwu.edu>

The RPM specs for the GNAT/RTEMS toolsets would probably be a good start. The source RPMS are at:

ftp://ftp.oarcorp.com/pub/rtems/snapshots/c_tools/gcc3.2.3newlib1.11.0-2/linux_x86/SRPMS/

You can grab any recent RTEMS source distribution for the source we use to generate the target CPU specific RPM spec files.

FWIW with gcc 3.2.3, we generate GNAT for i386, sparc, mips, and powerpc. With gcc 3.3, we can add m68k. We haven't tried the other RTEMS ports which do include the H8300, SH, and ARM. There seems to be interest in the ARM gnat.

[And from a later message: -- dc]

> You mentioned there is a port of RTEMS for SH. Do you know if anyone has it running on a Dreamcast video game system (it uses SH4)?

There are SH4 users but I think they are all on custom hardware. I personally don't know of anyone on a Dreamcast. But that would be a welcome BSP in the tree.

Joel Sherrill, Director of Research & Development, On-Line Applications Research, Huntsville AL

Status of Ada in GCC 3

From: Steve <steved94@attbi.com>

Date: Sat, 31 May 2003 21:38:25

Subject: Re: What's the status of GCC-3.x Ada?

Newsgroups: comp.lang.ada

> Can someone please tell me the current status of [the GCC-3.x] compiler? [...] I'm asking because I'd like - for a bit of fun - to port GNAT over to the PS2 (at work), but the only compilers available are 2.95.3 and a beta version of 3.2. [...]

Good places to start are: for the current gcc sources that include GNAT: <http://gcc.gnu.org/>; for the current ACT sources: <http://libre.act-europe.fr/>.

GNAT has become a part of the standard GCC library maintained on the gcc site, but ACT still maintains their own tree. The sources at the GCC site are the most current with respect to the back end, the sources at the ACT site are the most current with respect to Ada. The last I heard (on the GCC mailing list) is that ACT is working toward keeping the two source trees fairly well synchronized.

For a new port it's probably best to use the GCC site. Once you figure out how to use CVS you can get the latest sources from either site at any time.

Ada on a Sharp Zaurus PDA

From: wojtek@power.com.pl (Wojtek Narczynski)

Date: 9 Jun 2003 08:23:50 -0700

Subject: Ada on PDA - success!

Newsgroups: comp.lang.ada

I managed to build GNAT [...] compiler for Sharp Zaurus PDA -

<http://www.zaurus.com>. It was relatively easy, as Zaurus runs Linux. GCC 3.3 did not work for ARM, because there is something undefined for ARM architecture regarding exceptions and libgnat did'n want to link.

The GNAT version from libre.act-europe.fr worked. I had to fix a few glibc header parameter names, but it was pretty easy. [...]

From: wojtek@power.com.pl (Wojtek Narczynski)

Date: 10 Jun 2003 03:10:27 -0700

Subject: Re: Ada on PDA - success! Newsgroups: comp.lang.ada

> There are two Zaurus PDAs. The SL5600, which uses the IntelR 400MHz XScaleT processor and the SL5500, which uses the StrongARM SA-1110, 206MHz. Which does your version of GNAT work with?

With SL5600, but it should work on the slower one as well. Do you have the hardware to give it a try? [...]

From: wojtek@power.com.pl (Wojtek Narczynski)

Date: 10 Jun 2003 03:04:50 -0700

Subject: Re: Ada on PDA - success! Newsgroups: comp.lang.ada

> [...] compiling for another Linux platform is easy.

When you do it second time Seriously, in this case getting to know which sources to use was the most time consuming part. Maybe posting this information here will save somebody this time in the future.

> Try porting the runtime.

There is no need to port runtime for Zaurus, because it is supposed to run Linux. Yeah, currently it aborts with 'Tasking not supported', but I think it's the problem of bad choice of files for libgnat. I'll try to fix the Makefile later this week. [...]

[And from another message, sent 7 hours later: -- dc]

So I did, the diners example is now working over pthread tasking on Sharp Zaurus PDA.

Are the ACVC tests freely available? I cannot find them anywhere. And if not, are there some other tests I could run?

[One day later: -- dc]

> The ACVC test suite is now called ACATS. It is available at www.ada-auth.org.

My [...] compiler passed 2302 (+ 12 tentatively) of 2371 I managed to compile for it. I have no idea what it means though

From: wojtek@power.com.pl (Wojtek Narczynski)

Date: 11 Jun 2003 04:30:36 -0700

Subject: Re: Ada on PDA - success! Newsgroups: comp.lang.ada

> SAL (http://www.toadmail.com/~ada_wizard/) comes with a complete set of tests. They have found bugs in most of the pre-releases of GNAT, including the latest; if you are using GNAT from gcc 3.3, you'll find the same bugs.

I used GCC 3.2.3 and GNAT 5.01w (from libre CVS). It worked: ["test_all_harness" ran all 22 tests successfully -- dc].

From: Stephen Leake

<Stephe.Leake@nasa.gov>

Date: 11 Jun 2003 12:22:42 -0400

Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)

Subject: Re: Ada on PDA - success!

Newsgroups: comp.lang.ada

That's good. However, not all of the SAL tests are in test_all_harness; that's an AUnit test harness, and only the most recent tests are in it. Run SAL/Build/Gnat_Debug/Makefile for the complete list of tests; most generate an output file which is then diff'ed against a known good output file.

GNAT 5.01w from CVS should in fact have all the bug fixes.

GNAT for Mac OS X

From: jim hopper

<hopperj@macconnect.com>

Date: Thu, 17 Jul 2003 23:50:42

Subject: Re: ANNOUNCE: GNAT Programming System

Newsgroups: comp.lang.ada

[...] The Mac version is based upon Apple's update to gcc 3.2 which has a lot of fixes to it, and our update to the Ada code. Andrew and Alan Reynolds have done a lot of work on the compile to make it work well with Mac and fixed a number of bugs. A couple of companies have now ported Ada products to Mac using it, and I am using it with our companies line of radar sims and it works great, so while there are some things we would like to get fixed (like zero cost exceptions for Mac) we are pretty happy with how well it works.

[See <http://www.adapower.net/macos/> and also the same topic in AUJ 24.1 (Mar 2003), p.11. -- dc]

Reading Image Files

From: Patrice Freydiere <frett27@free.fr>

Date: Fri, 2 May 2003 08:48:08

Subject: Re: CONSTRAINT_ERROR

Newsgroups: fr.comp.lang.ada

[Translated from French: -- dc]

There is a very simple library to load PPM and PNM files in Ada. It is called OpenImage. You can find a copy on my site: <http://pfreydiere.free.fr>

[And from a later message: -- dc]

> By using the Jewl library, it is possible to display an image in bitmap format.

But does there exist a library or any other means to read and display an image in another format? [...]

With AdaGtk, and the ImLib extension, you can read various image formats... JPEG, TIF,.. but that requires the use of ImLib.

You can also use the "simple_jpeg_lib", which allows you to read files with in JPG format, and to build a table of RGB components. This library is available on the site <http://pfreydiere.free.fr> [See also "Simple JPEG Library" in AUJ 23.4 (Dec 2002), p.194. -- dc]

From: Patrice Freydiere <frett27@free.fr>

Date: Sat, 09 Aug 2003 13:29:03

Subject: Re: Binding to C

Newsgroups: comp.lang.ada

[...] if you are looking for building a complete full Jpeg binding, you can browse the AGP project (Ada Game Programming) it contains a JPEG binding, and a PNG binding. [...]

PS: when I started writing the simplejpeglib binding for Ada I had a lot of interfacing problems due to the portability macros of the jpeglib. (It's surely possible, but when I've started my binding, my C interfacing skills from Ada were poor.) So I decided in my design to build a simple C interface to make Ada binding easier. This permit me to be "C structure independent" and provide a more easy upgrade of the jpeglib.

[And from another message: -- dc]

> How about a link? AGP is hardly a unique search term.

I haven't found a web site for AGP, the only link I have is this <http://www.gnuada.org/alt.html>. You'll find a HREF to the sources.

"The Ada Game Package (AGP). The package `agp95_v0.1_x11.tar.gz` (1719 kB) is targeted for XFree86, the package `agp95_v0.1_fb.tar.gz` (1719 kB) for the GNU/Linux framebuffer console. There is also a partial binding to the Linux framebuffer console (17 kB). These packages compile with `gnat-3.11p`. [...] The software is GPLed, the authors are Wesley Y. Pan and Weston T. Pan."

TeXCAD - Drawing and Retouching Pictures in LaTeX

From: Gautier de Montmollin

<gdemont@hotmail.com>

Date: Fri, 09 May 2003 23:09:38

Subject: Ann: TeXCAD 4.0

Newsgroups: comp.lang.ada

TeXCAD is a program for drawing or retouching pictures in LaTeX. More info at <http://www.mysunrise.ch/users/gdm/texcad.htm>

Why this announce in [the comp.lang.ada newsgroup]? Guess why...

For the moment you'll find a MS Windows incarnation but the core of TC is completely portable, so people interested in making a Mac, Linux, Unix, ... version are invited to do so (and contact me!)

From: gdemont@hotmail.com (Gautier)
Date: 11 May 2003 02:34:20
Subject: Re: Ann: TeXCAD 4.0
Newsgroups: comp.lang.ada

> How much work would it be to port it to GtkAda 2.0? [...] I guess the major work is porting the drawing widget?

Not so much: the drawing is done in the generic package TC.Display, i.e. in the platform-independent part (unlike all TC.GWin.*). All you have to provide is the appropriate placeholders for ClearScreen, SetColor, PutPoint, SetLineStyle, Line, Rectangle, Ellipse, Arc, SetTextJustify, OutTextXY. See TC.GWin.Display for an example. It is straightforward.

The real "platform-dependent" challenge is to manage the interactivity: windows, menus, scrolling, mouse's ergonomics. Each system has its own specialities: multi-document windows and registry for MS Windows, on mouse button on the Mac, application owning a menu and windows (rather Mac) or a main window owning a menu inside of it as well as subwindows whose menus overlap the main menu, etc., etc.)

AdaOpenGL 0.10 Released

From: David Holm
<david@realityrift.com>
Date: Wed, 18 Jun 2003 12:05:58
Subject: AdaOpenGL 0.10 released
Newsgroups: comp.lang.ada

AdaOpenGL 0.10, the thin Ada to OpenGL binding, is finally out in the open. You can find it at <http://adaopengl.sourceforge.net/> as usual. This is just a minor update that consists of style fixes, an install target and building of a dynamic library.

[See also "New OpenGL Binding" in AUJ 24.1 (Mar 2003), p.12. -- dc]

I have three things to finish before releasing 1.0 and those being: AGL binding, WGL binding, style fixes in all files (I like -gnaty).

I have used the current version on both Linux and MacOS X although on MacOS you will have to use something like glut, SDL or Gtk to create the OpenGL window for now. I'm working on a minimalistic portable GL library which will be used to open a window and optionally make it fullscreen and reading input from a keyboard or mouse. I have it working on X11 and Carbon so far. I wrote it mainly because I didn't like the

way glut forces you to use callbacks and SDL is very Ada-unfriendly on MacOS X as it uses some strange hack in order to use Cocoa with C apps.

Have fun, and let me know if you have any cool projects that use AdaOpenGL. I'm always interested in screenshots for the webpage =>.

[And from another message: -- dc]

I used a BSD-style license so you can do pretty much whatever you want with it.

User Experiences with GtkAda Portability

From: Stephane Riviere
<stephane@rochebrune.org>
Date: Fri, 06 Jun 2003 19:08:16
Subject: Re: [ada-france] interfaces graphiques
To: ada-france@ada-france.org

[Translated from French: -- dc]

A developer of my team, at the time of my last job, tested GtkAda during several months, on real applications. The result is without any doubt: GtkAda, it works and it is terribly portable. Software developed under Windows, then ported to Linux: conclusion, 1 line modified (a Gtk call was not clean under Windows). Except for that: nothing to do. [...]

Now, with Gtk 2, one can have the native Windows XP look... I tested under Windows 2000 and it is already very close. But under XP, it is without comment... (see the URL).

Gtk-Wimp:
<http://sourceforge.net/projects/gtk-wimp>
 GTK-Wimp ("Windows impersonator") is a GTK theme that blends well into the Windows desktop environment. GTK-Wimp was created by Raymond Penners based upon code from Redmond95 by Owen Taylor and Nativewin by Evan Martin.

Features: The system font and colors from the Windows appearance control panel are used. When running on XP, the Windows theming API is used so that GTK widgets look like native widgets. On older Windows versions the theme behaves more or less the same as the Remond95 theme (but respecting system font and colors, of course).

Stéphane Rivière, Ile d'Oléron

GtkAda 2.2.0 Release

From: Arnaud Charlet
<charlet@gnat.com>
Date: Thu, 3 Jul 2003 10:26:48
Subject: ANNOUNCE: GtkAda 2.2.0 release
Newsgroups: comp.lang.ada

GtkAda is an Ada95 graphical toolkit based on Gtk+. It allows you to develop graphical applications in Ada95 using Gtk+ and Gnome.

The primary download site is <http://libre.act-europe.fr/GtkAda>

This version represents a major update compared to GtkAda 2.0.0, and is suitable for use with Gtk+ 2.2.0 and above. A binary package is also provided with this release for Windows platforms.

New features in 2.2.0:

- Support for Glade2.
 - Support for GtkExtra.
 - Support for libglade2.
 - New functions bound in Gtk.Label.
 - New type UTF8_String to make it clearer when UTF8 strings are expected.
 - New capabilities in GtkAda.MDI, see spec for details.
 - Support for drag-n-drop in GtkAda.MDI.
 - Bugs and memory leak fixes in GtkAda.MDI.
 - New package Glib.Graphs (was in GtkAda 2.0.0, but not announced).
 - Update to GNOME 2.0 API.
 - Memory leaks fixed in Glib.XML.
 - Code clean ups.
 - Fixed Gtk.Text_Buffer.Cut|Copy|Paste_Clipboard.
 - Speed improvements and code clean up in GtkAda.Canvas.
 - Full support for Pango (font handling and internationalization support).
 - New package Glib.Messages.
 - New package Glib.Unicode.
- [See also AUJ 22.3 (Sep 2001), pp.141-142, AUJ 22.2 (Jun 2001), p.73, and AUJ 21.3 (Oct 2000), p.163. -- dc]

Klokka - Small GtkAda-based Clock

From: Preben Randhol
<randhol@pvv.org>
Date: Thu, 17 Jul 2003 13:10:47
Subject: Re: Gcc-3.3 tripples the executable size.
To: GNAT Discussion List
<gnatlist@lyris.seas.gwu.edu>

I just went from gnat-3.14p to gnat-3.3 (gcc) and I recompiled GtkAda-2.2.0 and a small clock I made [...].

<http://www.pvv.org/~randhol/Ada95/Klokka/>

Bindings for GUILE Interpreter

From: Bobby D. Bryant
<bdbryant@mail.utexas.edu>
Date: Sat, 10 May 2003 13:10:13
Subject: Re: Ada & Scripting
Newsgroups: comp.lang.ada

> Have anyone written a binding, or is in the middle of writing a binding of some nice scripting language like Lua or Io? (I know there exist Python and Tcl bindings but those are rather slow)

If you do Scheme I have bindings for the GUILE interpreter at
<http://www.cs.utexas.edu/users/bdbryant/guile-for-ada/>

Bobby Bryant, Austin, Texas

[See also "Ada Bindings for GUILE (GNU Ubiquitous Intelligent Extension Language)" in AUJ 22.1 (Mar 2001), p.14. -- dc]

AdaGPGME 0.4.1 - Binding to "GNUPG Made Easy"

From: Andreas Almroth

<andreas@almroth.com>

Date: Tue, 01 Jul 2003 16:26:02

Subject: AdaGPGME (was Re: GnuPG binding?)

Newsgroups: comp.lang.ada

As there was some interest a couple of months back in an Ada binding to GnuPG, I thought I should let you know that there is a new version available at <http://www.almroth.com/gpgme>.

[From that URL: "AdaGPGME is used to interact with GNUPG for key management, encryption, decryption, and signing through an Ada95 API." See also "GnuPG Binding" in AUJ 24.2 (Jun 2003), p.78. -- dc]

The new version is 0.4.1 and reflects the major changes that has been made to the C API. I have also added a binding to gpg-error which is now used by gpgme. For further information on changes, I recommend reading the NEWS file in the gpgme-0.4.1 distribution.

New with this release is that GnuPG 1.2.2 must be used, as well as the new error-handling library gpg-error-0.1.

I'm still porting the regression tests, so if you find issues, bugs or just have general comments, please contact me.

[Direct URL for the archive file is <http://www.almroth.com/adagpgme-0.4.1.tar.gz> -- dc]

OS Bindings to Run an External Program

From: Martin Krischik

<krischik@users.sourceforge.net>

Date: Sun, 22 Jun 2003 19:20:53

Subject: Re: Running external program, getting output

Newsgroups: comp.lang.ada

> As far as I have gathered the best way of (non OS-specific) running an external program from Ada is using the Spawn procedure in g-os_lib, correct? But how do I get the output from the

command that I run? Lets say that I want to run "ls" for example.....

AdaCL has a support class do just this. See TestCommand (test 3) on how to do it: http://adacl.sourceforge.net/html/TestCommand-Main__adb.htm. You find AdaCL at: <http://www.ada.krischik.com>. [See "AdaCL - Ada Class Library" further in this AUJ issue. -- dc]

Also, you can use the GNAT.Expect package.

From: David C. Hoos

<david.c.hoos.sr@ada95.com>

Date: Sun, 22 Jun 2003 14:21:57

Subject: Re: Running external program, getting output

Newsgroups: comp.lang.ada

> or look at Adapower where J. Rogers has provided a "popen" binding

www.adapower.com/reuse/pipes.html [or:]

www.adapower.com -> "The Ada Source Code Treasury" -> "Source Code Packages for Reuse" -> "Ada Binding to Pipes"

From: Benoit Cartie

<benoit.cartie@airbus.com>

Date: Wed, 23 Jul 2003 18:30:36

Subject: [ada-france] lancement de commande

To: ada-france@ada-france.org

[Ada-France postings translated from French: -- dc]

I wish to launch a Unix script from an Ada program. I think that it is feasible via a C interface but do you know another method?

Benoit Cartié, Airbus-EYYWAG

From: Antille Julien

<julien.antille@opengl.ch>

Date: Wed, 23 Jul 2003 19:12:55

Subject: Re: [ada-france] lancement de commande

cc: ada-france@ada-france.org

This depends on the implementation. If you use GNAT, you can use the GNAT.OS_Lib package. You will be interested in particular in the procedures Spawn and Non_Blocking_Spawn (new process). The [package specification] is available at:

http://www.seas.gwu.edu/~adagroup/adali_b_html/gnat-html/g-os_lib.html

From: Thomas De Contes

<d.l.DeContes@free.fr>

Date: Thu, 24 Jul 2003 01:40:43

Subject: Re: [ada-france] lancement de commande

To: liste ada-france <ada-france@ada-france.org>

It is also possible with "OS_Services" of AdaLog. And besides I believe it is easier than with GNAT.OS_Lib, since one directly can execute a shell command, without having to type the command line

to launch the shell, this is already in OS Services: -) And it is portable: -)

[Direct link http://www.adalog.fr/compo1.htm#OS_Services -- dc]

POSIX Bindings

From: Jean-Pierre Rosen

<rosen@adalog.fr>

Date: Thu, 24 Jul 2003 09:24:53

Organization: Adalog

Subject: Re: [ada-france] lancement de commande

To: liste ada-france <ada-france@ada-france.org>

[Translated from French: -- dc]

> Where can I find "POSIX" packages?

For Linux, download Florist, the Ada/POSIX binding.

For Windows, use Win32Posix of Pascal Obry.

Links can be found on

<http://www.adalog.fr/adaweb.htm>.

GNADE 1.4.3 - GNAT Ada 95 Database Development Environment

From: Michael Erdmann

<michael.erdmann@snafu.de>

Date: Sun, 22 Jun 2003 09:14:09

Subject: Release of GNADE 1.4.3

Newsgroups: comp.lang.ada

[See also same topic in AUJ 24.1 (Mar 2003), p.14, and AUJ 23.4 (Dec 2002), p.197. -- dc]

The source code release 1.4.3 of the GNADE project is available at:

<http://gnade.sourceforge.net/> or <http://sourceforge.net/projects/gnade>

for the Linux, Windows and Solaris 2.8 platform.

This environment allows, using the GNAT Ada 95 compiler, to implement Ada 95 applications using RDBMS products such as MySQL, PostgreSQL, MimerSQL and Oracle via the ODBC interface or native bindings.

This release includes besides bug fixes the following features: thin bindings to the ODBC interface; ISO 92 embedded SQL preprocessor for Ada 95 generating code for the ODBC interface; Oracle Call Interface; bindings for MySQL; bindings for PostgreSQL; a small command line toolset to import/export tables and to executed SQL queries; documentation in pdf, postscript and html format.

From: Michael Erdmann

<michael.erdmann@snafu.de>

Date: Fri, 11 Jul 2003 10:25:56

Subject: Re: Release of GNADE 1.4.3

Newsgroups: comp.lang.ada

> Is there a document with new features and fixed bugs in this release?

Please look at the first chapter of the release note. This release is basically about some bug fixes and some major corrections for the mysql bindings.

ODB 0.4.0 - Framework for Object Persistency with GNAT

*From: Michael Erdmann
<michael.erdmann@snaflu.de>
Date: Mon, 21 Jul 2003 21:23:11
Subject: Release of ODB 0.4.0
Newsgroups: comp.lang.ada*

ODB is part of the GNADE project. It provides a simple framework for object persistency with Ada 95. The release is located at:

<http://gnade.sourceforge.net/>

This is still an early beta release, therefore the documentation is still a bit poor.

Dia & dia2code - Open Source Ada UML Tools

*From: okellogg@freenet.de (Oliver Kellogg)
Date: 7 May 2003 09:34:04 -0700
Subject: case tools
Newsgroups: comp.lang.ada*

> dia with the dia2code add-on.

If you want support for UML packages, CORBA stereotypes, and many other improvements, I recommend pulling the CVS version. For details, see http://sourceforge.net/cvs/?group_id=15307

[See also same topic in AUJ 23.3 (Sep 2002), p.138. -- dc]

The same goes for KDE-Umbrello, the Ada generator is being improved these days, see <http://webcvs.kde.org/cgi-bin/cvsweb.cgi/kdesdk/umbrello/umbrello/codegenerators/>

Benchmarking Basic Timing Parameters

*From: tmoran@acm.org
Date: Wed, 11 Jun 2003 20:29:50
Subject: Re: milliseconds and delay until
Newsgroups: comp.lang.ada*

See <http://www.adapower.com/reuse/tt.html> for a benchmarking program that will tell you how often your clock ticks, minimum nonzero delay, etc. [From that URL: -- dc]

This is a small, portable, program that finds and displays some of the basic timing parameters of an Ada compiler/OS/hardware combination. It gives the actual clock tick, the time for a zero or non-zero "delay", and a simple task switch. It helped me track down a highly visible performance problem that occurred on only one out of several platforms.

UML to Ada Working Group

URL: <http://www.dedicated-systems.com/VPR/layout/display/pr.asp?PRID=6060>

Mapping UML To Ada

Release Date: Thursday, July 03, 2003

Summary: Derek Russell gives insight in to this tricky issue.

[See also "Ada-UML Profile" in AUJ 24.2 (Jun 2003), pp.79-80. -- dc]

As systems become more complex and computer infrastructures expand, it is essential to change our methods and thought process when developing new applications. Within the defence sector, not only are the timescales associated with software development often longer than the life of the hardware platforms on which they are originally designed to execute. In addition, many of these new systems must interoperate with legacy systems.

The system designer must not only consider the system now, but must design for the future - a difficult task. Who knows what the future holds for the system? New facilities might be added, the hardware platform might change, or the implementation language might change! Implemented correctly, the UML can help address these issues.

When developing software, it is all too common to think in terms of the target language. However, keeping these thoughts to a minimum is the key to building a software system that is maintainable for years to come. It is, however, naïve to think that we can ignore the target language completely.

The mapping from a UML design to an Ada implementation is not as obvious as some would think. The size of the Ada language leads to many different ways of implementing even a simple UML model. In addition, the UML notation (1.4) is often not rich enough to fully specify the designer's intentions.

To avoid excessive model annotations, organizations (or toolset vendors) should define a set of rules for mapping UML modelling elements to the target language. These are termed mapping rules; a set of mapping rules should exist for each flavour of the target language. These mapping rules should provide the general, and most common, set of mappings. When code generation is performed, the appropriate mapping rule is selected and used by the code generator.

It cannot be assumed that the mapping rules would be sufficient to cover all possible implementation requirements. There will be model specific mappings such as active class priorities, entry point semantics, etc. To accommodate such requirements, language specific

annotations should be added to the model to allow specifics to be identified and enhance code generation. These annotations should be in the form of stereotypes and tags and are known as the profile (or Ada profile in the case of the Ada community). The profile should override the mapping rules if applied to any modelling element.

It is important that the design model is not unnecessarily cluttered with implementation specified annotations and therefore it is recommended that an implementation model is created to visualize the implementation specific elements. Normally annotations are identified as part of the low level integration and test activities. Designers should be encouraged to think as abstractly as possible at each stage of the development process. Adding this type of information too early causes thinking at too low a level. Building the implementation model forces developers to leave the implementation thoughts until the appropriate point.

Derek Russell has been nominated as Chairman of the UML To Ada Working Group. If you would like to learn more about the Ada Working Group or become involved please contact Derek Russell.

Contact: Derek Russell,
derek.russell@objektum.com

AdaCL - Ada Class Library

*From: Martin Krischik
<krischik@users.sourceforge.net>
Date: Fri, 09 May 2003 19:00:23
Subject: [Announcement] AdaCL 1.4.0 released.
Newsgroups: comp.lang.ada*

Well, I have released a new version of my very personal Ada Library:
<http://adacl.sourceforge.net/>

[From that URL: -- dc] AdaCL is a library to write small, script like, programs in Ada.

*From: Martin Krischik
<krischik@users.sourceforge.net>
Date: Fri, 16 May 2003 19:39:42
Subject: [Announcement] AdaCL 1.5.0 released.
Newsgroups: comp.lang.ada*

I have updated the AdaCL Library.

A new commandline parser has been added. All the demo programs have been changed to support the parser. Simple execution of external programs is now possible as well. For details see: <http://www.ada.krischik.com>.

I have also started to add some simple documentation to the Sourceforge documentation area. Its really only a start.

AdaCL supports GNAT and is tested under OS/2, Windows NT and Linux.

From: Martin Krischik
<krischik@users.sourceforge.net>
Date: Fri, 30 May 2003 08:50:09
Subject: [Announcement] AdaCL 1.6.0
released
Newsgroups: comp.lang.ada

Version 1.6.0 now features "in place" text filtering. That means that the in and out file can be the same. A backup is preserved. In OS/2 the extended attributes are preserved as well. Because of this, filtering with wildcards is now convenient without Rexx scripts.

The Sync execution of external commands has been improved. However Async and InOut execution is still work to do.

A file utilities package has been added. Trace has been improved as well.

An introduction Page has been added to the online documentation.

From: Martin Krischik
<krischik@users.sourceforge.net>
Date: Sun, 15 Jun 2003 15:59:47
Subject: [Announcement] AdaCL 2.0.0
released
Newsgroups: comp.lang.ada

AdaCL is a library to write small, script like, programs in Ada. The main tasks currently implemented are the filtering (search and replace) of text files, execution of external programs and command line parsing.

While GNAT itself has build in support for all theses functionality, AdaCL support them in an object oriented manner. Also care have been taken to keep all objects thread save.

Release Notes:

AdaCL.OS.Command can now redirect the input, output and error stream of the programs started into three Ada.Text_IO files.

Async execution of external programs now works with more than one program.

AdaCL.Trace has been extended to better support error and informational output.

Unless in "--verbose" mode the library will not produce any output any more.

Workarounds for the GNAT problems in OS/2 are now part of the library.

Note: The last of the 4 tests in TestCommand.Main does not work. This is due to the fact that I do not know how to tell "sort" that input has ended and output should start.

From: Martin Krischik
<krischik@users.sourceforge.net>
Date: Fri, 18 Jul 2003 19:19:56
Subject: [Announcement] AdaCL 3.0.0
released
Newsgroups: comp.lang.ada

AdaCL is a library to write small, script like, programs in Ada. The main functionality currently implemented are

filtering of text files (global search and replace), execution of external programs (inclusive redirection of standard input and standard output) and command line parsing.

To support the main functionalities of AdaCL a thick binding to the Boehm-Demers-Weiser conservative garbage collector and an extension kit for the Ada95 Booch Components are included as well.

Release Notes:

This Release features a thick binding to the Boehm-Demers-Weiser conservative garbage collector as well as an extension pack for the Ada95 Booch componentes.

These two features remove the need for the cumbersome reference counting used in the 1.x and 2.x releases.

Please see <http://adacl.sf.net> for details.

Nice Bit: The extensions for the Booch Components are largely generated from original source with the help of AdaCL's own text filter library. Please see http://adacl.sourceforge.net/html/sarBC-CommandLine__adb.htm for how it is done.

Grace.Config_Files

From: Stephen Leake
<Stephe.Leake@nasa.gov>
Date: 13 May 2003 13:54:01
Organization: NASA Goddard Space Flight
Center (skates.gsfc.nasa.gov)
Subject: Re: Ann: TeXCAD 4.0 (-epsilon)
Newsgroups: comp.lang.ada

[On an Ada application that uses the Windows Registry to keep configuration information: -- dc]

That's a job for Grace.Config_Files: http://www.nongnu.org/Grace/Grace_Home.html, currently only accessible via the CVS web interface: <http://savannah.nongnu.org/cvs/?group=grace>

Be aware that Grace.Config_Files is not yet fully implemented, but it may meet your needs. And it will be more portable and user-friendly than the Windows Registry. Feel free to contribute!

[See also "Platform Independent Access to Configuration Information" in AUJ 24.1 (Mar 2003), p.16. -- dc]

From: Preben Randhol
<randhol@pvv.org>
Date: Sun, 25 May 2003 10:02:43
Organization: Norwegian university of
science and technology
Subject: Re: config file library
Newsgroups: comp.lang.ada

> Is there any config file libraries in the wild (i.e. LGPL or public domain, so it could be linked in the proprietary program)? Something able to handle windows-like .ini file with sections and

comments or apache-like config would be fine. [...]

Yes Stephen Leake has made this: http://savannah.nongnu.org/forum/forum.php?forum_id=1201

Pretty-Printing Ada in LaTeX

From: maa@liacc.up.pt (=ISO-8859-1?Q?M=Elrio_Amado_Alves?)
Date: 4 Jun 2003 08:41:04 -0700
Subject: Re: Ada2latex Converter available?
Newsgroups: comp.lang.ada

> Is there a free Ada2latex - Converter available?

Ben-Ari has one on his book Ada for Software Engineers, published by Wilew. If there is a companion webpage, you're serviced

[See also same topic in AUJ 21.3 (Oct 2000), p.166. -- dc]

From: Preben Randhol
<randhol@pvv.org>
Date: Wed, 4 Jun 2003 11:28:03
Organization: Norwegian university of
science and technology
Subject: Re: Ada2latex Converter available?
Newsgroups: comp.lang.ada

Look at highlight at <http://www.andre-simon.de/>

[From that URL: -- dc] Highlight is a universal source code converter for Linux and Windows, which transforms code to HTML, XHTML, RTF, LaTeX or TeX - files with coloured syntax highlighting. There is support for 65 programming languages.

From: Frank Piron
<frank.piron@konad.de>
Date: Fri, 06 Jun 2003 10:53:46
Organization: KonAd GmbH
Subject: Re: Ada2latex Converter available?
Newsgroups: comp.lang.ada

Great! I changed the ada.lang config-file, since the ' is configured as a string delimiter, which causes problems with ada-attributes. After that I copied ada.lang to new files adb.lang and ads.lang, to enable the filename-extensions (adb,ads) for winhighlight. Finally I translated the generated tex-file with latex-pdf on windows (miktex-distribution). Very good result!

[These patches have been sent to the program's developer for integration. -- dc]

AdaDoc 2.01 - Documentation tool for Ada Package Specifications

From: Vincent Decorges <vega01@sf.net>
Date: Sat, 07 Jun 2003 13:00:18

Newsgroups: comp.lang.ada
Subject: ANN: AdaDoc 2.0

We are announcing the release of AdaDoc version 2.01. A maintenance release.

[See also same topic in AUJ 23.4 (Dec 2002), pp.200-201. -- dc]

AdaDoc is a tool for developers using the Ada95 programming language. Its goal is to create documentation in different format from a specification package.

The program analyzes the specification (by controlling its syntax), then the draft to write a document with the desired format. The format of exit depends on the selected module. AdaDoc writes a file XML (temporary) containing all information necessary to the modules to write the other formats (HTML, Latex, etc...). The writing of a module for other formats is very easy.

AdaDoc is freely available to the following address: <http://adadoc.sf.net>

One can find there: binaries versions for Linux and Windows, a user guide (English and French), a guide of creation of module (English and French), sources of the software (English), the complete documentation of the project (only in French), a mailing list to be informed of the new release. AdaDoc is under the GPL.

AdaBrowse 4.0 - HTML Generator for Ada 95 Library Unit Specifications

From: Thomas Wolf
<t_wolf@angelfire.com>
Date: Wed, 9 Jul 2003 22:40:17
Subject: ANN: AdaBrowse 4.0 released
Newsgroups: comp.lang.ada

I've published version 4.0 of AdaBrowse. It's available at the URL http://home.tiscalinet.ch/t_wolf/tw/ada95/adabrowse/ in source form and as pre-built executables for Win NT/2k and GNAT 3.15p as well as GNAT 3.16a. AdaBrowse is distributed under the terms of the GPL.

AdaBrowse 4.0 is a major step forward from the last announced release 3.0 from Aug 2002 (or even from the last published version 3.4.2 from June 2003). [See also "AdaBrowse 3.0 with XML support" in AUJ 23.4 (Dec 2002), p.200. -- dc]

New features:

1. User-definable indices. You can define what kinds of items shall go into any index using expressions of boolean queries like e.g. "(type or subtype) and not (task or protected or controlled)", which would produce an index of all types and subtypes that are not task or protected or controlled types, or "constant and not incomplete", which would produce an index of all constants and

named numbers, but not include deferred constants.

There may be any number of indices, each with its own expression, file name, and title.

2. GNAT project manager integration. AdaBrowse 4.0 can work with GNAT project files; it now also has a -P command line option. This simplifies using AdaBrowse considerably if you are using project files already.

Using project files, AdaBrowse now also fully support arbitrary naming conventions as defined in the project file.

The pre-built executables do have project manager support. If you're building from the sources, you'll need to have the GNAT sources, otherwise the Makefile will build an AdaBrowse without project manager support.

3. The XML output now also contains the indices. The DTD has been changed accordingly; it's now at V2.0.

Please read the User's Guide on how to work with these new features!

AdaBrowse 4.0 has passed all my regression tests. If you still find bugs, please report them to the address given in the User's Guide in the distribution.

Ada-related Products

ACT - Public Release of GNAT Programming System IDE (GPS)

From: Arnaud Charlet
<charlet@gnat.com>
Date: Tue, 15 Jul 2003 18:19:52
Subject: ANNOUNCE: GNAT Programming System
Newsgroups: comp.lang.ada

Ada Core Technologies and ACT Europe are pleased to announce the first public release of GPS, the GNAT Programming System IDE, for the GNU/Linux, Solaris and Windows platforms.

[See also "ACT - GNAT Programming System IDE (GPS)" in AUJ 24.2 (Jun 2003), p.82. -- dc]

Designed by programmers for programmers, the GPS IDE integrates the GNAT Ada 95 tools within a single visual development environment. GPS is Free Software. This version is intended for use in academic and Free Software projects.

GPS is available at <http://libre.act-europe.com/gps>

From: Arnaud Charlet
<charlet@gnat.com>
Date: Thu, 17 Jul 2003 11:19:36
Subject: Re: ANNOUNCE: GNAT Programming System
Newsgroups: comp.lang.ada

It's very nice to see all these answers and excitement about GPS.

May I suggest that people having specific questions on how to build or use GPS go to <http://libre.act-europe.fr/gps> and subscribe to either of the two existing mailing list dedicated to that purpose? [...]

ACT - GNAT Pro 3.16a1

From: Jamie Ayre <ayre@act-europe.com>
Date: Fri, 1 Aug 2003 15:41:25
Subject: [GNATPro-announce] GNAT Pro 3.16a1 release
To: <gnatpro-announce@act-europe.fr>

We are pleased to announce the release of GNAT Pro 3.16a1, our high performance Ada development solution.

This latest version offers exciting improvements in our base debugger technology - GDB 5.3, our new IDE - GPS 1.2.2, and our graphical toolkit - GtkAda 2.2.0.

[See also "ACT - GNAT Pro 3.16a" in AUJ 24.2 (Jun 2003), p.82. -- dc]

GNAT Pro's availability over an ever increasing number of native and cross platforms, combined with exceptional, responsive support from Ada experts provides for focused, time saving, project development.

Find out how our tools and service can make the difference. Contact us at sales@act-europe.com or sales@gnat.com

GNAT Pro: http://www.act-europe.fr/texts/products/prod_intro.htm;

GPS: <http://www.act-europe.fr/texts/products/gps.html>;

GtkAda: http://www.act-europe.fr/texts/products/prod_gtk.htm;

Platforms supported:

http://www.act-europe.fr/texts/products/prod_platforms.htm;

Support: http://www.act-europe.fr/texts/services/service_support.htm.

GNATPro-announce mailing list:
GNATPro-announce@lists.act-europe.fr;
<http://lists.act-europe.fr/mailman/listinfo/gnatpro-announce>

Aonix - ObjectAda for Windows 7.2.2 Patch Update

From: Aonix Ada Support
<adasupport@aonix.com>
Date: Wed, 18 Jun 2003 19:09:17
Subject: Intel-OA: New ObjectAda updates
To: intel-objectada@aonix.com

New patch updates for ObjectAda for Windows 7.2.2 (1102) are now available [...] The update download files and the Release Notes are available at <http://www.aonix.com/index-oap.html>

Please see the Release Notes for more information.

Aonix Ada Support,
http://www.aonix.com/support_ada_contact.html

ARTiSAN - Real-Time Studio 4.3 Release Continues to Improve Ada Support

From: Richard Gastwirt

<richard.gastwirt@artisansw.com>

Date: Mon, 30 Jun 2003 14:08:22

Subject: Press Release: ARTiSAN Software Tools Announces the Release of Real-time Studio 4.3

ARTiSAN Software Tools Announces the Release of Real-time Studio 4.3 Offering UML 2.0 Compliant Template Packages

ARTiSAN Real-time Studio 4.3 offers template packages, improved C, C++, and Ada Synchronizers, Improved API

[See also "Artisan - Real-time Studio's Ada Support" in AUJ 24.1 (Mar 2003), pp.18-19. -- dc]

Cheltenham, UK - June 30, 2003 - ARTiSAN Software Tools, a global leader for UML-based, real-time systems and software modeling tools, today announced the release of Real-time Studio version 4.3.

Real-time Studio 4.3 includes improvements to the C, C++, and Ada Synchronizers, a matured API, and support for features of UML 2.0.

According to Alan Moore, ARTiSAN's VP of Product Strategy and a member of the OMG's UML 2.0 finalization task force, "Model Driven Architecture (MDA) is predicated on the effective use of model fragments larger than components. UML 2.0 meets this need with the introduction of the concept of template packages, and ARTiSAN is the first company to implement the metamodel and notation in a released product."

Real-time Studio 4.3 sets another benchmark for flexibility, featuring an improved read/write API for accessing models using OLE automation. This API allows users to customize the tool to fit their exact needs. ARTiSAN now makes available a complete model of the API, including diagrams, with extensive documentation including examples. "This truly differentiates Real-time Studio from its competition giving users complete flexibility to seamlessly dovetail a COTS product into an established development process," added Moore.

Improvements to the C Synchronizer include full template customization of the generated code. This gives users complete control in deciding how they wish their C code to be generated from UML models. In addition, a C profile is included that is used to extend the UML meta-model to map to C more directly. The Synchronizer also includes a full differencing

capability, so users can graphically highlight deltas between code and the UML model. The C++ Synchronizer also benefits from the improved generation templates, now supporting nested classes and function templates and offering improved parsing and differencing.

The Ada Synchronizer continues to improve in light of wider usage in the community and now supports a more complete range of Ada95 constructs in reversing and generation. This includes protected types, separates, abstract operations and types, and more. It also offers more powerful template customization features than previously offered. As part of acceptance testing of the new Synchronizer, one customer reverse engineered 300k lines of Ada project code successfully in under an hour, enabling them to generate UML models based on this legacy code. This customer estimated that the same job would have taken several man-years without using RTs.

"We continue to be focused on Ada, and are extremely pleased with the reaction of the Ada user community," noted Jeremy Goulding, President and CEO of ARTiSAN. "We signaled our strategic intent with the release of the first Ada support in a real-time UML tool in 2001, and our success within this community has reinforced that decision to invest in Ada. Real-time Studio version 4.3 is the third major release of an Ada Synchronizer in RtS in as many years, and in contrast with other vendors that have blown hot and cold regarding their support of Ada, we have very intention of continuing to provide a way for our customers to leverage their investment in legacy Ada, to bring that intellectual property into a UML modeling environment and reuse it for their future designs."

About ARTiSAN Software Tools

ARTiSAN Software Tools provides electronic product development teams with the "fastest path to the right product." The company offers products, services and a process for systems and software modeling to accelerate the development of next-generation real-time systems while ensuring that they meet requirements. ARTiSAN enables engineering teams to visualize, design and validate systems before building them, and simplifies implementation with code generation and software reuse. ARTiSAN is a practical partner offering easy-to-adopt, down-to-earth solutions. ARTiSAN Software Tools, Inc., founded in March 1997, is privately held with headquarters in Cheltenham, United Kingdom. The company has regional sales offices and distributors throughout the world. For more information, call +44 (0)1242 229300 internationally, or

1(888)511-7975 from the US; or visit www.artisansw.com

Press Contact: ARTiSAN Software Tools, Richard Gastwirt [...]

DDC-I - SCORE-653 Development Environment for CsLEOS RTOS

URL: http://www.ddci.com/news_vol4num4.shtml

Subject: Embedded News from DDC-I - DDC-I Online News

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

SCORE-653 for CsLEOS RTOS - A Strong Platform for Safety Critical Software Development

[See also "DDC-I - Joins Forces with BAE SYSTEMS on ARINC-653 Operating System" in AUJ 24.1 (Mar 2003), p.19. -- dc]

The integration of DDC-I's SCORE (Safety Critical, Object-oriented, Real-time Embedded) suite of programming and testing tools with the CsLEOS RTOS (Real Time Operating System) from BAE SYSTEMS Controls combines a robust development environment and an ARINC-653 compliant RTOS that are both designed for use in the development of high-integrity embedded systems. For the first time, the SCORE multi-language support provides the application developer the ability to utilize C, embedded C++ & Ada for developing ARINC-653 applications to run with the CsLEOS RTOS.

The SCORE Integrated Development Environment with CsLEOS RTOS support runs on Solaris and Windows NT platforms to generate and debug applications targeted for the PowerPC family of processors. The SCORE CsLEOS RTOS support encompasses multiple language compilers, an Ada librarian, a target linker, disassembly tools, testing tools, and a Multi-Language Debugger (MLD), all unified through a Graphical User Interface (GUI).

The CsLEOS RTOS

The CsLEOS Real-Time Operating System is the first commercial, off-the-shelf RTOS, offered by a safety-critical systems company that is already certified to the highest FAA DO-178B safety level. Designed from the outset to implement ARINC-653 spatial and temporal brick-wall partitioning, the CsLEOS RTOS ensures that safety-critical functions are protected from other processes running on the same hardware.

The architecture also makes it possible to add, revise, and test system functions without re-certifying the entire application. With the cost of the certification process being so high, this can significantly reduce the bottom line

when developing or enhancing airworthy software.

The RTOS's ARINC 653-compliant API (Applications Programming Interface) is a true open system, offering users the ability to develop application software to a standard set of interfaces. This ensures ease of use and the most efficient use of development resources.

The CsLEOS RTOS is available now; already certified to DO-178B, Level A (D0-178B, developed by RTCA Inc., a nonprofit company in Washington, D.C., is the international standard for certifying software used in safety-critical airborne systems,) eliminating the need for the application developer to certify the operating system as well as their own application code in the final product.

CsLEOS RTOS has pre-configured BSPs (Board Support Packages) for a wide range of PowerPC target boards. The CsLEOS RTOS BSPs contain all the required code for initializing and utilizing all the board specific hardware required by the kernel.

SCORE-653 Support for the CsLEOS RTOS

The SCORE-653 development environment supports compilation, linking, testing, and debugging of applications running on top of the CsLEOS RTOS. Applications can be written in embedded C++, C, Ada, or a mixture of the languages. Ada and C APIs provide access to CsLEOS RTOS processes to support multi-process applications, including processes, error handling, and inter-process communications, all compliant with the ARINC-653 standard.

The SCORE-653 system has a clean, board-independent interface to the CsLEOS RTOS, which is provided through the SCORE-653 CsLEOS RTOS UCC (User Configurable Code). This interface is the link between the SCORE target libraries, primarily the SCORE Run-Time System, and CsLEOS RTOS. The interface has been designed to ensure that hardware and board specific details are positioned entirely within the CsLEOS RTOS domain. This means that the SCORE target code is 100% board independent and that SCORE-653 CsLEOS RTOS applications can run on any hardware that has a CsLEOS RTOS BSP (Board Support Package).

An Ada library is provided with interface packages containing Ada bindings to the CsLEOS RTOS system calls. This provides a seamless interface from Ada to CsLEOS RTOS while maintaining the safety of Ada's inherent type checking. An Interface pragma has been included in the Ada compiler, which allows the user to check at compile-time that no unsafe features have been included in the application code. The combination of

Ada's reliability, provided through strong typing and rigorous checks, and the built-in safety features of the certified CsLEOS RTOS, offers a strong platform for safety-critical software development.

The complexity of generating a set of CsLEOS RTOS executables is eased by automatic transfer of addresses between the CsLEOS RTOS configuration files and the SCORE build process. Partition addresses from the CsLEOS RTOS configuration database must correspond to those entered for the SCORE build process. Normally these hex addresses would need to be transferred by hand, but the integration of SCORE and CsLEOS RTOS has automated that process, reducing the possibility of transcription errors.

SCORE also provides a PowerPC simulator to allow source code debugging of algorithms on a PC. Within the simulator the user also has full access to the PowerPC register set and can examine the execution of programs at machine code level. Debug statements may be added to help trace execution and will display information on a dedicated window. The CsLEOS RTOS provides a special BSP to support the simulator.

On the real target hardware, symbolic debugging of the source code is accomplished through a JTAG interface. Applications will run in real time until a breakpoint is reached. The SCORE MLD (Multi Language Debugger) will seamlessly debug Embedded C++ code, C code, Ada code, and machine code, changing its operating mode automatically to match the language at the point of visibility. Watch windows allow the monitoring of both registers and expressions to ease the detection of changing data.

Based on personal preferences, the SCORE-653 CsLEOS RTOS user can choose either to work from the command line using SCORE commands, or from the SCORE GUI, which integrates all the SCORE-653 tools. Being able to compile and link from the command line allows the user to automate the compile and build processes, while the easy-to-use GUI executes SCORE commands with a single mouse-click.

The SCORE-653 MLD (Multi Language Debugger) can be used either from the GUI or from the command line to perform the usual Ada and C source code debugging of CsLEOS RTOS applications. In addition, SCORE has added expression watch windows that allow the user to automatically evaluate an expression at each tracepoint or breakpoint. Extensive machine code debugging options allow disassembly of machine code instructions, single stepping of machine code, and display and monitoring of register contents. The MLD interfaces to the hardware under test

through a JTAG connector. The MLD can be used to download programs onto the hardware, or can provide full symbolic, source level debugging of programs already downloaded into RAM.

The debug process may be automated through the powerful command language, which may be embedded in startup and initialization files, and session logging ability. This allows the user to create automated test scripts which can be used to perform white box testing. The ability to repeat the same tests easily is invaluable during regression testing.

To support testing even more thoroughly, SCORE is integrated with the SCORECast toolset, an automated module test system designed for embedded systems. This provides unit level testing of C, Ada, and C++ code as it runs on the PowerPC simulator. It will show source level coverage at the statement level, the branch level, and MC/DC level, an essential tool for certifying to DO-178B level A.

In summary, SCORE-653, a natural integration of DDC-I's Safety Critical Real Time multi-language development environment and BAE SYSTEMS Platform Solutions CsLEOS ARINC-653 compliant Real-Time Operating System, is the solution of choice for applications requiring DO-178B certification or any application where safety is paramount.

DDC-I - Windows Migration Package for TADS-i960 Customers

URL: http://www.ddci.com/news_vol4num4.shtml

Subject: *Embedded News from DDC-I - DDC-I Online News*

[...] May 2003 [...] DDC-I Online News
DDC-I Provides Current TADS-i960 Customers With Cost-Effective Windows Migration Package

Phoenix, AZ May 15, 2003. To streamline the transition from VAX or UNIX-hosted development systems, DDC-I announces the availability of a new Windows (NT/2000/XP) migration package for existing TADS-i960 users. A fully customizable limited-time package, it offers current customers a simple, affordable migration path to the most popular PC-based network and enterprise computing platform.

"Our flexible TADS for Windows migration package allows customers to define which tools and support they require, rather than sticking them with a rigid list of tiered options," explains Harold "Bud" Blum, DDC-I Senior Software Engineer and Product Champion for the TADS-i960 product line.

With DDC-I's expert guidance, customers dictate package parameters, creating a

least-cost migration path including necessary license transfers and keys to replace all current TADS licenses. To keep recurring costs level, software support from any current license agreement carries over, and the customer has complete freedom to select the quantity of seats to rehost and whether to upgrade their software versions during the migration.

The package also includes two days of on-site consulting at no additional charge to assist with rescripting, memory and segment set up, tool adaptation, related Ethernet work and UCC upgrading. A final project report with detailed recommendations is included.

"Our customers need their safety-critical software development tools to keep pace with their development environment, and upgrading our TADS products for the Windows platform allows them to intelligently utilize their computing assets with minimal disruption to the development environment they depend on," Blum concludes.

DDC-I - Updated Sun/Solaris TADS Ada Development System v5.2.4

URL: http://www.ddci.com/news_vol4num5.shtml

Subject: Embedded News from DDC-I - DDC-I Online News

[...] June/July 2003 [...] DDC-I Online News

DDC-I Releases Updated Sun/Solaris TADS Development System

Phoenix, AZ and Lyngby, Denmark. July 15, 2003. DDC-I is pleased to announce the release of maintenance upgrades for their Sun/Solaris-hosted TADS Ada Development System (v5.2.4), incorporating important improvements and enhancements realized during their recent TADS for PC/Windows rehosting development program. All current TADS targets, MIL-STD-1750A, Intel 960 MC/KB and Motorola 68xxx, are covered by the release.

"Providing consistency across the TADS product line for all hosts and targets is the best way we can support every customer's changing needs, whether they're rehosting legacy development projects to the PC/Windows platform, or performing normal maintenance activities on existing programs using the Sun/Solaris infrastructure," explains Harold "Bud" Blum, DDC-I Senior Software Engineer and TADS Product Champion.

Specific updates include improved REM and MOD operations, which now operate on 16-bit unsigned integers, and code generation for comparison of two 16-bit unsigned integers. The linker has also been enhanced to more effectively handle

situations where a module name specified in a linker control file could not be located due to the restructuring required to produce the corresponding internal compiler-generated module name.

A mature and reliable solution for each target processor, TADS combines a highly optimizing compiler with selective linking and modular run-time systems to generate the most compact and efficient code available. Classical and Ada 83-specific compiler optimizations offer code size and performance benefits specifically tailored to the individual processor architecture, alongside a complete development toolset for each target chip.

"TADS is a proven and trusted development environment for real-time embedded systems in numerous aerospace, avionics, defense, and other safety-critical programs. DDC-I remains dedicated to providing unsurpassed software tools for all of our clients, as well as superior customer support and engineering services," concludes Blum.

DDC-I - TADS-68xxx Ada Development System v6.0 Now Also Supported on Windows Platforms

URL: http://www.ddci.com/news_vol4num5.shtml

Subject: Embedded News from DDC-I - DDC-I Online News

[...] June/July 2003 [...] DDC-I Online News

DDC-I Announces Full Support for Windows NT, 2000, and XP Host Environments for the TADS-68xxx Software Development System

Phoenix, AZ and Lyngby, Denmark July 21, 2003 DDC-I today announces the addition of support for Windows based development platforms for the TADS Ada Development System (v6.0) targeting the Motorola 68xxx. Now offering Windows NT4.x and 2000/XP Professional host capabilities, TADS-68xxx also supports the original Sun SPARC, and DEC VAX/VMS host development environments.

"The enhancement of TADS-68xxx successfully completes our migration of the proven TADS product family of safety-critical real-time software development tools to the Windows enterprise network platform," explains Harold "Bud" Blum, Senior Software Engineer at DDC-I and Product Champion for the TADS product line.

All Windows hosted TADS-68xxx, TADS-1750A and TADS-i960 tools support both VAX-style and Solaris-style command-line options, mitigating the porting effort for existing build scripts.

Runtimes built with TADS v5.x toolsets can normally be reused with v6.0.

A highly optimizing compiler, modular runtimes, and selective linking generate fast, compact code, with classic optimizations tuned for the 68xxx architecture. Ada-specific optimizations include constraint and overflow check elimination, data packing, and static aggregate initialization, alongside 68xxx-specific optimizations like auto-increment detection, condition code tracking, minimization of procedure overhead and exception response time.

The AdaScope debugger provides full source- and machine-level debugging in a customizable environment. Other tools include AdaList and AdaRef, static analyzers to help programmers navigate the source code and locate component information.

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

RainCode - New XMLBooster Version for Ada Released

From: Carole Devillers

<carole@raincode.com>

Date: Tue, 20 May 2003 12:58:44

Subject: XML parsing solution for Ada
Newsgroups: comp.lang.ada

A completely new XMLBooster version for Ada has been released. It includes a number of new features for more efficient integration of XML in Ada applications.

[See also "RainCode - XMLBooster for Ada Released" in AUJ 23.3 (Sep 2002), pp.145-146. -- dc]

XMLBooster is the fastest solution for XML integration in Ada. By using native code generation rather than a generic approach, XMLBooster guarantees top notch performance for demanding transaction-based applications:

- * 5 to 50 times faster than a generic XML parser.
- * Generates parser from DTDs, SCHEMAS, or even just sample XML instances.
- * Applications can be deployed royalty-free.
- * Reduced memory footprint, can deal with even the largest inputs.
- * Support for mixed content and permutations.

- * Support for regular expressions, enumerations, boolean values, etc.
- * Multi-lingual support to generate the Ada server and the Java/COBOL/C++/C/Delphi client simultaneously.
- * Licences start as low as 675 US\$.
- * Freely available XMLBooster Lite version.

<http://www.xmlbooster.com>

From: Carole Devillers

<carole@raincode.com>

Date: Fri, 23 May 2003 18:11:14

Subject: Re: XML parsing solution for Ada
Newsgroups: comp.lang.ada

Here are the answers to your questions:

1) Authentication, verification, validation,...

XMLBooster produces naturally validating parsers, far beyond the validation possibilities of DTD's.

2) Is there also a MacOSX version available?

XMLBooster runs natively on Windows, and is portable on virtually any other platform, as we can deliver the software in a shrouded C source form, to be recompiled on your specific platform.

You can try the free evaluation version to give you an idea of what it can do for you. Other questions are welcome...

From: Carole Devillers

<carole@raincode.com>

Date: Thu, 10 Jul 2003 11:35:30

Subject: Re: XML parser for Ada83
Newsgroups: comp.lang.ada

> I'm looking for a XML Parser in Ada 83. Is there any? Is it better to include a parser written in C and if so, which one? Is it easier to port a parser from Ada 95 back to Ada 83?

Yes, there is at least one: XMLBooster. You can download the free Lite version here: <http://www.xmlbooster.com/downloadnew.html>

Top Graph'X - PrismTech Adds CORBA Ada Support to its OpenFusion Product Line

URL: http://www.topgraphx.com/version_am/fichier_press_release.htm
Subject: Press release

PrismTech adds Ada support to its OpenFusion Product Line

Enabling safety critical CORBA support for users in the defense, aerospace, communications and transportation sectors

OMG's Real-Time and Embedded Systems Workshop - Arlington, VA, USA - 14 July, 2003 - PrismTech, a leading vendor of distributed software infrastructure, today announced that it had

added an Ada version to its OpenFusion product line by signing an exclusive distribution agreement with French based vendor Top Graph'X.

Since its introduction in 1997 as the first full CORBA Ada solution, the Top Graph'X OrbRiver Ada ORB has gained a very high reputation amongst the Ada user community particularly in the defense, aerospace, communications and transport sectors with major users such as Eurocontrol, Thales, CNES, Canal+ Technologies, EADS, Raytheon, Northrop Grumman and the US Navy.

The product has an unmatched set of features and benefits for CORBA Ada users including extensive real-time support:

- * A mature product based on a 7 year legacy of Ada development for customers worldwide.
- * Proven efficiency and reliability.
- * High performance, including fast processing of Any Types.
- * CORBA 2.6 compliant.
- * A pure Ada95 language implementation making it highly portable across multiple platforms.
- * Available for native and cross-development platforms.
- * Optimized memory safe behaviors.
- * Pluggable transports: IIOP, MIOP (reliable MIOP coming soon), Shared Memory.
- * Support for CORBA Messaging implementation including AMI and QoS control.
- * Support for CORBA Real-time specification v1.0.
- * Naming, Event and highly performant Notification Ada95 CORBA Services.

"We are very pleased to be entering into this relationship with PrismTech, a vendor with an outstanding reputation for its CORBA products, as it will allow our OrbRiver Ada products to be marketed to a much wider audience particularly in the USA," said Jean-Claude Mahieux, VP Sales and Marketing, Top Graph'X.

"After an exhaustive evaluation of possible partners, we are delighted to be working closely with Top Graph'X who consistently achieved the highest marks in our selection process," said Dr. Shahzad Aslam-Mir, Product Manager, PrismTech. "We are convinced that they have the highest quality, most reliable Ada ORB implementation available in the market backed by an excellent reputation amongst their existing customers."

"This agreement is part of our continued strategy to bring a well integrated complete suite of CORBA middleware to the safety critical industry of which the Ada community is an extremely important

part," added Keith Steele, CEO, PrismTech. "We are already unique in the embedded and real-time CORBA marketplace for the diversity of languages supported and this new alliance allows us to fully meet the varied needs of both our existing and future customer base."

Further information about OpenFusion OrbRiver can be found on PrismTech's website at <http://www.primstechnologies.com/English/Products>

Ada and Linux

GNAT 3.15p RPM Packages

From: Oleksandr Havva

<alex@Lviv.Bank.Gov.UA>

Date: Fri, 25 Jul 2003 15:31:14

Subject: ANN: GNAT-3.15p RPMs are available

To: GNAT Discussion List

<gnatlist@lyris.seas.gwu.edu>

There are GNAT-3.15p "Ada for Linux Team" like RPMS:

<http://www.alex.wubn.net/packages/3.15p/RPMS> [and] SRPMS.

This build was been done on the RedHat-7.2. I hope it will be interesting for somebody else

Ada and Microsoft

NT/Win95 Console Package

From: Jerry van Dijk

<jvandyk@attglobal.net>

Date: 19 Jun 2003 00:24:28

Subject: Re: help with Ada95
Newsgroups: comp.lang.ada

> Jerry van Dijk wrote a nice Package for console-handling with win32. Unfortunately I do not know where to download it.

See my homepage. [Direct URL is: <http://users.ncrvnet.nl/gmvdijk/packages.html#CONSOLE--dc>]

Jerry van Dijk, Leiden, Holland

Ada Terminal Emulator for Windows

From: rossh@zip.com.au (Ross Higson)

Date: 22 Jun 2003 21:35:24 -0700

Subject: Ada Terminal Emulator for Windows

Newsgroups: comp.lang.ada

I've seen quite a few requests here recently for ANSI terminal support for Windows (notably Windows NT/2000/etc that no longer include the ANSI.SYS driver).

While that's not the main purpose of the Ada Terminal Emulator (see summary description below), it happens to do it quite well.

You can either run existing ANSI applications (Ada or non-Ada) using the "redirect" utility included in the package, or (Ada only) recompile them after substituting "Term_IO" for any references to Text_IO. Both methods replace the default files (i.e. stdin, stdout, stderr) with ANSI compatible terminal windows.

You can get the Ada Terminal Emulator from [<http://www.members.optusnet.com.au/rosshigson/>]

Note that the package also requires GNAT, and also GWindows (available from www.adapower.com). For more details, see the documentation in the Terminal Emulator package.

Ada Terminal Emulator - Summary

The Ada Terminal Emulator package provides a set of terminal emulation capabilities, implemented in Ada 95, to run under Windows 95/98/NT/2000. All source code for the terminal emulator is provided under the GNU General Public License. The package was developed using GNAT and GWindows.

In addition to providing simple "dumb" terminal emulation, the package provides full emulation of DEC VT52/VT100/VT102 compatible terminals, including double height and double width characters, smooth scrolling, special graphics, display controls and national replacement character sets. The emulator also implements a substantial subset of VT220/VT420 and ISO 6429 capabilities.

The package consists of three main components:

Terminal_Emulator: An Ada package that provides facilities for creating, configuring and interacting with terminal windows from an Ada program. Multiple terminal windows can be created from the same program. Each window implements a completely independent terminal emulator. Each can be separately configured as a simple "dumb" terminal, or emulate full DEC VTxxx capabilities. Major features: [see URL above -- dc]

Term_IO: An Ada package that can be used as a complete and transparent replacement for the standard Ada text-handling package Text_IO. In addition to supporting all the normal file handling capabilities of Text_IO, Term_IO replaces the default files (i.e. standard input, standard output and standard error) with one or more terminal windows. It also allows the creation and manipulation of user-defined terminal windows using only Text_IO syntax. Each terminal window can be configured to either strictly implement the semantics of Text_IO as defined in the Ada 95 Language Reference Manual (LRM), or to

implement relaxed semantics that allow the user to take advantage of the terminal capabilities in a more intuitive manner.

Redirect: A stand-alone program that allows existing text based application (including non-Ada applications) to have their standard I/O (i.e. standard input, standard output and standard error) redirected through a terminal window - without recompiling or relinking. Redirect supports all the features of the Terminal_Emulator, and adds line editing, command and filename completion and command history. It can be used to create a sophisticated window-based command line interpreter with only a few dozen lines of programming. A complete Ada 95 example is included.

From: rossh@zip.com.au (Ross Higson)
Date: 24 Jun 2003 16:54:15 -0700
Subject: Re: Ada Terminal Emulator for Windows

Newsgroups: comp.lang.ada

[...] By the way, I've updated the terminal emulator package to include a simple "comms" wrapper program that allows a PC to be more easily used as a serial terminal. Hopefully, this will encourage someone to do a side-by-side comparison with a real terminal.

From: rossh@zip.com.au (Ross Higson)
Date: 26 Jun 2003 16:38:54 -0700
Subject: Re: Ada Terminal Emulator for Windows

Newsgroups: comp.lang.ada

The terminal emulator now compiles with GNAT 3.14p as well as 3.15p, but it required a small source change. The latest source distribution (v0.5) is available [from <http://www.members.optusnet.com.au/rosshigson/>]

I've also added a compiled version of the stand-alone Comms and Redirect programs for anyone who wants to try out the emulator without having to install GWindows first. But you will probably still need the source distribution, since it contains the documentation.

References to Publications

Ada Compatibility Guide Still Available

From: rod.chapman@praxis-cs.co.uk (Rod Chapman)
Date: 26 May 2003 07:57:45
Subject: Re: Difference between Ada 83 and Ada 95
Newsgroups: comp.lang.ada

> Does somebody know or have a short overview of the difference between Ada 83 and Ada 95?

Check out the Ada83/Ada95 compatibility guide at <http://www.adaic.com/standards/ada83.html>

[Note: text, PostScript and Word versions of this document are available at <http://archive.adaic.com/docs/compat-guide/--dc>]

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>

Date: Fri, 30 May 2003 17:10:21

Subject: Real-Time Industry Updates - News From DDC-I

To: R9DK May 2003 Online News <jcdk@ddci.com>

DDC-I Online News. Real-Time Industry Updates - News from DDC-I. May 2003, Volume 4, Number 4 -

[http://www.ddci.com/news_vol4num4.shtml] A monthly news update dedicated to DDC-I customers & registered subscribers.

SCORE-653 for CsLEOSTM RTOS. A strong platform for safety critical software development.

Cost-Effective Windows Migration Package for TADS-i960 Customers. Customers define package parameters to create a least-cost migration path.

On the Front Lines. Meet Harold (Bud) Blum, DDC-I TADS & DJCS (JOVIAL) Product Champion. [...]

From: jc <jcdk@ddci.com>

Date: Fri, 1 Aug 2003 11:14:33

Subject: Real-Time Industry Updates - News from DDC-I

To: S9DK June_July 2003 Online News <jcdk@ddci.com>

DDC-I Online News. Real-Time Industry Updates - News from DDC-I. June/July 2003, Volume 4, Number 5 -

[http://www.ddci.com/news_vol4num5.shtml] A monthly news update dedicated to DDC-I customers & registered subscribers.

DDC-I Releases Updated Sun/Solaris TADS Development System. Providing consistency across the entire TADS product line for all hosts & targets.

Full Windows Support for TADS-68xxx. This successfully completes our migration of the entire TADS product line to Windows. [...]

For this month's complete newsletter, go to http://www.ddci.com/news_current_issue.shtml

Login: - Article on Fractals

*From: lomba <news@pressemicro.net>
Date: Thu, 12 Jun 2003 23:24:23
Subject: Login: de ce mois ci
Newsgroups: fr.comp.lang.ada*

[Translated from French: -- dc]

Just a small message to announce that the article on fractals in "Login:" this month only contains Ada code. This is rare enough to be announced:)

*From: Etienne Baudin <pfox@free.fr>
Date: Tue, 17 Jun 2003 13:48:14
Subject: Re: Login: de ce mois ci
Newsgroups: fr.comp.lang.ada*

I'd like to point out that, following the publication of these sources in Ada, this journal would readily accept a complete article on Ada (concepts, syntax, etc). Hence if somebody is interested (spare time during your holidays, a little money to be made ...) it would really not be bad for the "promotion" of the language.

COTS Journal - 21st Century Ada: Faster, Stronger ... and Higher

From: mcq95@earthlink.net (Marc A. Criley)

*Date: 13 Jun 2003 06:25:46 -0700
Subject: Ada Article in "COTS Journal"
Newsgroups: comp.lang.ada*

The belatedly received May 2003 issue of COTS Journal has a very nice article about Ada, specifically its use in embedded systems. They cite the Atlas V and JSOW, and note that Ada business continues to grow.

This particular issue is `_not_` yet online [...], but considering that I just got the issue (in mid-June), it appears they're running behind and it should show up eventually.

[And from a later message: -- dc]

It's now there:

<http://www.cotsjournalonline.com/2003/05/>

[The article is entitled "21st Century Ada: Faster, Stronger ... and Higher" and is introduced: "Still delivering powerful benefits from its original DoD development as Ada 83, as well as extended advantages developed for Ada 95, Ada continues to propel demanding safety-critical, real-time embedded-system programs." -- dc]

High-Integrity Ravenscar Paper Available On-line

*From: rod.chapman@praxis-cs.co.uk (Rod Chapman)
Date: 24 Jun 2003 10:01:53 -0700
Subject: ANN: High-Integrity Ravenscar paper now available on-line
Newsgroups: comp.lang.ada*

I'm pleased to announce a new paper from SPARK Team - "High Integrity Ravenscar" by Peter Amey and Brian Dobbing. This paper was presented last week at the Ada Europe conference in Toulouse.

PDF is now available on the publications page of www.sparkada.com.

This paper provides the first public preview of what's coming in the next release of the SPARK language and toolset.

Abstract:

The Ravenscar Profile is an exciting development for the Ada community since it provides, for the first time in the history of our industry, support for deterministic, multi-tasking programming as an integral part of a standardized language. Despite its many advantages, the profile leaves several areas where behaviour is implementation defined and can result in run-time errors; this is unfortunate in a profile aimed clearly at the critical systems market. The SPARK language is a well-established sequential Ada subset that avoids ambiguity and allows all language rule violations to be detected prior to execution. The authors show how the principles of SPARK have been successfully extended to encompass the Ravenscar Profile thereby statically eliminating the profile's problematic areas. The result should allow concurrent Ada programs to be constructed with the same degree of rigour that is now possible using sequential SPARK.

Rod Chapman, SPARK Team, Praxis Critical Systems

Embedded Systems Programming - More on Enumerations

*From: prenom_nomus@yahoo.com (Mark)
Date: 5 Jul 2003 16:36:29 -0700
Subject: C++ enumerations the Ada way
Newsgroups: comp.lang.ada*

Interesting article in Embedded Systems Programming this month:
<http://www.embedded.com/2003/0307>.
Look for: Programming Pointers, More on enumerations, by Dan Saks.

Several references to the Ada way of doing/using enumerations and how to do something similar in C++. It might even make a few people look into Ada.

PCQuest - Designing the 777

*From: Carlisle Martin C Dr USAFA/DFCS <Martin.Carlisle@usafa.af.mil>
Date: Mon, 21 Jul 2003 13:38:54
Subject: FW: Article on the Boeing 777 with some interesting stats
To: team-ada@acm.org*

FYI. Some good Ada press from this April.

Martin C. Carlisle, Associate Professor and Advisor-in-Charge, Department of Computer Science, United States Air Force Academy

<http://www.pcquest.com/content/topstories/103040301.asp>

[Quotes: -- dc] Produced starting in 1995. First jetliner to be completely digitally designed. 10,000 people in 238 build teams contributed to the design and its testing. Most of the code was written in Ada. When the first plane was finally assembled the parts fitting error margin was 0.23 inches as compared to 0,5 inches when design was done with mockups.

Ada Inside

Europe - Mars Express & Beagle 2

From: Bill Findlay <BillFindlay@blueyonder.co.uk>

*Date: Mon, 02 Jun 2003 14:45:21
Subject: Re: Ideas for Ada 200X
Newsgroups: comp.lang.ada*

[...] BTW The UK Beagle 2 Mars probe is sitting atop a Soyuz rocket at Baikonur waiting for a launch this evening. It is controlled by an open-source (ERC32) computer programmed in Ada 95 using an open-source (GNAT) compiler. Let's wish it luck, and hope it suffers no unbounded priority inversions (or Constraint_Errors)!

From: Vinzent Hoefler <ada.rocks@jlfencey.com>

*Date: Mon, 02 Jun 2003 15:57:51
Subject: Re: Ideas for Ada 200X
Newsgroups: comp.lang.ada*

> an open-source (ERC32) computer programmed in Ada 95 using an open-source (GNAT) compiler.

Ah, I guess, you mean that:
<<http://www.estec.esa.nl/wsmwww/erc32/freesoft.html>>. Neat. [...]

From: volkert@nivoba.de (Volkert)

*Date: 3 Jun 2003 01:40:21 -0700
Subject: Ada on the way to Mars
Newsgroups: comp.lang.ada*

Yesterday the first European Mars mission (Mars Express) started. Many embedded applications (for example in the Beagle2) are developed in Ada. Has someone more information on this?

From: Vinzent Hoefler <ada.rocks@jlfencey.com>

*Date: Tue, 03 Jun 2003 11:45:38
Subject: Re: Ada on the way to Mars
Newsgroups: comp.lang.ada*

See [the first item above -- dc] and perhaps you should check out <http://www.esa.int/SPECIALS/Mars_Express/index.html> and <<http://www.beagle2.com>> if you haven't already. Unfortunately no mentioning of Ada there, it seems.

[See also "Europe - IPL Supports Beagle 2" in AUJ 21.2 (Jul 2000), p.122. -- dc]

From: volkert@nivoba.de (Volkert)
Date: 3 Jun 2003 11:27:45 -0700
Subject: Re: Ada on the way to Mars
Newsgroups: comp.lang.ada

> Ada 83 or Ada 95?

For the Beagle2 it is Ada 95 with Ravenscar tasking subset.

From: Colin Paul Gloster
<Colin_Paul_Gloster@acm.org>
Date: 17 Jun 2003 15:05:18
Newsgroups: comp.lang.ada
Subject: Re: Ada on the way to Mars

> Many embedded applications (the for example in the Beagle2) are developed in Ada. Has someone more information on this?"

Yes, Beagle 2 has an ERC32 running at about 10 MHz (or it may be 8MHz).

Elsewhere in the mission...

The landing software could be construed to be an example of code reuse because it is using Ada 83 code from the Huygens lander. However, Huygens is not due to start entering Titan's atmosphere until long after its first Martian heir gets into action.

Europe - Space Projects

From: Colin Paul Gloster
<Colin_Paul_Gloster@acm.org>
Date: Tue, 29 Jul 2003 17:45:10
Subject: DASIA 2003 excerpts
To: team-ada@acm.org

At last month's DATA Systems In Aerospace conference Christophe Moreno of Alcatel Space announced in his "Plug & Play Architecture for On-Board Software Components" presentation that he is using Ada 83.

24 minutes later during Astrium's Matthias Wiegand's "Next Generation Avionics System for Satellite Application" presentation, Matthias Wiegand said "Ada is possible".

During the "Software Agents 2: A Minimal Real-Time CORBA ORB for Space" presentation by what Science Systems (Space) had become, it was revealed that the company is working on a minimal realtime CORBA ORB for which Ada 95 support is being developed.

CNES's Francois Bossard spoke about a software architectural model in his "Event driven architecture for hard real-time embedded systems" presentation which has been implemented in Ada. An interface for C in middleware was later added.

A strange part of the conference ...

Indirect Information on Ada Usage

[Extracts from and translations of job-ads and other postings illustrating Ada usage around the world. -- dc]

From: Richard Riehle
<richard@adaworks.com>
Date: Sun, 04 May 2003 18:20:19
Organization: AdaWorks Software Engineering
Subject: Re: employment with ada
Newsgroups: comp.lang.ada

[...] There are people using Rational Rose and Ada together quite comfortably. Also, Rational is not the only game in town. Aonix will soon announce a new set of tools, and they are still very much in the Ada business. Aonix also has a UML tool, Software Through Pictures. I recently talked with the President of Green Hills and learned they are doing really well with the Ada component of their business. Also, they support an excellent inter-language development capability along with a pretty slick development environment. Some of my clients are using Green Hills quite happily.

There are certainly some excellent Ada embedded development options out there. Even GNAT, public as it might be, is currently serving in the embedded marketplace. ICC and DDC-1 both continue to produce good products for embedded systems. If it is not sufficient to be able to select from a least six good compiler publishers, how many do we need? [...]

From: Samuel Tardieu <sam@rfc1149.net>
Date: Wed, 14 May 2003 12:18:12 +0200
Subject: [ada-france] Re: job ada
To: ada-france@ada-france.org

> Hello, I'm looking for employment in Ada development [...]

On the site of the APEC, <http://www.apec.fr/>, there are currently 23 job offers directly relating to Ada.

From: Paulo Urbano <urbano@ias.uni-stuttgart.de>
Date: Wed, 28 May 2003 15:49:32 +0200
Subject: Intel-OA: Sockets sample code
To: <intel-objectada@aonix.com>

In an educational related project, I am trying to implement a system composed by a Flash graphical interface executed in a Web browser and an Ada95 system running on a PC. My idea is to link the events on the Flash GUI with Ada using a TCP connection. For that, I need just a minimum server functionality on the Ada application (wait for a connection, send a reply and close the connection). I am using ObjectAda 7.2.2 [...]

Paulo Urbano, M.Sc., Institute of Industrial Automation and Software Engineering, University of Stuttgart, Germany, www.ias.uni-stuttgart.de

From: Brian Barker <bbarker@ball.com>
Date: Mon, 2 Jun 2003 07:32:41
Subject: Intel-OA: Debug Problem
To: Aonix <intel-objectada@aonix.com>

I'm using ObjectAda 7.2 with the native Windows compiler. I have linked other language object files into my Ada application (C++ and Jovial). [...]

From: robeybd@npt.nuwc.navy.mil
(Brian R.)
Date: 3 Jun 2003 07:25:50 -0700
Subject: AdaCAST, APEX Ada and Clearcase
Newsgroups: comp.lang.ada

We are using VectorCAST's AdaCAST unit testing tool, their newest Release 3.0 in conjunction with Rational Corp's APEX Ada using Clearcase as the version control. The system is a SPARC Solaris Sunblade 1000. [...]

From: Bruce Hennessy
<bhennessy@decilog.com>
Date: Thu, 5 Jun 2003 17:27:03
Subject: Ada software engineers needed
Newsgroups: comp.lang.ada

We are looking for Ada software developers in the New York and New Jersey area to fill immediate positions at various levels. [...]

From: Martin Dowie
<martin.dowie@baesystems.com>
Date: Fri, 13 Jun 2003 08:43:10
Subject: Re: [Ada-Comment] AI request: number_base enlargement
To: Ada-Comment List <ada-comment@ada-auth.org>

[In a thread on "The coding of numbers into shorter strings by means of higher number bases is sometimes employed in practice (e.g. short filenames, etc.). I see no compelling reason to prevent number systems beyond hexadecimal.", the comment was given "This would be very handy for things like digital maps systems and map preparation facilities." Which prompted the question "Can you (or anyone) provide an example demonstrating usefulness for this purpose?" -- dc]

Yes, the digital maps that we (BAE Systems, Avionics Systems Division, Edinburgh) produce use such bases and have done since long before I moved here! They are currently flying in C-130s, Harriers and a whole load of other platforms. [...]

From: Roger Racine
<rracine@draper.com>
Date: Mon, 16 Jun 2003 10:27:55
To: team-ada@acm.org
Subject: Any Course Slides Available?

I have volunteered to give a short (half day at most) presentation on a number of Ada95 topics to the members of a project using Ada. The team is starting with existing code (Ada83 code that has already been ported to Ada95) and making modifications needed by the new

project. The team has been in place for a bit, and the members have picked up the basics already, but they would like to have a seminar on some topics that are not so easy to pick up, were not used in the existing software, or were just thought to be needed by the team leader. [...]

Roger Racine, Draper Lab.

*From: Robert Airley <roberta@cisny.com>
Date: Mon, 23 Jun 2003 09:47:56
Subject: new Ada job
To: team-ada@acm.org*

We are looking for a Software Engineer for our Defense client based in Clifton, NJ. They must be experienced in complex real time embedded software Development. A background in weapon system Development preferably with experience in electronic warfare. Must be skilled in Ada or assembly language. Must have secret clearance. [...]

*From: Björn Lundin
<bjorn.lundin@swipnet.se>
Date: Sat, 28 Jun 2003 20:03:34
Subject: Segmentation Fault on Aix
To: GNAT Discussion List
<gnatlist@lyris.seas.gwu.edu>*

I've compiled gnat 3.15p using gcc 2.8.1 on Aix 4.3.3.

I'm interfacing to Oracle 8.1.7 in my processes. There are about 20 processes in my system, and they communicate via System V shared memory and semaphores. [...] (It's a crane controller for a warehouse management system, and uses one task per crane). [...]

*Date: Mon, 28 Jul 2003
Subject: Posted on job-scareer.be website*

4 Ada software engineers [...]

Function description: You will be integrated in a strategic project including architectural & detailed design, development of the application, programming, testing and writing of the design documentation.

Profile: [...] Knowledge of a programming language preferably Ada 83-95 or C++ [...]

On Creating More Ada Jobs

*From: crebralfix@angelfire.com (tom)
Date: 1 May 2003 17:36:30 -0700
Subject: employment with Ada
Newsgroups: comp.lang.ada*

I've been reading and learning about Ada due to some ... annoyance ... with C++. The first computer programming book I ever read was Booch's "Software engineering with Ada" (got it for \$2) and I've liked the language ever since. Now, it's 4 years later and I'm looking for a programming job.

My question is: what strategies do you use to successfully obtain work where you

can program in Ada on a regular basis? [...] What advice do you have for someone that is new to the language and fairly junior as a programmer?

*From: Marin David Condic
<mcondic@acm.org>
Date: Fri, 2 May 2003 08:08:09
Subject: Re: employment with Ada
Newsgroups: comp.lang.ada*

Realistically, most programming positions aren't going to involve Ada and the programming language is only one aspect of the job. Most people are going to get the best job satisfaction if they are working for a company that gives them interesting work in a field they enjoy and located in an area they like. So pick the part of the world you want to live in, find companies there doing something you think is important and interesting and don't worry too much about what language their major software products are written in.

Consider also that you might find a position where the product you are working on is in some other language. You probably won't be able to get anyone to convert that product into an Ada product, but that doesn't mean you can't use Ada on your job. Almost always, you can "invent" your own job if you really want to. Your "real" work may involve maintaining some big C++ system, but inevitably, you will see a need for some sort of support tool or add-on or process improvement aid. Get yourself the GNAT compiler and GtkAda and whatever else you need and build that in your "spare time". (Put in some free overtime, steal some hours when there is some slack time, etc.) When you get the thing built you take it to your boss and say "Look what I did - would others in the organization like to use it?" Soon you find yourself with full time work maintaining some sideline project in Ada.

Another way to invent your own job is with a little entrepreneurial spirit. Come up with an idea for a product that is either all software or has software as a major component. Build a prototype using Ada. See if you can get others interested in it. If it is useful and marketable, you might just find yourself becoming the next Bill Gates.

The main point is that if you wait around for some big company to see the sweet light of reason and switch to programming in Ada, you'll likely have a long wait. You need to create things in Ada in order to have work in Ada. Good luck with the job search!

*From: anisimkov <anisimkov@yahoo.com>
Date: Sat, 03 May 2003 12:03:06
Subject: Re: employment with Ada
Newsgroups: comp.lang.ada*

[...] Others have to invent Ada job on their own, it is harder, but possible. If the company does not have a strong programming policy (about language to use), then some inventive programmer could move the software development to the Ada way.

I'm going this way for about 3 years. Now the company client-server internet product has 4 executable files written in Ada for the server side. The web page about system architecture <http://www.actforex.com/architec.html> says nothing about Ada, but talks about "Java" (3 applets written in Java). Everybody knows that Java is "cool" ;-. I'm thinking about to ask management to write there some notes about Ada usage.

*From: Anders Wirzenius
<anders.wirzenius@pp.qnet.fi>
Date: Sat, 03 May 2003 07:07:47
Subject: Re: employment with Ada
Newsgroups: comp.lang.ada*

I am in pretty much the same situation. The core business of the company is not software development - hence, there is no formal programming policy. I use Ada for developing internal tools for my nearest co-workers. [...]

*From: Marin David Condic
<mcondic@acm.org>
Date: Sat, 3 May 2003 10:44:47
Subject: Re: employment with Ada
Newsgroups: comp.lang.ada*

[On the remark that due to the need for a security clearance for many USA government jobs, "it is easier for a USA citizen to find an Ada job in some company." -- dc]

Only in so far as one is talking about government related work - and that isn't necessarily all done in Ada anyway. It certainly is one of the places with a higher density of Ada projects, but it is also not the only one. Ada is used in commercial applications, but you kind of have to look around for it. People don't generally advertise that they do their software in a specific language. Except, of course, when trying to hire programmers with appropriate experience.

> If the company does not have a strong programming policy (about language to use), then some inventive programmer could move the software development to the Ada way. I'm going this way for about 3 years.

Good for you. That's exactly how Ada is going to find a commercial base. If people develop things in Ada when the company doesn't care what they use, or if it sneaks in the back door or if it is done as a speculative venture, you'll see more growth in Ada jobs. Make useful software in Ada and it will create its own market.

Ada in Context

Ada vs. Graphical Languages

From: Stephen Leake

<Stephe.Leake@nasa.gov>

Date: 07 May 2003 09:35:53

Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)

Subject: Re: employment with ada

Newsgroups: comp.lang.ada

> There are tools out there that want to create the code for you (MatrixX, SCADE, etc).

They don't "create code"! They let people write code in different languages, using a partially graphical syntax (UML tools work the same way; UML is a language). Then they translate that code into an earlier language that can be compiled for an actual machine.

This is a traditional path to developing a new language. I believe the very first C++ "compiler" (Cfront?) translated C++ into C. The Gnu compiler still translates high-level languages into assembly code; that is not called "creating code".

> Are these tools any good?

Some of them are useful, for people who like the graphical languages, because they are similar to the languages used in their domain (control flow diagrams for control systems, for example).

However, the graphical languages are often poorly defined. Even UML 2.0 is not completely defined in terms of being able to generate code; each vendor has to add some semantics and make some architecture decisions. The best vendors give control over the missing parts to the user, via extensions to UML.

I don't like them, because I have found that Ada is the best language for my domain (embedded hard real-time control systems). In large part, that is because the language is clearly defined.

[See also "On UML and Graphical Languages" in AUJ 24.2 (Jun 2003), pp.105-106. -- dc]

Choice of Language Does Matter

From: Wesley Groleau

<wesgroleau@despammed.com>

Date: Tue, 24 Jun 2003 17:48:37

Subject: Re: Boeing and Dreamliner

Newsgroups: comp.lang.ada

> I was merely responding to the statement that one should be sued for using C++.

In some application domains, using C++ should be criminal. The fact that Ada is not perfect will NEVER justify using

something that's worse when lives are at stake. [...]

From: snarflemike@yahoo.com (Mike Silva)

Date: 25 Jun 2003 11:00:48 -0700

Subject: Re: Boeing and Dreamliner

Newsgroups: comp.lang.ada

> C++ isn't worse. It's different.

Here's a report that reaches a different conclusion:

``...results of the UK Ministry of Defense's own retrospective IV&V program that was carried out by Aerosystems International at Yeovil in the UK. It should be remembered that the code examined by Aerosystems had already been cleared to DO-178B Level A standards, which should indicate that it was suitable for safety-critical flight purposes.

Key conclusions of this study follow:

Significant, potentially safety-critical errors were found by static analysis in code developed to DO-178B Level A.

Properties of the SPARK code (including proof of exception freedom) could readily be proved against Lockheed's semi-formal specification; this proof was shown to be cheaper than weaker forms of semantic analysis performed on non-SPARK code.

SPARK code was found to have only 10 percent of the residual errors of full Ada; Ada was found to have only 10 percent of the residual errors of code written in C. This is an interesting counter to those who maintain that choice of programming language does not matter, and that critical code can be written correctly in any language: the claim may be true in principle but clearly is not commonly achieved in practice."

from <http://www.sparkada.com/downloads/Mar2002Amey.pdf>

Unless you are prepared to demonstrate that C++ is 10 (ref. full Ada) to 100 (ref. SPARK Ada) times safer than C, the only reasonable conclusion is that C++ is indeed "worse." BTW, I imagine that the C code in question was already based on a "safe subset" of the C language, so that's what you'd need to show improvement over.

[See also "Programming Language Choice Matters" in AUJ 24.2 (Jun 2003), pp.100-101. -- dc]

A Proof of Ada's Strength

From: Thierry Lelegard

<thierry.lelegard@cptchno.com>

Date: Wed, 25 Jun 2003 10:08:58

Organization: Canal+ Technologies

Subject: Re: Boeing and Dreamliner

Newsgroups: comp.lang.ada

Isn't it a satisfaction to be obliged to go 10 years backward to find an big failure where Ada was involved (I mean *involved*, not responsible)?

In the mean time, thousands of Windows systems are crashing every day around the world. That makes a few millions Windows crashes since the Ariane 5 crash. Millions of MS Word crashes (resulting in thousands of corrupted and unusable documents, and among them a few of mine!). How many "unavailable" ATM, due to software crash? "Unavailable" Web servers? Etc...

So, on a software point of view, the Ariane 5 crash is not a proof of Ada's weakness. Having *only* this event to speak about is a proof of Ada's strength.

On Language Quality and Reducing Reuse Risk

From: Richard Riehle

<richard@adaworks.com>

Date: Fri, 27 Jun 2003 10:15:59

Organization: AdaWorks Software Engineering

Subject: Re: Boeing and Dreamliner

Newsgroups: comp.lang.ada

Since this discussion began as a dialogue about the Boeing 7E7, and someone raised the question of whether C++ would be appropriate for software on that aircraft, the lesson of Ariane 5 is important for the engineers. [...], we have the issue of reuse, as noted by Hyman Rosen in his mistaken appraisal of the Ariane 5 failure. I and others have commented on this earlier.

If Boeing does decide to use Ada, and we would hope they would, the lessons of Ariane 5 are valuable. Those lessons indicate that, even when using superior technology, one can make other engineering decisions using incomplete data.

C++ would be a dangerous choice, not only because the language itself can lead to so many undecidables and unpredictable fragments of code, but also because the language, itself, implies a heavy reliance on reusable components. Frankly, I have greater confidence in the savvy of Boeing engineering management and would expect them to have learned the lessons of Ada from the B-777, along with the lessons being learned in the on-going upgrades (in Ada) of software for the B-757, B-747, and B-767.

As far as I know, there is no DO-178B compliance inherent in C++. One can comply with DO-178B using a carefully selected subset of C. Even in Ada, one must take care to apply the appropriate pragmas from Annex H, apply the constraints of Ravenscar or SPARK, and avoid certain low-level features of the language a less experienced engineering might be tempted to engage.

C++ might be appropriate for certain systems such as cabin entertainment, but it would be a serious error in engineering management to choose it for any of the

safety-critical software. The more I see of C++, the more experience I gain with it, the more I realize why Ada is designed to a more rigorous set of rules. Those rules may be annoying to some programmers, but those rules make sense to an engineer.

A fly-by-wire aircraft is an engineering problem, not a programming problem, even when software (and programming) are part of the solution space. When one looks at this kind of system as a total engineering effort, one must also consider the software as part of the engineering, not separate from it. With C++, it is too easy to disengage the software effort from the rest of the engineering effort.

The argument that one cannot find trained and experienced Ada programmers is one of the most bogus arguments proposed by military and civilian contractors.

We are looking first for engineers. In my experience, good engineers, when exposed to Ada, do learn to create excellent software designs, and they learn to do so independent of the the search for the perfect algorithm. Often, it is better to start with engineers and teach them Ada than to start with programmers who have already developed bad habits. I see lots of C++ programmers who have to be re-educated to think as engineers when given problems in embedded systems environments.

I have trained engineers to program in Ada and they take to it well and understand the underlying rationale for its design. I have trained C++ programmers and many of the spend their time arguing about how they can do such-and-such in C++ and why can't they do it that way in Ada. We can train experienced programmers in Ada, but we need to first train them to think like engineers. It seems that, many engineers grasp the reasons for Ada's design quickly. Those same engineers are not focused on resume-building, but on problem-solving. They realize that Ada is an excellent tool for solving engineering problems.

For the past three years, I have been teaching Ada at the Naval Postgraduate School. My students take Ada As A Second Language. At the end of each Quarter, I require them to write a paper comparing Ada with their first (or other) language. They often express their preference for Java (rarely for C++), but most of them understand the difference between using Ada for dependability and Java for ease of creating screens and little GUI programs.

I believe that the Boeing engineers also understand this. They are building software where life and safety are at stake. When one objectively examines the current choices in software engineering languages, Ada continues to be the most appropriate choice when one is concerned with high dependability. Let's hope I am

right about those Boeing engineers. They have shown good judgement in the past in making software decisions. They will probably continue to do so in the future.

From: aek@vib.usr.pu.ru (Alexander Kopilovitch)

Date: 27 Jun 2003 19:18:51 -0700

Subject: Re: Boeing and Dreamliner Newsgroups: comp.lang.ada

> C++ would be a dangerous choice, not only because the language itself can lead to so many undecidables and unpredictable fragments of code, but also because the language, itself, implies a heavy reliance on reusable components.

This observation is both true and important, I think. The implicit link between C++ language and cheap/easy reusability really exists. This is because C++ was designed and evolved primarily for solution space. [...]

Alexander Kopilovitch, Saint-Petersburg, Russia

From: Richard Riehle

<richard@adaworks.com>

Date: Sun, 29 Jun 2003 11:26:54

Organization: AdaWorks Software Engineering

Subject: Re: Boeing and Dreamliner Newsgroups: comp.lang.ada

Alexander Kopilovitch wrote:

> Actually, the SRI software code for Ariane 4 was *perfectly good for Ariane 4*, although it quite probably was not so good in general [...]

Hyman Rosen wrote:

> The Ariane 4 programmers exactly reproduced the Y2K problem in microcosm. They wrote code that took advantage of limited input range, and sent it out into the world with insufficient protection against the future. It was perfectly good in the same way as two digit years.

A key difference between designing in Ada and many other languages is the use of problem-space constraints. For example, a designer might declare an integer type such as, "type Number is range 12..451;" if the problem under consideration called for that kind of constraint. If, at some later time, I decide to use the solution bounded by that constraint to solve a problem that requires a different set of constraints, I will have made a mistake. The mistake, in that case, is mine, not that of the previous designers.

What we often encourage, for Ada designs, is that the algorithmic details be independent of some particular set of constraints. Generic components are sometimes useful for this. For example, generic

type Num is range <>;

function Compute(Data : Num) return Num;

where the internal algorithmic construct will behave exactly the same way on every instantiation of Compute after associating the generic formal parameter with a generic actual parameter.

[...], the issue of specifying constraints that precisely map the solution-space to the problem-space continues to be a useful feature of the language. If I accept this constraint during development, many potential problems will be identified early in my process.

If, on the other hand, I choose to bypass the language safeguards and use unchecked features of the language, I am putting my entire design at risk. This is one aspect of the Ariane 4 software that contributed to the Ariane 5 event. The developers of the Ariane 4 software took the trouble to ensure that the unchecked operations were appropriate. The engineers on Ariane 5, many of who were the same people from Ariane 4, failed to evaluate the potential consequences of those same unchecked operations.

Is this a failure of the language? One might suggest that the option of unchecked operations in Ada is a language problem. However, we must also recognize that the language clearly specifies that unchecked operations are "unchecked" by the compiler. In most languages, such unchecked operations are the default, not an option. For example, in the C family of languages, automatic type promotions rarely present any kind of warning to the programmer. In Ada, the programmer must bypass the normal rules of the language to achieve the same result.

So, the software design for Ariane 4 was exactly right for Ariane 4. It was not designed for some future system. To suggest that this is the equivalent of Y2K is interesting. It re-raises some issues related to the original subject of this thread, that of the Boeing 7E7 and Boeing 777. And those issues directly support the folly of even thinking about using C++ for this aircraft. They also indicate where caution should be the watchword when transitioning 777 software to the 7E7.

As I indicated in an earlier posting, I am confident the Boeing engineers will understand the lessons of Ariane 4/5 when re-using Ada code from the 777 on the 7E7. I am also confident that, pressures from resume-builders notwithstanding, they will realize the value of using contemporary Ada, with its excellent record for software safety, instead of a language so characterized by unpredictability that they could never be sure that some undetected behavior might manifest itself long after even the best of testing has been completed.

From: Richard Riehle

<richard@adaworks.com>

Date: Mon, 30 Jun 2003 18:14:42

Organization: AdaWorks Software Engineering

Subject: Re: Boeing and Dreamliner

Newsgroups: comp.lang.ada

Hyman Rosen wrote:

> You're talking about a situation where every arithmetic operation in the code was carefully scrutinized. I'm sure that in the cases where protection was left in the Ariane 4 code it did not consist of allowing an Ada exception to be raised on overflow, but rather coding in such a way that a correct numeric result would be produced. I don't see why such scrutiny would not result in equally safe C++ code.

C++ is simply not designed this way. Ada is. In C++ it is perfectly legal to do all kinds of assignment statements where the result is not entirely predictable. A key difference between Ada and many other languages is the absence of structural equivalence in favor of named equivalence. That is, the fact that there might be some possible structural compatibility between objects with different names (or type designations) is never sufficient justification for a legal operation involving types of different names.

Even unchecked operations, when there is a potential for an erroneous result, result in a warning in better Ada compilers. Also, unchecked assignment (Unchecked_Conversion) only permits assignment in the direction for which it is instantiated. This eliminates many kinds of errors.

Quite simply, the amount of error checking performed by an Ada compiler is substantially greater than in C++. This does not mean C++ is evil. It simply means it is inappropriate for software that demands a high level of dependability, especially when one has Ada available as an option.

I really don't want this to sound like C++ bashing. However, the more I learn about C++, the more I realize it is an excellent language for certain classes of problems, but is more dangerous for safety-critical software than Ada. [...] do not use [C++] for software where life and safety are involved. This is simply a matter of selecting the right tool for the right job.

> I obviously disagree with you about the suitability of C++, but I certainly agree that Ada is a fine choice as well. As for resume building, I would just let such people do their building elsewhere.

I know you are a skilled and experienced C++ developer. Does it not occur to you that simple little things such as automatic type promotion, incorrect placement of curly braces, and errors in pointer arithmetic, to name a few issues, might result in questionable code? What about incorrectly defined destructors? Or default

assignment operations? Or many more such entertaining features of C++ that add power, but also add the potential for undetected errors? You know the list is even longer than the one I just presented.

From: Warren W. Gay VE3WWG

<ve3wwg@cogeco.ca>

Date: Fri, 04 Jul 2003 12:42:04

Subject: Re: Boeing and Dreamliner

Newsgroups: comp.lang.ada

I would disagree with your position on the basis that even where code is carefully scrutinized, within Ada you have the advantage of builtin language features to check areas that you might neglect [...]

For example, where a short (16 bit integer) in C/C++ might hold the value -32768, and be negated and assigned to a short result, this operation might be undefined (I am not sure if any newer standard like C99 addresses this). On some implementations at least, that result is silently set to 0, which clearly is incorrect! In Ada, this cannot be ignored without deliberately working around it (or turning the checks off).

[Long response deleted about the possible implementations of the above, and how different cases can be distinguished by checking the values in <limits.h>. -- dc]

So what this says, in the end, is that it is up to the programmer to make these "checks" and to "do the right thing".

This is clearly one specific area that Ada has a advantage in safety. Unless you run/test your code with the checks deliberately turned off, you will discover this little gem immediately (assuming that you encounter the right data, that is). Conversely, in C/C++, this "undefined behavior" may go completely unnoticed, until it has a disastrous side effect somewhere else down the line.

This is the precise scenario I ran into when porting some C code into Ada some time ago. The SOX C code was oblivious to this "error", but the Ada code that I ported caught it. So this was a real example of the C programmer failing to "do the right thing", and Ada forcing _this_ programmer to do "the right thing."

[See also "AdaVox 0.5 - Wave Sound File Player" in AUJ 22.3 (Sep 2001), pp.138-140, and "C developer switches to Ada for Linux development" in AUJ 21.4 (Jan 2001), pp.246-248. -- dc]

Warren W. Gay VE3WWG,
http://home.cogeco.ca/~ve3wwg

Reliable Real-Time Java Not Expected Soon

From: Colin_Paul_Gloster@acm.org (Colin Paul Gloster)

Date: 29 Apr 2003 18:33:56

Organization: Dublin City University (DCU)

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

> Which feature of Ada critical to our [...] development cannot be achieved with C/C++/Java?

Of those people who work as subcontractors to the European Space Agency, two of the most ardent supporters of Java and detractors of Ada (namely Alessandro Pasetti -- who unlike most people whose first choice is not Ada, he does actually understand Ada 83 and Ada 95 and profiles very well and has actually given me valuable support in an Ada project -- of P & P Software (<http://www.PnP-Software.com/>) and Erik Aad Visser of Computer Hardware Een Embedded Software Systems (http://www.chess.nl/chess_alg.php?language=english)) said at the Data Systems In Aerospace (<http://www.Eurospace.org/dasia.html>) conference last year that they expect it would be nearly as late as 2007 before there would be a Java implementation of high enough reliability to use in space (in orbit and unmanned).

As far as the language chosen not being a fundamental detail and only being an implementation issue is concerned, there are actually flexibility benefits or penalties in choosing between different programming languages.

From: john@assen.demon.co.uk (John McCabe)

Date: Wed, 30 Apr 2003 08:09:09

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

As a matter of interest, who is carrying out the implementation of High-Rel Java? Is it Sun? For what it's worth (although this may have changed) the license on the Java SDK explicitly forbade the use of Java on mission or safety critical systems. It's been fairly clear for what purpose Java was originally developed, and I see that as an issue for future High-Rel development.

> there are actually flexibility benefits or penalties in choosing between different programming languages.

Of course - some languages are better suited to some tasks than others. Java is well suited to little boys' toys (e.g. STBs, phones etc), and Ada is well suited to big boys' toys (e.g. aircraft, trains, spacecraft etc)

From: Colin_Paul_Gloster@acm.org (Colin Paul Gloster)

Date: 30 Apr 2003 22:55:01

Organization: Dublin City University (DCU)

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

> I hope they were happy to be named here!

They have published their views and announced them in person too.

> As a matter of interest, who is carrying out the implementation of High-Rel Java?"

There are two vendors, and there are two rival realtime Java consortiums.

> Is it Sun?

They are not Sun.

> For what it's worth [...] the license on the Java SDK explicitly forbade the use of Java on mission or safety critical systems. [...]

Yes. An explicit example given in a Netscape Navigator license forbade it for use on the same workstation as one being used in air traffic management. As for Java, I do not remember if that exact example was also given.

> (although this may have changed)

Highly unlikely.

Java Updates

*From: Lionel Draghi
<lionel.draghi@free.fr>
Date: Wed, 28 May 2003 01:28:51 +0200
Subject: [ada-france] Java 1.5
To: Ada France <ada-france@ada-france.org>*

[Extracts translated from French. -- dc]

> Ada a better means to teach re-use? (I believe Java will allow generics in version 1.5!)

Java 1.5 [will] also introduce a reasonable foreach and management of enumerations. [...] Eight years to have a foreach... Regarding "time to market", what a performance!

Porting Experiences

*From: john@assen.demon.co.uk (John McCabe)
Date: Mon, 28 Apr 2003 08:29:28
Subject: Re: Ada in Iraq
Newsgroups: comp.lang.ada*

> Those who are bullied into using Ada resent Ada. The resentment strikes at Ada, but it is really aimed at the bullies. In my experience, those who use Ada successfully, and there are lots of people who do, persuade themselves of its benefits.

But then there are people like me who are 'bullied' into Ada, resent it, then end up using it successfully and see the light!

My 'blinding flash' was being able to write a simple little utility, compile it on SunOS, have it run correctly (admittedly on the second attempt), then compile and run the same source code on a PC and have it run correctly! Until then I was very sceptical. I knew the utility was needed on SunOS and PC and the only compilers we had for both machines were Ada ones so it had to be Ada.

From: Steve <steved94@attbi.com>

Date: Mon, 28 Apr 2003 13:36:28

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

This reminds me of a discussion I was having with a C++ programmer about porting a C++ application. He said that porting their sizable C++ application only took a few weeks. He thought this was pretty good. As I told him, I ported an Ada application of similar size, it took a few hours (I had to re-write the interface to sockets from Win32 to Linux).

Maintenance Cost Issues

From: rod.chapman@praxis-cs.co.uk (Rod Chapman)

Date: 30 Apr 2003 10:01:00

Subject: Re: Enforcing good software process

Newsgroups: comp.lang.ada

> I was talking about reliability, and about liability. Ford and GM are liable when their cars break; it would be nice if there were more software companies that took the same attitude.

We have delivered software with warranties, and we will continue to do so. Examples are CDIS (part of the London air-traffic management systems), SHOLIS, and the MULTOS CA. In the case of the latter project, four defects were reported in the first year of operation - these were corrected under our warranty at no cost to our client.

Rod, Praxis Critical Systems

*From: Robert I. Eachus
<rieachus@attbi.com>*

Date: Sun, 11 May 2003 23:02:37

Subject: Re: Enforcing good software process

Newsgroups: comp.lang.ada

That reminded me of one of my favorite Ada stories from way back. Honeywell Small Systems Division -- its name changed several times while I was there and it is now part of Bull -- did a study to see which systems programming language should be chosen to replace assembler. Three languages were chosen for further testing C, Pascal, and Ada. Two of six implementation projects for the next OS version all about the same size and complexity were chosen for the study. One of the Pascal projects was shifted to Ada when it was determined during the detailed design phase that concurrency support was needed in the programming language.

The three Ada projects came in on time and on budget as did one of the C projects. The other two projects were over schedule and budget, but that is not the story.

Six months after the OS version containing all these projects shipped, there was a meeting to review the support costs. The OS support manager was a big C fan and dead set against Ada. He

showed lots of charts that indicated that when repairing STARS (bug reports) on the C projects turnaround was twice as fast as with assembler, and the cost per report was a third less. "What about the Ada projects?" he was asked by the division manager.

"Oh, we don't know what it costs to fix bugs in the Ada systems. There haven't been any reported." To this day, I don't think he understands why everyone present started laughing uncontrollably.

Similarly I was involved with one DoD project where "safety critical" was a total understatement. (One joke among the requirements analysis team was that if there was an accidental nuclear war, this system would insure that we won it.)

Several years after deployment, I was copied on an analysis of remaining problems with the system. There was just one. The decision on whether to have the original contractor maintain the software or to do it organically (within the Air Force) had never been made. Under reasons? "No software maintenance has been required." That I could live with.

*From: snarflemike@yahoo.com (Mike Silva)
Date: 13 Jun 2003 15:00:20 -0700
Subject: \$14,000 per bug
Newsgroups: comp.lang.ada*

That's what Nortel supposedly spends on average, according to a comment reported by Jack Ganssle in Embedded Systems Journal, June, p46. He presumes this refers to fielded bugs.

True or not? Can't say regarding the particulars, but does anybody doubt that bugs are getting more and more expensive, as more and more software runs on more and more desktops and devices?

Managers, bean-counters, are you listening?

Bug Rate Experiences with Restricted C

*From: Colin Paul Gloster
<Colin_Paul_Gloster@acm.org>
Date: Tue, 29 Jul 2003 17:45:10
Subject: DASIA 2003 excerpts
To: team-ada@acm.org*

[See also "Europe - Space Projects" earlier in this AUJ issue for other reports from the Data Systems In Aerospace conference. -- dc]

GMV had a presentation on its "CPFPS: Development of a Safety Critical Hard Real Time Distributed Application for EGNOS". The programmers programmed in C. They coded over 120000 lines. They were worried about memory leaks. They found hundreds of memory leaks which had not already been detected in testing. For the project, verification was automated whenever possible, including memory leak analysis.

Hundreds of errors were found at unit level. After fixing these, about twenty errors were found at system level.

MISRA C was not used because of its restrictiveness, but '99%' of MISRA C was used. The GMV presenter claimed that a lot of erroneous C programs can not be detected by Lint. [...]

[See also "Official Recommendations of Ada" in AUJ 22.3 (Sep 2001), pp.165-166, and "C Guidelines for Vehicle Based Software Recommend Ada" in AUJ 19.2 (Jul 1998), p.108. -- dc]

On Languages, Productivity and Quality

*From: Roger Racine
<rracine@draper.com>
Date: Thu, 31 Jul 2003 07:53:44
Subject: Re: DASIA 2003 excerpts
To: team-ada@acm.org*

Last week there was a meeting of The Open Group's Real-Time & Embedded Systems Forum, which works on software standards. A lot of the work is in the area of safety-critical software, and is focusing on Java as a language. The interesting thing from the Ada perspective is that many of the people at the forum seemed (at least from the comments I listened to) to be unhappy that they were not able to use Ada. [...]

I talked to a couple of people about this, and was told that companies that wanted to use Ada were getting rejected by potential employees specifically because of the use of Ada. I asked if that is still occurring, or if it was a few years ago when there were many fewer available people, but no one knew the answer. [...]

By the way, Ben Brosgol gave a quite good talk on lessons learned using Ada for safety-critical applications and did point out that Ada is still alive and being used on new projects. The point I got out of the talk was "The Ada community has solved the safety-critical problems. Why waste time fixing the same problems for Java?" [...]

Roger Racine, Draper Laboratory, Cambridge, MA, USA

*From: Chris Sparks <mrada@catalina-inter.net>
Date: Thu, 31 Jul 2003 06:40:44
Subject: Re: DASIA 2003 excerpts
To: team-ada@acm.org*

[...] If someone doesn't want to use Ada in their profession than they should go elsewhere as far as I am concerned. I want to use Ada and have always wanted to use Ada since 1983. I have been blessed with Ada jobs all during this time. [...]

*From: rleif <rleif@cox.net>
Date: Thu, 31 Jul 2003 09:09:07
Subject: Re: DASIA 2003 excerpts
To: team-ada@acm.org*

Stupidity is the oldest mental disease! Unfortunately, the software profession does not have any means for the determination what constitutes malpractice or imposing the appropriate penalties. [...]

Robert C. Leif, Ph.D., rleif@rleif.com

*From: Roger Racine
<rracine@draper.com>
Date: Fri, 1 Aug 2003 08:12:51
Subject: Re: DASIA 2003 excerpts
To: team-ada@acm.org*

I don't really think the actions of either the contractors or the DoD constitute either stupidity or malpractice. Ignorance and lack of backbone, maybe. [...]

The people at the meeting who gave indications that they were sorry to have to use languages other than Ada were saying (correctly, I think) only that projects are more productive with Ada. It is possible (with a lot of effort) to get defects out of C and C++ code. My company wrote the software for the Apollo program, in assembly code. Our country's strategic missiles have software written in assembly code. I firmly believe that both those systems had (for Apollo) and have (for our missiles) very few defects. But the cost of the analysis and testing for both of those programs were -quite- high. The same is true for C and C++. I finished a C project last year, and am fairly confident in its quality. But I am also quite familiar with the weeks of time it took to find each of a number of bugs that would have either been found by an Ada compiler or given an exception if we had used Ada. So the cost of getting to the end quality would have been lower (which I unsuccessfully had argued 2 years previously; the sponsor wanted C).

So I don't think anyone at that meeting was stupid. The people who did the interviews with prospective employees - might- have been able to convince them to work on Ada projects if they had better knowledge of the benefits. Show hard evidence that the employee would be more productive, and perhaps they would not have had as many rejections. So ignorance of productivity gains could be a problem. [...]

*From: S. Ron Oliver
<sroliver@caresscorp.com>
Date: Fri, 1 Aug 2003 08:24:54
Subject: Re: DASIA 2003 excerpts
To: team-ada@acm.org*

Well, these [productivity gains] have been fairly well documented in a number of different cases. It seems that people who choose a language consider "productivity" a separate issue from language choice. Either that or they simply don't care about productivity.

[See also "Productivity is More than Speed of Coding" and "On Early Adopters and Productivity" in AUJ 24.1 (Mar 2003), pp.34-35, "On Languages,

Productivity and Quality" in AUJ 22.4 (Dec 2001), pp.222-224, and "Measuring Productivity" in AUJ 22.3 (Sep 2001), p.167. -- dc]

Successful Projects Without Prior Ada Experience

*From: John R. Stroh
<strohm@airmail.net>
Date: Fri, 25 Apr 2003 00:34:31
Subject: Re: Ada in Iraq
Newsgroups: comp.lang.ada*

[On the claim that a specific project cannot justify switching to Ada because "there is no in-house large-scale Ada application development experience": -- dc]

Boeing faced EXACTLY those problems on the 777 project. In particular, the brake controller subcontractor had NO Ada experience, and they screamed bloody murder. After Boeing told them "If you don't want to do this project our way, we can certainly find another subcontractor who will", they sat down and learned Ada. Result: even though they had to start late, and even though they had to train people from scratch, and even though they had to throw away their earlier efforts, they STILL came in on schedule and under budget.

General Dynamics faced EXACTLY those problems with F-16C/D development, although they were using JOVIAL J73. They had some limited experience with JOVIAL J3B on F-16A/B, but nowhere near enough to be at critical mass. They also had to contract out development of the JOVIAL J73 compilers: there were none in existence at the time the project started.

[See also "On Finding Ada Experience and Training" in AUJ 23.3 (Sep 2002), p.158, and "Mainstream vs. Competitive Advantage" in AUJ 21.1 (Apr 2000), pp.45-46 -- dc]

*From: Wesley Groleau
<wesgroleau@despammed.com>
Date: Fri, 25 Apr 2003 22:25:17
Subject: Re: Ada in Iraq
Newsgroups: comp.lang.ada*

The naysayers had proclaimed the AN/BSY-2 project was doomed because (among other reasons) of the learning curve for Ada. Five years later, folks were pointing to Ada as the reason for the project's success.

*From: John R. Stroh
<strohm@airmail.net>
Date: Sat, 26 Apr 2003 01:48:15
Subject: Re: Ada in Iraq
Newsgroups: comp.lang.ada*

> Or is it your contention that even people who don't know Ada can build a large successful project with it? If so, I'll start right away!

The brake subcontractor for the Boeing 777 did exactly that. They started behind the power curve, with no previous Ada experience. They were already well underway with their C implementation, figuring Boeing would give them a waiver. Instead, Boeing explained to them that they would either do the project in Ada or Boeing would hire someone else who would. They came in on schedule, under budget, and thoroughly convinced that they'd been seriously wrong in their opposition to Ada.

So, how soon are you going to start your Ada project?

From: John R. Strohm

<strohm@airmail.net>

Date: Sat, 26 Apr 2003 22:11:46

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

> Actually, I find this extremely interesting. It means that "incompetent" programmers can actually produce better by switching from C to Ada (provided the verification of the subsystem was accepted). Hence you don't need "expert programmers" to produce quality software, just the right tool.

Not quite. The brake contractor had competent engineers; they just didn't have anyone with any Ada experience. They had to hit the ground running and learn fast. And they did. They EXPECTED to crash and burn, but they figured they would be able to blame the crash-and-burn on Ada. They were pleasantly surprised when they succeeded instead.

Phil Crosby (of "Quality is Free" fame) once tried to argue with his CEO when the CEO wanted him to deliver zero-defect products. The CEO told him, just as Boeing told the brake subcontractor, "If you can't do it, I'll find someone who can!" Phil said his reaction was "Oh, well, when you put it THAT way..."

> Has this been documented anywhere? It seems to be a good success story.

Start at <http://www.adaic.com> (I think).

[The Boeing 777 story is at <http://www.adaic.com/atwork/boeing.htm>. -- dc]

Ada vs. Mainstream - Rebuttal of Common Excuses

From: John R. Strohm

<strohm@airmail.net>

Date: Sat, 3 May 2003 19:25:53

Subject: Re: employment with ada

Newsgroups: comp.lang.ada

[More on excuses given to avoid doing a specific aeronautical software project in Ada. -- dc]

[Excuse 1: "There's a lack of experienced Ada programmers."]

Oh, really? In about 1980, General Dynamics/Fort Worth Division cranked up the F-16 Multinational Staged Improvement Program, which, among other things, featured new Operational Flight Programs (OFPs) written in JOVIAL J73, for MIL-STD-1750A and Zilog Z8002. At that time, to my recollection, there was one existing JOVIAL J73 compiler, targeting the 1750A, and it was nowhere near production quality, and there were NO compilers available targeting Z8000. Also, there were no JOVIAL J73 programmers available. Nor were there any universities teaching JOVIAL J73. By your reasoning, General Dynamics should not have been able to build F-16C/D at all, yet clearly they did: the airplane has been flying for almost twenty years, and is projected to remain in service for another twenty. How do you reconcile your results with their experience?

[Excuse 2: "People will flee the company if they are trained in Ada."]

How often did that reason show up in exit interviews for F-16A/B firecontrol computer OFP programmers, who were using JOVIAL J3B? How often did it show up for F-16C/D programmers, who were using JOVIAL J73? What happened to all those JOVIAL J73 programmers when F-16C/D switched to Ada in the mid-1990s? [...]

[...] The assumption appears to be that [this company] does not believe in retaining people who are proven performers, retooling and retraining them as necessary. In the 1960s and early 1970s, General Dynamics (and many other companies) did a massive internal retraining program to convert engineers in other disciplines into embedded software engineers, because they had a critical shortage of software engineers when software suddenly became important.

So why, if [they] were in fact intent on retaining people, would those people choose to flee? Possibly, they perceive that [it] is not their employer of choice? Possibly, there are human relations problems?

[Excuse 3: "Code changes will be needed as Ada compilers are dropped."]

When General Dynamics bought the JOVIAL J73 compilers for F-16C/D, they faced this very same set of problems, only worse. THERE WERE NO PRODUCTION-QUALITY JOVIAL J73 COMPILERS AVAILABLE. The ONLY bids to develop compilers came from small business. See Judy Edwards and Barry Mowday's paper on how to buy a compiler from a small business. As a result, GD bought the compilers INCLUDING ALL SOURCE CODE AND BUILD MATERIALS, so that they would continue to be able to build it, on whatever platform.

How does the fact that you are using C change this? Vendors still go out of business, they still discontinue products.

[Excuse 4: "Choosing Ada is a bad business decision for airplane software development."]

Having been at GD/FW during F-16C/D, I am peripherally aware of the risks involved in airplane software development. Part of your analysis had to be a risk assessment, of the relative risk of a Class A mishap caused by a software defect. It is well-known from the industry that Ada vs. C by itself has significant impact on software defect density (Consult Pratt & Whitney for their experience: you buy engines from them). What did your assessment of this factor show, given that the cost of a Class A mishap is many millions of dollars for an airplane, and about the same for the pilot?

[See also "The Money Argument" in AUJ 24.1 (Mar 2003), pp.38-39. -- dc]

COTS and Reliability

From: jimmaureenrogers@worldnet.att.net
(Jim Rogers)

Date: 24 Apr 2003 11:10:00

Subject: Re: Tools vs. Parts

Newsgroups: comp.lang.ada

[Although this message is about military systems, it may apply to many other domains as well... -- dc]

The old "buy or build" issue has been around for a long time. In recent years the issue has become more important to U.S. Department of Defense projects because the Government officials have legislated the use of Commercial Off The Shelf products to save cost. The decision logic assumes that a product good enough for industry is good enough for the military.

In some cases the logic holds, and COTS products are appropriate.

There are other cases where the logic fails dramatically. COTS hardware has never been built to the usage specifications required by the military. These specifications include a very wide range of temperatures, humidity, and altitude. They also include resistance to shock and vibration beyond commercial standards. Finally, military equipment must be sealed against NBC exposure. (NBC stands for Nuclear, Biological, and Chemical).

COTS software is very useful when used the same way it is in industry. COTS software is inappropriate in mission critical systems. Military systems tend to have lifetimes counted in decades, while commercial software tends to have lifetimes counted in months. This means that a military system built upon COTS software will be unsupported by the vendor within 18 to 24 months, while it must continue to operate for 30 years.

COTS products also present a major challenge in reliability predictions. Few commercial vendors will share reliability data about their products, if they even keep such data. One of the factors that should be used to choose between COTS and custom products is reliability. A cheap initial COTS purchase may result in high maintenance and risk costs if it exhibits low reliability. Conversely, an expensive initial purchase of a custom product with high reliability can result in a low overall cost of ownership.

Restricting C vs. Enabling Ada For Reliability

*From: Richard Riehle
<richard@adaworks.com>
Date: Sat, 03 May 2003 12:17:31
Organization: AdaWorks Software Engineering
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

[In response to a report that some aeronautical safety critical software will be developed "in a safety critical subset of C", although they "strongly believe the Ada language to be superior to anything else around": -- dc]

Reading through the list of reasons given [...] I cannot agree with their decision. [...] The focus of the [...] effort is to produce the best quality software possible for the aircraft. Instead, they cobble together a set of restrictions for C, restrictions we can be assured will be ignored over the lifetime of the project. [...]

[This decision] will cost them more in the long run, they will be fighting with quality issues in C they would not encounter with Ada, and the programmers they are trying to retain with C will leave just as quickly if not more so than if they were using Ada.

It is interesting to me that, as I see DoD contractors making the error of moving away from Ada toward inferior technologies, non-DoD organizations are discovering its benefits (some of them are not in the U.S.) and enjoying success with it. This is one more example of the old phrase, "grabbing defeat from the jaws of victory."

*From: DPH <rally2xs@compuserve.com>
Date: Sun, 04 May 2003 09:14:15
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

> [...] Instead, they cobble together a set of restrictions for C, restrictions we can be assured will be ignored over the lifetime of the project.

They're using an automated tool to enforce them, so ignoring them will be difficult. [...]

> It will cost them more in the long run, they will be fighting with quality issues in C they would not encounter with

Ada, and the programmers they are trying to retain with C will leave just as quickly if not more so than if they were using Ada.

If you don't know the particulars of the 172 C language restrictions, nor the tool used to enforce them and check the code for other errors, I don't see how you can say that. I doubt there are any studies [...] comparing the error rates between Ada and their own particular way of doing C. This C strategy may indeed be close enough to Ada in error avoidance to be superior to Ada when considering the stated drawbacks that would be incurred by using Ada. [...]

Dave Head

[See also "Bug Rate Experiences with Restricted C" earlier in this AUJ news section. -- dc]

*From: Richard Riehle
<richard@adaworks.com>
Date: Sun, 04 May 2003 18:20:19
Organization: AdaWorks Software Engineering
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

> They're using an automated tool to enforce [these restrictions on C], so ignoring them will be difficult.

I'm glad you countered this point. It highlights the widespread misconception that Ada is about what one cannot do rather than what one can do. This is a problem not easily overcome when trying to compare C to Ada.

There are many other characteristics of Ada that lead to quality besides those that seem, at first, restrictive. Some of these support what Grady Booch calls, the "ilities." For example, let's consider traceability. Well-formed Ada code tends to be easy to trace from unit to unit, in part because of the strict visibility rules. The fundamental language constructs, from separate compilation, through child library units provide a structural integrity not easily achieved in a low-level language such as C. More trivial, but important features such as named association provide a level of readability that one might be able to emulate with automated tools, but not as well as when they are integrated into the language.

Putting aside the newer features of Ada such as protected types, inheritance, and dynamic binding, since these are not as useful in safety-critical avionics, one cannot ignore the other features of Ada that contribute to a powerful capability for project-level code reuse.

Yes. One can certainly constrain (even cripple) C, so it is a little bit safer than its standard language description. One cannot promote any construct in C so it will correspond to the built-in capabilities an Ada designer or programmer can enjoy. [...]

> This C strategy may indeed be close enough to Ada in error avoidance to be superior to Ada when considering the stated drawbacks that would be incurred by using Ada.

As noted earlier, this is a shortsighted view of the benefits of Ada. It presupposes that Ada is valuable for what it restricts rather than for what it enables. Taking that view, C cannot begin to approach Ada for what it enables. Also, even with automated error avoidance, it is difficult to take a language in which the default is "unsafe" and make it more safe. As for the "stated drawbacks" I have commented on these earlier and consider them wrongly concluded. [...]

*From: Dennis Lee Bieber
<wlfraed@ix.netcom.com>
Date: Sun, 04 May 2003 15:07:27
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

> How is [a set of restrictions for C] different from SPARK?

Or Ravenscar?

Within a year, I predict, the programmers saddled with this tool to restrict C will all be griping and looking for ways around it because it objects to all the code they are used to writing.

That, or management decides that too much time is spent running source files through what is essentially the parser/semantic analyzer of a compiler for a "safe-C" before it goes to the real C compiler that they decree the tool will not be used, and programmers are to be trusted to follow the restrictions without needing a check program. (Anyone remember the days of Ratfor, Mortran, etc. -- all meant to make FORTRAN IV meet the style of "structured coding"? My college used on for one year -- until the department got the bill for the computer time! One session of preprocessed FORTRAN ate up as much CPU as a full year of COBOL.

*From: Richard Riehle
<richard@adaworks.com>
Date: Sun, 04 May 2003 18:24:43
Organization: AdaWorks Software Engineering
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

> How is [a set of restrictions for C] different from SPARK?

Excellent question.

When one looks at Ada (or SPARK) as a tool that simply constrains what can be done rather than tool that enables things to get done, there is no difference at all.

Sadly, too many people take this restrictive view of Ada. It is far more than that. In another post, I make a few points about this, but to keep this short, let me emphasize that Ada is not about what one

cannot do but what one can do. It is what one can do that is important.

*From: William J. Thomas
<wjthomas@wcv.com>
Date: Mon, 5 May 2003 00:07:57
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

> [...] The safety critical subset of C is C with 172 restrictions, augmented by a source code analyzer to look out for problems. [...]

C with 172 restrictions, what does that leave about 5 features and a "goto" statement? Talk about being stuck on a job with a dead language, gee won't those programmers be worth their weight in gold.

Compared to Ada, 'C' has always been restricted. In 'C' I have that incredibly heavy weight abstraction mechanism, the function, Oh and let's not forget that other architectural powerhouse, the file. In Ada I actually have to choose which features I'm going to bring to bear on my architecture. The features I get in Ada (which are enforced by gee, a compiler) I will have to achieve and enforce outside of the 'C' language. And with what, code checking tools, methodologies, UML, please..., any of us with a few gray hairs knows just how, with lots of cheap 'C' programmers and plenty of debugging time.

The arguments that I've heard against Ada are the same arguments I've been hearing for 20 years. First of all anyone can learn Ada, and with a little in-house training you can actually bring most developers around to your way of thinking. Second, since 1986 good Ada compilers have always been and always will be available. And third, well, who cares, [they have] made a decision and they are going to have to live with it.

I just hope [this] project gathers some good metrics, and then I hope someone eventually compares them to the F22 project!

Differences between SPARK and MISRA-C

*From: Chad R. Meiners
<crmeiners@hotmail.com>
Date: Sat, 10 May 2003 16:29:02
Organization: Michigan State University
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

SPARK is an excellent (my opinion, I am not affiliated with Praxis) language that allows you to practice a correctness by construction methodology which is backed up by powerful proof techniques. In SPARK's case the restrictions are helpful since they make the proofs more tractable. If you read through the SPARK rationale it will explain why almost every restriction is in place.

So when asking how this differs from a restricted subset of C, I would speculate that the restricted C is MISRA-C. Praxis appears to have some information on the differences between SPARK and MISRA-C at <http://www.sparkada.com/assent.html>

*From: Hyman Rosen <hyrosen@mail.com>
Date: Sun, 11 May 2003 03:32:30
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

I wouldn't call it "information" if it's the PDF slide show that we're talking about. If MISRA-C were a person, then it would be an ad hominem attack - there's a lot of name calling, but no information about actual problems. It's just the usual C-hating screed.

On the other hand, the sample set of rules for MISRA-C seem pretty silly to me - they mostly boil down to "program in C correctly" and many of them are checked by ordinary compilers.

*From: Chad R. Meiners
<crmeiners@hotmail.com>
Date: Sun, 11 May 2003 00:25:32
Organization: Michigan State University
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

[...], but C/C++ are unsuitable for practical formal analysis. Formal analysis was the context of the pdf slides so the attack on C is not an ad hominem type attack. This would not be the case if a similar claim was made for not using C to write a text adventure game. To be fair to the slides they did give some reasons (ambiguity, and non-standardizable tools).

*From: Chad R. Meiners
<crmeiners@hotmail.com>
Date: Sun, 11 May 2003 19:04:56
Organization: Michigan State University
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

> And so is Ada [unsuitable for formal analysis], since SPARK gets there by doing things like eliminating pointers.

Yes Ada has some ambiguous features that make formal analysis difficult, but fortunately Ada can be subsetted into well defined languages like SPARK.

*From: Robert A Duff
<bobduff@shell01.TheWorld.com>
Date: 11 May 2003 11:29:43
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

> If MISRA-C were a person, then it would be an ad hominem attack - there's a lot of name calling, but no information about actual problems. It's just the usual C-hating screed.

There is one such "ad hominem" type statement -- it quotes somebody saying, "MISRA C is a shack built on a swamp!". Anonymously.

But the stuff after that seems pretty well reasoned and backed up by evidence. For

example, they point out that the rules of MISRA C are ambiguous, all the tools interpret them differently, and it's not feasible to check the rules by hand. Why is that a "C-hating screed"?

*From: Hyman Rosen <hyrosen@mail.com>
Date: Sun, 11 May 2003 17:14:34
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

They complain about MISRA-C being ambiguous because C itself is ambiguous. I don't know exactly what that is supposed to mean - probably things like integer sizes being implementation dependent and such, but they don't say. Then they talk about how you should be afraid of being sued if you use C for your implementation.

Trust me. To anyone who is not already emotionally invested in Ada, this sounds exactly like the whiny sort of C bashing that makes anyone who hears it immediately distrust the source. [...]

*From: rod.chapman@praxis-cs.co.uk (Rod Chapman)
Date: 11 May 2003 12:24:13
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

I wrote and presented the slides in question, and I'm very happy to answer any questions about their content, either personally or in comp.lang.ada.

Rod Chapman, Praxis

*From: rod.chapman@praxis-cs.co.uk (Rod Chapman)
Date: 12 May 2003 00:20:47
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

Hyman Rosen <hyrosen@mail.com> wrote

> Well, my primary question is how you expect to convince anyone except for the already converted when you start talking about "a shack built on a swamp".

That quote actually comes from a prominent member of the C community, not from me, and was used with his permission. It is indeed a slightly tongue-in-cheek assessment of the language, and I presented it as such. Perhaps this doesn't come across clearly enough in the slides alone.

> The PDFs talk about C ambiguity. What exactly is this supposed to mean, and why is it a problem?

C (like Ada) has plenty of ambiguity in its definition - to use the Ada terminology, these are the implementation-defined, implementation-dependent and erroneous bit of the language. ISO C90 lists about 205 of these I think.

> On the other hand, the sample rules for MISRA-C seem to just be rules of correct C as opposed to limitations on

> the language. I assume that there are actually some rules which do impose constraints.

Absolutely - some of the rules limit the use of casts and pointer arithmetic, for example, which are very important constraints.

> The ASSENT product pages claim that they have a complete checking environment for all the rules which can be examined by source, but you seem to doubt that this can be true without explaining why.

Owing to the ambiguity of the definition (both of C and of the MISRA rules), the notion of "compliance" is very hard to pin down. Some of the rules also require very deep analysis (MISRA rule 30, for example, effectively requires global inter-procedural data-flow analysis which is very hard in the presence of general pointer effects and aliasing.) In the study conducted by Pi Technology, some tools were found to attempt a deep analysis (but are slow) and some tools do a less deep analysis but are faster. Which tool is correct? Which would you buy?

> Are there multiple implementations of the SPARK Examiner from different companies? If not, criticism of multiple MISRA-C examiners for producing somewhat different results is at least a little disingenuous.

SPARK is deliberately and carefully designed so the question "Is this program legal SPARK?" always has a "Yes/No" answer that is decidable in polynomial time. (The latter point is important - trying to solve NP-Hard problems in static analysis is bad news if you want to encourage constructive use of the technology...)

So... No - no other company has chosen to compete with us in tool support for SPARK at present. BUT... since SPARK_does_ have an unambiguous definition, it should be possible for such a diverse implementation to be constructed and for it to give identical results to the existing SPARK Examiner.

(Actually - now I think of it - there is such a tool, but it isn't publically available. Still - its existence illustrates the point.)

On the Effort Needed to Master a Programming Language

From: Richard Riehle

<richard@adaworks.com>

Date: Fri, 18 Apr 2003 11:55:26

Organization: AdaWorks Software Engineering

Subject: Re: the Ada mandate, and why it collapsed and died

Newsgroups: comp.lang.java.advocacy, comp.object, comp.lang.ada

[In response to a discussion whether Ada is an "easily understood language": -- dc]

I suppose I should interpret that as a response to my contention that Ada is not as hard to learn as many have suggested. In fact, Ada, once some fundamental ideas are understood, is not that difficult. However, in its early days, when those fundamental ideas were new, even revolutionary, they were difficult for even experienced programmers to grasp. Now, so many of those ideas have been adopted in the design of other programming languages, that, when one mentions one of them in the context of Ada, the response is often, "So what? Language XYZ has that feature too."

True, not all of the ideas have been adopted. The most difficult idea for many programmers to grasp is the set of visibility rules in Ada. In my experience, once those are understood, everything else falls into place. Some Ada programmers spend years fighting the visibility rules, and continue to raise questions, debate, and engage in endless language lawyering in an attempt to subvert, thwart, or whine about them.

[In response to a remark that "Ada is conceptually an extremely difficult language to understand": -- dc]

I disagree. Ada is not conceptually difficult to learn or understand. It does include some new ideas, and new ideas are sometimes difficult to grasp at first. That is quite different.

Again, many people who were required to learn Ada for their work were resentful of the mandate and spent their time inventing reasons why they believed they could not do something instead of discovering how easily they could do it.

If a programmer wants to fight the visibility rules, for example, he can do so, can grumble about them, find reasons why writing a particular program is so hard, and so on. This attitude reminds me of my first experience on a part-time high-school job in which I was required to use a floor-buffer. The standard floor-buffer uses a large circulating brush which, as it polishes the floor, can exert a powerful thrust vectored over its clockwise motion. A human, regardless of their strength, cannot overcome the force of the brush using brute strength. That same human, learning to gently balance the rotating brush, can make it go anywhere. I became quite skilled at controlling the buffer and could eventually command it using a single finger.

The Ada visibility rules are a powerful force in the language design. The programmer can exert all kinds of energy in complaints about it. However, once one understands this feature, takes charge of its inherent force, and incorporates it into a design and corresponding source code,

it makes the Ada programming (and especially the maintenance) process much more pleasant than one might find in competing programming languages.

Most of Ada is pretty simple. Besides the visibility rules, there are two other parts of the language that are a bit difficult for some newcomers: accessibility rules (for objects and access types) which prevent many of the pointer related mistakes so common in many languages, and tasking, a feature designed to incorporate concurrency in the language.

Most of the other features of the language, such as type safety rules, separate compilation, type derivation, object aggregates, abstract data types, control over defined scope, generic templates, exception handling, packaging of common entities, information hiding, encapsulation, etc; these features that once seemed so strange to programmers experienced in procedural abstractions, are now commonplace throughout the programming language community. When introduced in Ada, they seemed strange, even bizarre. Now they are accepted practice.

Even so, Ada is not intended for one-person, single program, projects where an individual programmer can grasp every detail of the program. It is intended for large-scale projects where many programmers and designers must interact over a set of complex requirements and integrate those requirements into a single, well-functioning whole. Ada makes that kind of project easier to develop, easier than one might experience in most other languages.

One can make the case that Ada is not the best choice for certain classes of software. I still believe it is not appropriate, using the current compiler technology, for eight-bit microcontrollers (e.g., I-8051), but a subset of Ada might be. Smalltalk is still a better language for some kinds of software, especially where one needs a lot of dynamic properties and frequent modification of the code. Smalltalk is, however, a terrible choice for the kinds of software Ada is targeted to build.

There are other languages one can compare to Ada that also have serious potential for embedded-real-time safety-critical software. As I examine them, in some detail, I keep discovering that they simply fall short. It is said, "The Devil is in the details." C++, for example, when I look at it closely, seems inappropriate for developing software of the kind I just described. Each time I look for a competing technology, when compared to Ada, it simply falls short.

A better Ada could be designed. There are things about the language that could be improved. I hope someone will eventually design a language that is better than the current Ada design. At this point, though,

when I am looking for a language that has a high regard for reliability and dependability, Ada still has the edge.

From: Richard Riehle

<richard@adaworks.com>

Date: Tue, 22 Apr 2003 22:57:40

Organization: AdaWorks Software Engineering

Subject: Re: the Ada mandate, and why it collapsed and died

Newsgroups:

comp.lang.java.advocacy.comp.object.comp.lang.ada

> I think nobody complained about any particular feature. The major complaint was that [Ada 83] had all the itsy-bitsy little ideas that anybody had thought at the time. And no truly innovative orrevolutionary ones.

There were a few itsy-bitsy ideas that were unique in the language at the time of Ada's introduction.

> Take something trivial, e.g. string slicing. Not a terribly useful feature, but not entirely useless either. The trouble is, there was a whole lot of such mildly useful features that Ichbiah had seen somebody use somewhere (this one from PL/1 I think), or was able to think up.

String slicing was unnecessary, still is, when one used a Strings package. In contemporary Ada, this is handled quite nicely with language-defined packages.

However, aside from that, the interesting and powerful idea in Ada was not the string slicing, but the array management model. A string, in Ada is a first class type. It is a kind of array type. Arrays in Ada are quite powerful. This is one of the features I most like about the language at the algorithm design level. From my experience with C and C++, I would much rather program with Ada arrays than with C or C++ arrays.

> Also, Ada proponents made much of its strong typing. But it was not at all novel or revolutionary. It was a very common language idea at that time, and then-popular Pascal had it too.

We should probably re-phrase this phrase "strong typing" to type-safety. Even though we saw some movement in the direction of type-safety in Pascal, PL/I, and other earlier languages, Ada went well beyond those earlier attempts and actually achieved what they did not.

> That wasn't bad by itself. But from a practical point of view, over-emphasizing strong typing made interfacing to C a virtual necessity for projects of any significant complexity.

Well, this, of course is dead wrong. Type safety did not drive the move to interfacing with C. Also, Ada was, and is, designed to permit a programmer to get to a very low-level of programming, even to

the machine code level, if it seems appropriate.

One can, in Ada, relax the type safety as necessary. Some organizations set up rules that prohibited using such features, and opted, instead, for writing the code in C. In such cases, they gained nothing, lost a lot, and often created a mess. If they had remained with straight Ada, they would have been better off.

That being said, there is a lot of operational code written in C, C++, Fortran, Assembler, and other languages. Contemporary Ada is designed to make it easy to use existing code regardless of how it is coded. Mature code that works is often better than new code that has not been tested in the real world, regardless of what spiffy new language one might use. Ada allows the developer to leverage the best of existing code while building better and more reliable code than would be likely in C or C++.

> (C and later C++ style of type-safety turned out to be good enough for real-world projects.)

Once again, dead wrong. C++ is not type-safe. It offers an illusion of type-safety under some circumstances, but such oddities as type-promotion (also in Java) fly in the face of type safety. There are all sorts of holes in C++ that make the idea of type-safety there somewhat at odds with what it ought to be.

> But the huge collection of mostly mildly useful features made the language too large and complex, which is what got Hoare and other lights of the time (relatively unknown in the DoD circle) upset with Ada.

Actually, Hoare later wrote a tribute to Ada in one of the Ada 83 books. His earliest criticism of Ada had to do mostly with the inclusion of exception handling in the language. Oooops! We now have exception handling in C++, Java, Eiffel, etc. So when you mention Hoare and Ada, you need to understand that, subsequent to his Turing lecture, he had some pretty nice things to say about the language. [See also "Tony Hoare on Ada - a Quote" in AUJ 22.4 (Dec 2001), pp.230-231. --dc]

From: Frank Randolph CIV Beard

<frank.beard@navy.mil>

Date: Wed, 23 Apr 2003 09:19:15

Subject: Re: the Ada mandate, and why it collapsed and died

Newsgroups: comp.lang.ada

> That's exactly the problem -- because there were so many itsy-bitsy features in Ada, a novice needed to learn all of them, because somebody somewhere finds it useful and it will be found in real-world code.

Nothing says you have to learn every feature of Ada to use it. I think you will inevitably miss out on something if you

don't, but that's a different point. I'm sure you could write many programs with just integers and strings if you wanted to, but odds are you could find a much more elegant and efficient way by knowing more features of the language.

> Having tons of features in auxiliary libraries in the "C" style make mastering the language much simpler by chunking the task of learning without complicating the syntax issues.

I disagree. I see very little difference here. All you've done is defer the problem. If we are talking about a maintenance issue here, at least if it's defined in the language, I can go look it up in the reference manual if I come across something in code I have to maintain. I'd rather be able to look it up in a reference manual than try to figure out how to use something I may not have the documentation to.

You also have the problem of libraries that don't exist on certain platforms.

> But having them directly in the language itself makes just learning the basic language unnecessarily harder. And the trouble is, you don't get anything really worthwhile out of all the time you spend on mastering all that syntax. You would have been better off mastering concepts instead.

You are saying it's better to "master" a core that doesn't do much, and then pick out libraries (if you know about them) that help you do more.

What's the difference in that and picking out a subset of the language to "master", and then picking out more features that help you do more. I'd rather have that in a language that is standardized with features that are guaranteed to be there.

From: P S Norby <psnorby@hotmail.com>

Date: Wed, 23 Apr 2003 17:10:50

Subject: Re: the Ada mandate, and why it collapsed and died

Newsgroups: comp.lang.ada

> Having tons of features in auxiliary libraries in the "C" style make mastering the language much simpler by chunking the task of learning without complicating the syntax issues.

But mastering just the "C" syntax is never enough... you will need to master the libraries. So why is that any easier? And if it's so easy (to learn "C"), why do the ad's all want 5+ years experience?

From: Bill Findlay

<BillFindlay@blueyonder.co.uk>

Date: Wed, 23 Apr 2003 18:57:15

Subject: Re: the Ada mandate, and why it collapsed and died

Newsgroups: comp.lang.ada

> That's exactly the problem -- because there were so many itsy-bitsy features in Ada, a novice needed to learn all of them, because somebody somewhere

finds it useful and it will be found in real-world code.

Novices in Ada, even programming novices learning through Ada, do not need (note, present tense) to deal with any language features apart from those that support the programming concepts they are currently learning.

It is the experience of those who are teaching large numbers of novices (400-500 per year) that they do better in Ada than in any previous language of instruction. That includes Pascal, which was designed for teaching.

> Having tons of features in auxiliary libraries in the "C" style make mastering the language much simpler by chunking the task of learning without complicating the syntax issues.

And don't novices in C have to spend time learning about itsy-bitsy libraries because somebody somewhere depends on them in 'real-world' code?

> But having them directly in the language itself makes just learning the basic language unnecessarily harder.

Syntax is (almost) completely irrelevant to learning programming or a programming language, except insofar as a language actively undermines the student's confidence by virtue of error-prone inconsistencies.

> And the trouble is, you don't get anything really worthwhile out of all the time you spend on mastering all that syntax. You would have been better off mastering concepts instead.

You *do* get something that is very worthwhile for a minor investment in learning a very consistent and robust syntactic framework.

What you get is compiler assistance with the much more difficult and important issues of semantics.

Providing useful semantics via well-defined, built-in language features makes those semantics accessible to the compiler, which can exploit them to benefit the efficiency and the reliability of the object program.

No compiler can do this effectively with external library code, or with programmer-constructed equivalents of Ada's built-in high level features.

I mention, at random: machine-independent integer types, for loops that always terminate, secure dynamic arrays (including strings), secure variant records, exceptions, tasks, automatic mutual exclusion ...

*From: Preben Randhol
<randhol@pvv.org>*

Date: Sun, 27 Apr 2003 20:37:25

Organization: Norwegian university of science and technology

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

[...] At any rate C++ must be one of the worst programming language to be good at as it has so many side effects and peculiarities that you need to read several books to learn how to avoid its pitfalls etc...

*From: Wesley Groleau
<wesgroleau@despammed.com>*

Date: Sun, 27 Apr 2003 17:00:18

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

[...] I read more than half of "C Traps and Pitfalls" NONE of what I read could happen in Ada. [See also "Avoiding Traps and Pitfalls" in AUJ 23.3 (Sep 2002), p.156. -- dc]

*From: John R. Strohm
<strohm@airmail.net>*

Date: Sun, 27 Apr 2003 18:19:54

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

[...] About 90% of programming in Ada is pretty much the same as programming in any other decent language. Most of the lines in Ada AREN'T rep specs or tasking calls or instantiating generics.

*From: Jacob Sparre Andersen
<sparre@crs4.it>*

Date: Tue, 29 Apr 2003 16:02:37

Organization: CRS4, Center for Adv.

Studies, Research and Development in Sardinia

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

It is my experience that there is at least one aspect of Ada, which it is hard (for me) to explain to people in a way that is easy to understand: tagged types

Except for that, I am actually amazed how easy it is for people with a bit of programming experience behind them to get up to speed with programming in Ada.

On Selecting Ada for "Low Level" Programming

From: mcq95@earthlink.net (Marc A. Criley)

Date: 23 Apr 2003 09:31:33

Subject: Re: the Ada mandate, and why it collapsed and died

Newsgroups: comp.lang.java.advocacy,comp.object,comp.lang.ada

[In response to the remark that "Ada was, and is, designed to permit a programmer to get to a very low-level of programming, even to the machine code level, if it seems appropriate." -- dc]

> How many times in your whole life have you done this?

Uh, like, lots.

> Or known of other people who have done this?

You mean the various coworkers of mine over the years?

> And how many instances do you know when this was done by interfacing with C?

Only when having to hook up to code written by others--whatever we did, we did in Ada. The only times we ever had to dip into C was when some supplied function was defined as a macro and we had to put it into a function wrapper so that it could be called.

Marc A. Criley, Quadrus Corporation, www.quadruscorp.com

From: tmoran@acm.org

Date: Wed, 23 Apr 2003 17:18:15

Subject: Re: the Ada mandate, and why it collapsed and died

Newsgroups: comp.lang.java.advocacy,comp.object,comp.lang.ada

My first real Ada program used some machine code inserts to control a video capture card. It also used tasking to interact with the user and to control audio capture. No C drivers were available.

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

Date: 23 Apr 2003 19:31:08

Organization: Pushface

Subject: Re: the Ada mandate, and why it collapsed and died

Newsgroups: comp.lang.java.advocacy,comp.object,comp.lang.ada

The problem was to interface to the PPC timebase register (the mftb instruction) which is not supported in C either. So the choice, learn the asm features of GNAT vs learn the asm features of GCC (which are of course very similar) seemed a no-brainer; we did the GNAT version.

Later I tried the GCC version (this was on VxWorks) and took a couple of hours to realise that in that environment you need to say __asm, because WindRiver use -ansi in C compilations.

Whereas the GNAT version worked first time (it took rather longer to get the 64-bit version working, though).

*From: John R. Strohm
<strohm@airmail.net>*

Date: Thu, 24 Apr 2003 09:09:21

Subject: Re: the Ada mandate, and why it collapsed and died

Newsgroups: comp.lang.java.advocacy,comp.object,comp.lang.ada

> OK, I am sure a few instances happened. But if you had to develop device drivers for a living, and if the language was truly your choice, would you chose Ada as the language? I think the only "yes" answers would be from people who only know Ada and/or have never written a device driver.

Over the course of some 33 years spent in and around computers, I have probably dealt with some twenty different

computer instruction set architectures and assembly languages (PDP-8, PDP-11, Nova, Intel 8080, 8086, 68000, 68020, CDC 6600, DEC-10, to name a few) and probably half a dozen or more high-level languages (FORTRAN IV, FORTRAN 77, PASCAL, COBOL, LISP, Ada, FORTH, C, JOVIAL J73, that I remember offhand), and I have written quite a few device drivers in my day.

As a matter of fact, I *would* usually choose to do the driver in Ada. [...]

From: owski@hotmail.com (Adam Ruth)

Date: 30 Apr 2003 18:32:06

Subject: Re: the Ada mandate, and why it collapsed and died

Newsgroups:

comp.lang.java.advocacy, comp.object, mp.lang.ada

I recently had the (mis)fortune of writing a device driver for NetWare. The only language choices are C and C++. I used C, since C++ support wasn't really up to snuff on NetWare and I'm more comfortable in C. At the time I was writing the driver, I was learning Ada, and by the end of the project I really wish I could have used it. There were entire classes of error I encountered, and had a hard time debugging, that would simply not have existed in Ada. Of course, there were plenty of bugs that would have existed regardless of language, and I'm sure that there are new problems that the use of Ada would introduce. But I'm constantly amazed at the errors that are so easy to avoid in Ada because they simply don't exist.

If I had it to do again (I hope I don't!) and Ada was a choice, I'd give it a shot. I'm convinced that the quality of the product would have been superior.

[See also "Low Level Programming in Ada and C" in AUJ 22.3 (Sep 2001), pp.160-161. -- dc]

On Excuses for Not Selecting Ada

From: john@assen.demon.co.uk (John McCabe)

Date: Fri, 25 Apr 2003 08:20:30

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

> I have not seen any legitimate justification for switching to Ada.

It's hard to prove until you actually do it!

> Which feature of Ada critical to our **** development cannot be achieved with C/C++/Java?

Err - safety probably!

> There are several reasons why Ada is not practical.

No there isn't. There is only one reason why Ada can't be practical and that is that you don't have the time to wait for, or the money to pay for, a compiler to be

produced that supports whatever unusual processor you use. If a compiler exists for your processor there is no reason why Ada isn't practical.

> First, there is no in-house large-scale Ada application development experience among the software developers or civil servants. We can't wait around while 40 developers come up speed. Furthermore, I'll be blunt and say there was "little" in-house C++ experience when the **** was redesigned and we're still paying the price for that inexperience.

Interesting, and not surprising. If they had chosen Ada instead of C++ at the time he would probably be singing its praises! [...]

> The bottomline is that in today's world, if you want to draw from the largest pool of talent, you better be programming in C, C++ or Java. Call it inertia if you want. It doesn't matter really.

Nonsense. A talented software engineer will be able to program in whatever language you ask him to. The only issue here is salary – a young, inexperienced, but talented S/W Eng will choose to program in the language that they perceive as being most financially rewarding.

> Ultimately, any truly critical Ada feature will eventually be added to C++ or Java.

While this is partially nonsense, the fact is that even if C++ and/or Java had all of the features of Ada added, there is no way that the amount of time and effort spent in getting those features *right* in Ada will be done in C++ and especially Java! Java is a flavour-of-the-month language - whatever feature is perceived to be *nice* just gets added irrespective of the overall effect and consistency in the language!

From: Anders Wirzenius

anders.wirzenius@pp.qnet.fi

Date: Fri, 25 Apr 2003 08:33:58

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

> We can't wait around while 40 developers come up speed.

How about eating the elephant piece by piece? Perhaps he can wait around while 2 developers come up speed? Perhaps he can afford some paying for 2 developers inexperience in using Ada? That's roughly 5 per cent of the total of 40 developers. It is not more than what could be a later accepted fall behind of the project schedule.

There might also be a possibility to develop a small part of the system in both C and Ada as two concurrent projects? And then do some safety and efficiency measures on the two versions (and skip the version with lower quality).

From: snarflemike@yahoo.com (Mike Silva)

Date: 25 Apr 2003 10:46:35

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

> I have not seen any legitimate justification for switching to Ada.

So he *has* refuted the various studies and white papers you've pointed him to?

> Which feature of Ada critical to our **** development cannot be achieved with C/C++/Java?

The old Turing argument. We can do anything in any language!

> [...] The bottomline is that in today's world, if you want to draw from the largest pool of talent, you better be programming in C, C++ or Java.

[...] BTW, I imagine that the *percentage* of C/C++/Java developers who could easily step into an ATC project is a lot lower than the percentage of Ada developers who could easily do so. Lots less reading of resumes and interviewing only to end up at don't-call-us-we'll-call-you.

> Call it inertia if you want. It doesn't matter really. Ultimately, any truly critical Ada feature will eventually be added to C++ or Java.

This is the best yet! They don't have time to train their developers, but they have time to wait for truly critical features to be added to C++ or Java!

Oh, and there's a really big talent pool of VB programmers out there. Any truly critical Ada feature will eventually get added to VB, right?

I think the best you could hope for is to do a part of the project in Ada and have great success while the rest of the project founders. :-/

From: John R. Strohm

strohm@airmail.net

Date: Fri, 25 Apr 2003 14:52:11

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

Mike has the beginnings of a really good idea here, if you can find a sympathetic, smart senior manager.

Your firm probably has some internal R&D budget laying around. Take a good look at the upcoming air traffic management project, find a chunk that can be relatively easily carved out, that IS fairly central, not peripherally trivial. Put together a proposal package, containing the various success studies from Boeing, Pratt & Whitney, and elsewhere, along with the down-the-hall guy's diatribe, and some cost estimates, and go pitch a quiet, back hallway IR&D, to FIND OUT whether Ada in fact is as good as Boeing and Pratt say it is. Take the same metrics as the mainstream guys are going to take, track the same deliverables, and then, when the dust settles, see who got thar fustest with the mostest, to paraphrase

someone or other from the American Revolution.

Work from the same requirements specifications and interface control documents as the C/C++/Java folks. You want to demonstrate something that will plug-and-play with their system.

Admit up front that there will be some training costs, but these other studies (have them in your hands) show that it won't be that bad.

Play up the quality/reliability/maintainability stuff. Note in passing that the Ada programmers on the market tend to be MUCH more accustomed to working in high-reliability applications than the typical C++ programmers.

*From: Hyman Rosen <hyrosen@mail.com>
Date: Fri, 25 Apr 2003 11:01:00
Organization: KBC Financial Products
Subject: Re: Ada in Iraq
Newsgroups: comp.lang.ada*

> We can't wait around while 40 developers come up speed. There was "little" in-house C++ experience and we're still paying the price for that inexperience.

He has a point there. If you've got the same programmers who screwed up because they tried to develop a system in C++ without knowing C++, they'll screw up a system in Ada because they don't know Ada. Obviously, your management doesn't feel that it's necessary for programmers to be expert in the language they use. Too bad for them, and for you.

> if you want to draw from the largest pool of talent, you better be programming in C, C++ or Java.

The pool is large, but it's shallow, too. You have no idea (or maybe you do) how many people claim to be expert in those languages who barely know the first thing about them.

> Ultimately, any truly critical Ada feature will eventually be added to C++ or Java.

Certainly not within the timeframe of the project, if ever. And then you would be working with cutting-edge language changes, which is a recipe for disaster for production use. [...]

*From: Samuel Tardieu <sam@rfc1149.net>
Date: Fri, 25 Apr 2003 17:13:14
Organization: TELECOM Paris
Subject: Re: Ada in Iraq
Newsgroups: comp.lang.ada*

> He has a point there.

I don't think so. The "point" assumes that what was to blame was the insufficient knowledge of C++, and that this knowledge has not been acquired. What if the problem was in fact C++ itself?

From: john@assen.demon.co.uk (John McCabe)

Date: Mon, 28 Apr 2003 08:17:41

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

> Well, that's not what the e-mail said. It specifically blamed lack of experience with the language, not the language itself.

It is quite reasonable to hypothesize (as Sam has) that 'lack of experience' is an subjective view. That may not be the real problem, just the problem as seen by the person who wrote the original message.

The language itself, and more specifically the fact that it doesn't explicitly provide mechanisms for programming concurrent, safe and complicated systems (much of this needs to be provided by the operating system or by non-standard compiler extensions in C++) could be at least one of the major problems with this development.

Ada on the other hand has all this functionality built in to the language in a standard manor (sic:-). What is also not addressed in the original message is what experience the team did have prior to its use of C++. If it had been something like Pascal, Modula-2 etc (i.e. those languages that were similar to Ada) then moving to Ada could have been a relatively straightforward transition however moving to C++ could have been far more difficult.

On "Long Lasting" Languages

*From: tmoran@acm.org
Date: Sun, 04 May 2003 16:45:08
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

> [...] something like C. They don't want to be stuck out there in 10 years looking for support for a language and not finding it.

I don't understand this argument. "C" is not a long lasting language. There was K&R C, then ANSI C, then C++, then Java, then C#.

How many colleges are today teaching K&R C? How many K&R C compilers, or their vendors, are still around for 10 or 20 year old "C" programs? Where the project's "C compiler" has been carefully preserved (probably because the code depends on its known unique idiosyncrasies) how many of today's new hires understand it? How many want to add "experience in Defunct Corp's K&R C compiler" to their resumes?

On Language Experts, Training and Good Design

*From: Wesley Groleau
<wesgroleau@despammed.com>
Date: Fri, 25 Apr 2003 22:29:09
Subject: Re: Ada in Iraq
Newsgroups: comp.lang.ada*

> If you've got the same programmers who screwed up because they tried to develop a system in C++ without knowing C++, they'll screw up a system in Ada because they don't know Ada.

I'd put it slightly differently: the same programmers who screwed up because they tried to develop a system in a language they weren't capable of learning, will screw up a system in any language for the same reason.

*From: Hyman Rosen <hyrosen@mail.com>
Date: Sun, 27 Apr 2003 20:07:04
Subject: Re: Ada in Iraq
Newsgroups: comp.lang.ada*

It's not really incapability of learning. It's a matter of trying to use a language with only a cursory understanding of it, instead of a deep one. When you do that, you're going to fail, or at least "succeed badly", because you will fail to use the features of the language to their best advantage. It's the old concept of being able to code Fortran in any language. Think of how many posts in this newsgroup criticize Ada newbies for trying to transliterate C into Ada instead of learning how things should be done properly in Ada. Now imagine them actually going off to build a system in that style.

Without an expert in Ada around to supervise the training, the people learning Ada aren't going to know when they've learned enough about the language to use it properly.

*From: Wesley Groleau
<wesgroleau@despammed.com>
Date: Sun, 27 Apr 2003 17:00:18
Subject: Re: Ada in Iraq
Newsgroups: comp.lang.ada*

There were few Ada experts available when some of the successful projects started. [See also "Successful Projects Without Prior Ada Experience" earlier in this AUJ issue. -- dc]

*From: Preben Randhol
<randhol@pvv.org>
Date: Mon, 28 Apr 2003 17:33:37
Organization: Norwegian university of science and technology
Subject: Re: Ada in Iraq
Newsgroups: comp.lang.ada*

> When people set out to design a software system in Ada, and they don't know Ada thoroughly and deeply, how likely is it that the design will be "good Ada", as opposed to a bastard version of the language that they do know?

When people set out to design a software system in C++, and they don't know C++ thoroughly and deeply, how likely is it that the design will be "good C++" [*], as opposed to a bastard version of the language that they do know?

My answer is that I believe they will find it easier to make a good design in Ada as opposed to C++.

[*] A mindboggling concept, but for the sake of this exercise.

From: Preben Randhol
<randhol@pvv.org>

Date: Tue, 29 Apr 2003 06:21:49

Organization: Norwegian university of
science and technology

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

> I'm in favor of having a deep understanding of a language before I go off and build a large system with it.

IMHO this is truer for C++ as it has so many quirks and bogs to stay clear of. C and C++ expect the programmer not to do any mistakes, which is not exactly what you may expect no matter how well the person is or how extensive his knowledge of the language is.

> Probably so, but will it look anything like what an experienced Ada practitioner would have done?

Your point here being? My question is how many of the so called C++ programmers that are so readily available are able to program a big system in C++?

From: Ole-Hjalmar Kristensen
<oleh@vlinux.voxelvision.no>

Date: Tue, 29 Apr 2003 14:12:30

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

> I'm not talking about C++, I'm talking about whether you can take programmers who don't know Ada and have them produce a well-designed system in Ada before they acquire a deep and thorough understanding of the language.

In my experience, yes. But that's more a function of sound design than choice of language. The project I'm thinking of was originally going to be implemented in C, but the customer wanted us to switch to Ada after the design was finished. Design was done mostly in SDL, btw. The programmers got a crash course in Ada, and went at it, using a subset which was roughly equivalent to Pascal plus generic packages. Tasks were not used because of limitations in the DBMS we were interfacing to. We could certainly have used more of Ada, but given the time constraints and the existing design it seemed sensible to stick to a limited subset.

The system was the first ATCCIS demonstration facility, btw.

From: Jerry Petrey
<jdpetrey@raytheon.com>

Date: Tue, 29 Apr 2003 09:15:38

Organization: Raytheon Company

Subject: Re: Ada in Iraq

Newsgroups: comp.lang.ada

> [...] whether you can take programmers who don't know Ada and have them produce a well-designed system in Ada before they acquire a deep and

thorough understanding of the language.

I understand your point. I think some of the worst code I have ever seen has been Ada written by people who still think in their old language (Fortran, C, etc.) and who have not taken (or been given) the time to learn Ada properly. Companies rarely devote much to training their people in a new language and unfortunately, many software engineers also don't devote much to this effort. The attitude is that if you know how to program in one language, you can learn how to use another with little effort. While this may be true, I contend that you won't be writing the best (and most maintainable) code unless to devote some serious effort in learning to 'think' in the new language. This is true in any language, I believe, which is probably what Preben was trying to say when he challenged you about your question.

That said, I still feel that Ada offers a great advantage over most other languages in that even code written by under trained newcomers is still more maintainable and the strong typing and compiler checking helps these programmers to become more proficient.

Some of the things that would make a huge improvement in software quality, safety and maintainability would be:

1. Companies would allow developers to expend the extensive effort required to become true craftsmen in the language chosen.
2. Managers would manage. Experienced developers would have more control over the project.
3. A good process would be followed without exceptions.
4. Thorough testing would be done.
5. The first attempt on a new application (especially by a team inexperienced with the language) would be treated as a learning experience, discarded and rewritten.
6. The proper tools would be acquired and kept up to date.

Since these things are rarely done, I think the advantages of Ada gives it an edge in the software development process and we need every bit of help possible because most projects are so poorly managed and have time and budget constraints.

Jerry Petrey, Senior Principal Systems Engineer, Raytheon Missile Systems

On Ada Success Stories

From: Richard Riehle
<richard@adaworks.com>

Date: Fri, 25 Apr 2003 18:32:04

Organization: AdaWorks Software Engineering

Subject: Re: the Ada mandate, and why it collapsed and died

Newsgroups:

comp.lang.java.advocacy,comp.object,comp.lang.ada,misc.misc

> So you are saying the Ada projects that succeeded were secret, and the ones that were not secret did not succeed?

Go to www.adaic.org. There are plenty of case histories of Ada projects that succeeded. This includes a large number of Air Traffic Control systems, commercial aircraft avionics, power industry systems, manufacturing process control systems, transportation systems, the global positioning satellites, many communication satellites (Intelsat VII, for one), and more. They are not secret. They are simply not widely publicized.

Notice that these systems tend to be large-scale, safety-related, and have long lifetimes. The fact that you are personally ignorant of Ada, its successes, and its benefits does not detract from its history in the building of successful systems. The fact that some software developers have used Ada badly does not imply there is a problem with the language. Rather, it implies that those developers are less competent than those who have used it successfully.

I have seen people take on Ada projects and fail. I have seen others take on Ada projects and succeed beyond what they would have believed possible when they first started. Those who failed would probably have failed using any other toolset or language. More often than not, the failure is due to stupid project management. At least one huge project in Ada failed because the people managing it were incompetent, and Ada was their scapegoat. All they could do, for years afterward, was denigrate the language. The programmers, many of whom I know quite well, knew the real cause was not Ada, but the people in charge of making it work.

One well-known company, a manufacturer of office copying machines, allowed one of its software development groups to build the embedded software in Ada. The group, and the manager, was amazed at their success and decided they wanted to use Ada for future projects. The corporate level management of this company [...] dictated that all software would henceforth be written in C, this in spite of the phenomenal success of Ada.

There are plenty of success stories outside the Department of Defense. In fact, Ada in the hands of competent commercial developers may have been more successful, if not more widespread, than in the weapon systems area.

You may want to grouse about the language you seem not to understand, and that is your right. [...] When you look at the successes of those who do know how

to build software, you will find that Ada measures up quite well against any competing technology.

An Opinion on Market Niches

*From: Marin David Condic
<mcondic@acm.org>
Date: Tue, 6 May 2003 08:07:39
Subject: Re: employment with ada
Newsgroups: comp.lang.ada*

[In reponse to remarks that vendors would better focus on Ada's "core business" of safety-critical and embedded applications, instead of what is needed for PC applications: -- dc]

> Does realtime, embedded work involve GUIs? databases?

Not often, but if there were a few million Ada developers out there making PC apps with Ada, nobody would object that they can't find Ada developers. If there were a few million Ada/PC developers with good quality compilers and support tools on the PC backing them up, people might be tempted to start doing realtime/embedded work in Ada where they had a PC SBC as their target. If Ada compiler vendors were making money hand over fist selling PC-based compilers, they might be tempted to retarget them to other boards for embedded use.

Success breeds success. At this point, I don't think it would be easy or likely to get Ada adopted in most embedded efforts because of market resistance and lack of adequate environments to do the job. Its a lot easier to get it adopted in areas with less resistance and less demand for specific tools. You don't need nearly as much to get a PC based app up and running so Ada doesn't have to provide as much to be competitive in that market. If you got acceptance in one market, its easier to then bleed over into other markets. Start with what is achievable and work from there.

> How much safety-critical embedded avionics software gets written in C?

Lots.

We don't help ourselves if we delude ourselves into thinking that Ada is some giant success out there in some invisible market that if only we could lift the veil we'd see a multi-billion dollar business. Anything can be a "success" if we define "success" narrowly enough. (Q: How many engine controls for STOVLA aircraft, designed in Palm Beach Gardens, Florida are programmed in Ada? A: All of them. Does this make a difference to most Ada developers looking for work?)

Even if Ada had *all* the safety-critical embedded avionics software development in the US, this wouldn't be a really big market compared to the overall software market. Since that market is small, the people making decisions about where to go with tools, etc., get driven by costs and other factors to go use what the rest of the world is using. Arguing that Ada is "better" for this kind of work is interesting, but not compelling. Quality of the language is only one factor that has to be considered. Making Ada more of a market force that didn't leave us poor slobs building engine controls in P.B.G., FL feeling like we were off in a desert somewhere might make the decision to use Ada an easier one.

I'm a big fan of Ada, but I think it needs to get off bottom-dead-center in *some* market and start dominating it if it is going to have good long-term prospects. "Realtime/Embedded" as a market looks like a tough nut to crack and the Ada vendors don't have the budget to attempt to address it on speculation. Some subset of the PC app market would be an easier target and I think that developing libraries, GUIs, databases, etc., to suit some niche in there would be a lot more doable.

Marin David Condic. I work for:
<http://www.belcan.com/>.
My project is: <http://www.jsf.mil/>

On Experience Requirements in Job Ads

*From: Robert I. Eachus
<rieachus@attbi.com>
Date: Wed, 11 Jun 2003 18:05:05
Subject: Re: C bug of the day
Newsgroups: comp.lang.ada*

> I saw one ad that required "two years experience in C# or one year each at at least two jobs." This was when C# was less than three years old!

My favorite was an ad on the back cover of Communications of the ACM requiring 10 years of Ada experience. I'd have to check old ARG minutes to set an exact date, I think 1988. (There was an ARG meeting in Munich along with the Ada Europe meeting.) At dinner during the ARG meeting we tried to figure if there was anyone sitting at the table who qualified. (Jean Ichbiah was there along with John Barnes, Robert Dewar, Mike Woodger and a few others active early in the Ada design effort.)

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

- November 03-06 3rd **International Conference on Quality Software** (QSIC'2003) Dallas, USA, QSIC'2003 as rescheduled from 2003/09/25-26 in 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.
- ☉ 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 04-07 2nd **International Symposium on Formal Methods for Components and Objects** (FMCO'2003) Leiden, The Netherlands.
- ☉ November 09-11 5th **Real-Time Linux Workshop**. Valencia, Spain. Topics include: modifications and variants of the GNU/Linux operating system, contributions and tools around and to real-time Linux variants, real-time Linux applications, in academia, research and industry, tools for embedding Linux or real-time Linux, embedded real-time Linux applications, work in progress reports, educational material on real-time Linux.
- ☉ November 11 Baltimore **ACM SIGAda Meeting** - Writing and Filing US Patents - From a Software Perspective. Baltimore, USA.
- November 13-16 10th **Working Conference on Reverse Engineering** (WCRE'2003) Victoria, British Columbia, Canada. Topics include: Software maintenance and evolution, Program comprehension, Software architecture extraction, Software migrations, Experience reports, Reengineering patterns, Preprocessing and parsing, Software components, Refactoring, Reverse engineering tool support, Programming knowledge representations, UML and round trip engineering, Software metrics, etc.
- November 15-21 **International Conference for High Performance Computing and Communications** (SC'2003) Phoenix, AZ. Topics include: Embedded- and/or reconfigurable systems; Clusters and distributed systems; Fault-tolerance, reliability, maintainability, and adaptability; Parallel programming methods and languages; Software Tools; etc.
- November 17-20 14th **IEEE International Symposium on Software Reliability Engineering** (ISSRE'2003) Denver, CO, USA.
- November 17-21 John Robinson & Associates - **Public Ada Training Course**. Cheltenham, UK. A five-day practical course with two streams covering Ada 83 and Ada 95.
- November 18-21 4th **International IFIP Working Conference on Distributed Applications and Interoperable Systems** (DAIS'2003). Paris, France. In conjunction with FMOODS'2003. Topics include: models, technologies and platforms for reconfigurable, scalable and adaptable distributed applications. Includes a.o the following event:

- © November 17 **Workshop on Communication Abstractions for Distributed Systems (CADS'2003)**
- November 18-21 **6th IFIP International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS'2003)**. Paris, France. In conjunction with DAIS'2003
- November 21 **XP Day Benelux 2003**. Breda, The Netherlands
- 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 02-04 **16th International Conference on Software & Systems Engineering and their Applications (ICSSEA'2003)**. Paris, France
- 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)**. Graz, Austria
- 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.
- December 17-20 **10th International Conference on High Performance Computing (HiPC'2003)**. Hyderabad, India. Topics include: Scientific/Engineering Applications, Commercial Applications and Workloads, Parallel and Distributed Computing, Heterogeneous Computing, Embedded Applications and Systems, Parallel Languages and Programming Environments, etc. Deadline for early registration: November 15, 2003
- December 27-30 **5th International Workshop on Distributed Computing (IWDC'2003)**. Calcutta, India. Topics include: Fault Tolerance and Reliability; Distributed Objects, Data Management and Algorithms; Real-Time and Embedded Systems; Middleware; etc.

2004

- January 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, ...); Testing and Certification of Trustworthy Systems (New techniques for testing and certification of software systems, Object-oriented testing methods and tools, Integrating quality attributes into testing and certification, Automated tools for testing and certification support, Roles and techniques for correctness verification in system certification, Industrial case studies in testing and certification, ...), etc.

- © January 05-08 **Workshop on Distributed Objects Research, Experiences & Applications** (DOREA'2004). Cancun, Mexico. Topics include: Software architectures for distributed objects; Use of and extensions to technologies such as CORBA, J2EE, and .NET; Experience reports; Applications of distributed objects; etc.
- January 14-16 **31st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages** (POPL'2004). Venice, Italy. Includes a.o the following event:
- © January 13 **Workshop on Programming Language Technologies for XML** (PLAN-X'2004). Topics include: XML parsing, integration of XML in general-purpose programming languages, etc.
- January 21-23 **2nd European Congress on Embedded Real Time Software** (ERTS'2004). Toulouse, France. Topics include: Languages, Process, Methods and Tools; Dependability; etc.
- February 02-04 **3rd International Conference on COTS-Based Software Systems** (ICCBSS'2004). Redondo Beach, California, USA. Deadline for early registration: December 15, 2003
- February 11-13 **12th Euromicro Conference on Parallel, Distributed and Network-based Processing** (PDP'2004). A Coruña, Spain. Topics include: Advanced Applications (numerical applications with multi-level parallelism, real time distributed applications, distributed business applications, ...); Distributed Systems (distributed OO systems, heterogeneous distributed systems, distributed shared memory systems, distributed agent systems and applications, ...); Languages, Compilers and Runtime Support Systems (task and data parallel languages, object-oriented languages, dependability issues, ...); Parallel Computer Systems (embedded parallel and distributed systems, fault-tolerance, ...); etc.
- March 01-03 **16th Conference on Software Engineering Education and Training** (CSEET'2004). Norfolk, Virginia, USA. Theme: "Educating Software Engineers to Face the Future's Challenges" Topics include: Teaching principles and skills needed to produce secure software systems. Developing students' understanding of and commitment to software reliability. Current (or contemporary) software development processes and how best to teach them. Best practices for software engineering education both in computer science programs and in dedicated software engineering degree programs. Etc.
- March 08-11 **16th Software Engineering Process Group Conference** (SEPG'2004) Orlando, Florida
- March 14-17 **2004 ACM Symposium on Applied Computing** (SAC'2004). Nicosia, Cyprus. Includes tracks on Embedded Systems: Applications, Solutions, and Techniques; etc. Includes a.o the following events:
- © March 14-17 *Track on Object-Oriented Programming Languages and Systems* (OOPS'2004). Topics include: Programming abstractions; Advanced type mechanisms and type safety; Multi-paradigm features; Language features in support of open systems; Aspect-oriented and Component-based programming; Reflection, meta-programming; Program structuring, modularity, generative programming; Distributed Objects and Concurrency; Middleware; Heterogeneity and Interoperability; Applications of Distributed Object Computing; etc.
- © March 14-17 *Track on Programming Languages* (PL'2004). Topics include: Compiling Techniques; Garbage Collection; Language Design and Implementation; Practical Experiences with Programming Languages; Program Analysis and Verification; Programming Languages from All Paradigms (Agent-Oriented, Aspect-Oriented, Functional, Logic, Object-Oriented, etc.); etc.
- March 22-24 **3rd International Conference on Aspect-Oriented Software Development** (AOSD'2004). Lancaster, UK. Deadline for submissions: November 7, 2003 (demonstrations)
- March 22-26 **International Conference on Practical Software Quality Techniques & Testing Techniques** (PSQT/PSTT'2004 East). Washington DC Area, USA. Deadline for submissions: November 3, 2003 (tutorials), November 17, 2003 (papers)
- March 24-26 **8th European Conference on Software Maintenance and Reengineering** (CSMR'2004). Tampere, Finland. Topics include: Experience reports (successes and failures); Tools and enabling

technologies for evolution, maintenance and reengineering tools; Migration, wrapping and interfacing legacy systems; Dealing with legacy systems towards new technologies; etc.

- March 25-26 **8th IEEE International Symposium on High Assurance Systems Engineering (HASE'2004)**. Tampa, Florida, USA. Topics include: Formal Methods; Safety analysis, reliability evaluation and enhancement techniques; Fault-tolerant software design; Evolutionary design of complex systems; Software engineering for embedded systems; etc. Deadline for submissions: November 10, 2003 (fast abstracts, industrial experience papers)
- March 27-04 **European Joint Conferences on Theory and Practice of Software (ETAPS'2004)**. Barcelona, Spain. Event includes: conferences from 29 March to 2 April, 2004, affiliated workshops on 27-28 March and 3-4 April, 2004
- April 05-07 **4th International Conference on Integrated Formal Methods 2004 (IFM'2004)**. Canterbury, Kent, England
- April 14-16 **9th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'2003)**. Florence, Italy. Topics include: Tools, environments, and languages for complex systems; Formal methods and approaches to manage and control complex systems; Integration of heterogeneous technologies; Human factors and collaborative aspects; Interoperability and standardization; Systems and software safety and security; Industrial automation, embedded and/or real time systems; etc.
- April 19-22 **Annual Software Technology Conference (STC'2004)**. Salt Lake City, Utah, USA
- April 21-23 **5th International Conference on Software Testing (ICSTEST'2004)**. Duesseldorf, Germany
- © April 26-30 **International Parallel and Distributed Processing Symposium (IPDPS'2004)**. Santa Fe, New Mexico, USA. Topics include: Applications of parallel and distributed computing; Parallel and distributed software, including parallel programming languages and compilers, operating systems, runtime, middleware, libraries, programming environments and tools for parallel and distributed computing; etc. Includes a.o the following events:
- © April 26 **9th International Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS'2004)**. Topics include: Concepts and languages for parallel and Grid computing (Language interoperability, Concurrent object-oriented programming, ...), Supportive techniques for component environments and testbeds (Runtime systems, Compiler techniques, ...), etc. Deadline for paper submissions: November 17, 2003
- April 26 **2nd International Workshop on Parallel and Distributed Systems: Testing and Debugging (PADTAD'2004)**. Deadline for paper submissions: December 15, 2003
- April 28-29 **4rd International SPICE Conference on Software Process Improvement and Capability dEtermination (SPICE'2004)**. Lisbon, Portugal. Deadline for submissions: November 14, 2003
- May 02-06 **Conference on Design, Analysis, and Simulation of Distributed Systems (DASD'2004)**. Washington DC, USA. Topics include: Application oriented methods and tools; Aspects of real-time systems; Case studies, best practices and lessons learned; Fault tolerance / reliability; Modeling of distributed systems including analysis and simulation; New formal concepts and methods for validation and testing; Security and safety; Support for HW/SW codesign; etc.
- © May 12-14 **7th IEEE International Symposium on Object-oriented Real-time distributed Computing (ISORC'2004)**. Vienna, Austria
- May 14-17 **International Conference on Computational Science and its Applications (ICCSA'2004)**. Assisi, Italy. Topics include: Parallel and Distributed Computing; Reliability Engineering; Software Engineering; etc. Deadline for submissions: November 15, 2003 (workshops, sessions), December 15, 2003 (abstracts), December 31, 2003 (full papers), January 15, 2004 (exhibition proposals)

- ☉ May 23-28 26th **International Conference on Software Engineering** (ICSE'2004). Edinburgh, Scotland, UK. Deadline for submissions: March 8, 2004 (workshops, informal research demos, posters). Includes a.o the following event:
- May 24-25 7th **Workshop on Component-Based Software Engineering** (CBSE7)
Deadline for submissions: December 20, 2003
- ☉ May 24-27 **DAta Systems In Aerospace** (DASIA'2004). Istanbul, Turkey
- May 24-28 11th **IEEE Symposium and Workshops on the Engineering of Computer Based Systems** (ECBS'2004). Brno, Czech Republic. Deadline for submissions: November 15, 2003 (workshop papers)
- ☉ May 25-28 10th **IEEE Real-Time and Embedded Technology and Applications Symposium** (RTAS'2004). Toronto, Canada. Topics include: QoS issues in systems integration, software engineering, programming languages, system development tools, etc. Special focus is on embedded and real-time applications. Deadline for submissions: January 12, 2004
- May 26-28 10th **International Workshop on Future Trends of Distributed Computing Systems** (FTDCS'2004). Suzhou, China. Topics include: Real-time, Pervasive, and Embedded Systems; Security and Trust in Distributed Systems; Highly Available Distributed Systems; Novel Distributed Applications; etc. Deadline for submissions: December 1, 2003
- June 06-10 5th **International Conference on eXtreme Programming and Agile Processes in Software Engineering** (XP'2004). Garmisch-Partenkirchen, Germany. Deadline for submissions: January 11, 2004 (papers, tutorials, workshops, panels, March 1, 2004 (PhD symposium))
- June 07-09 4rd **International Conference on Computational Science** (ICCS'2004). Krakow, Poland. Deadline for submissions: December 1, 2003 (papers), January 15, 2004 (exhibitors)
- June 11-13 ACM SIGPLAN 2004 **Conference on Programming Language Design and Implementation** (PLDI'2004). Washington, DC, USA. Includes a.o the following event:
- ☉ June 11-13 **ACM SIGPLAN Conference on Languages, Compilers, and Tools for Embedded Systems** (LCTES'2004). Topics include: Programming languages for embedded applications; Real-time UML; Object-oriented modeling and design; Validation and verification techniques for embedded software; Real-time scheduling analysis; Exception and interrupt handling for real-time; Software design for multiprocessor embedded systems; Concurrent+distributed embedded environments/runtime systems; Support for partitioning; etc. Deadline for submissions: February 6, 2004
- ◆ June 14-18 9th **International Conference on Reliable Software Technologies - Ada-Europe'2004**. Palma de Mallorca, Spain. Sponsored by Ada-Europe, in cooperation with ACM SIGAda (approval pending) and Ada-Spain.
- ☉ June 14-18 18th **European Conference on Object-Oriented Programming** (ECOOP'2004). Oslo, Norway. Topics include: Analysis and design methods; Concurrent, real-time, and parallel systems; Design patterns; Distributed and mobile systems; Language design and implementation; Programming environments; Versioning, compatibility, software evolution; etc. Deadline for submissions: December 1, 2003 (papers, tutorials, workshops), April 1, 2004 (demos, posters, exhibits, practitioners reports)
- June 27-July 02 29th **Annual USENIX Technical Conference** (USENIX'2004). Boston, Massachusetts. Topics include: Distributed and parallel systems, Embedded systems, Reliability and availability, etc., plus FreeNix Track on free and open source software. Deadline for submissions: December 16, 2003
- June 28-July 01 **International Conference on Dependable Systems and Networks** (DSN'2004). Florence, Italy. Deadline for submissions: November 14, 2003 (papers, panels), January 12, 2004 (workshop submissions, tutorials), April 26, 2004 (student forum, fast abstracts)
- ☉ June 30-July 02 16th **Euromicro Conference on Real-Time Systems** (ECRTS'04). Catania, Italy. Topics include: embedded real-time systems; real-time control applications; frameworks and tools for

development and analysis; software architectures and languages; design, scheduling, timing and execution-time analysis; validation; etc. Deadline for full paper submissions: December 7, 2003

- July 12-16 10th **International Conference on Algebraic Methodology And Software Technology** (AMAST'2004). Stirling, Scotland, UK. Deadline for paper submissions: end December 2003 / early January 2004
- ☉ September 22-24 **GI-Jahrestagung Informatik 2003** - Teiltagung "Sicherheit - Schutz und Zuverlässigkeit". Ulm, Germany. Deadline for submissions: January 31, 2004 (workshop proposals), April 31, 2004 (workshop submission)
- December 06-08 6th **Symposium on Operating Systems Design and Implementation** (OSDI'2004). San Francisco, California. Topics include: distributed systems, embedded systems, etc. Deadline for paper submissions: May 14, 2004
- December 10 Birthday of Lady Ada Lovelace, born in 1815. Happy Programmers' Day!

Additional Numeric Packages for Ada 2005

John Barnes

11 Albert Road, Caversham, Reading RG4 7AN, England.; Tel:+44 118 947 4125;
email: jgpb@jbinform.co.uk

Abstract

This paper discusses proposed additional packages in the Numerics Annex for the forthcoming Amendment to Ada aimed for 2005. These are based in part on the facilities currently in the ISO secondary standard 13813 entitled "Generic packages of real and complex type declarations and basic operations for Ada (including vector and matrix types)".

Keywords: numerics, Ada.

Introduction

The background to this topic is partly described in a short paper published in this Journal in June 2001 entitled "Do you use Generic Arrays and Vectors?" [1]. To recap briefly, there were four secondary ISO numeric standards developed for Ada and all the functionality of three of these was incorporated into Ada 95. In the case of the fourth, 13813, some of the functionality was incorporated into Ada 95 but two packages were not. These were a (generic) package defining types for real vectors and matrices and operations upon them and a similar package for complex vectors and matrices.

The development of the Amendment for Ada 2005 provides an opportunity to incorporate these packages into the language standard rather than leaving them in a secondary standard which is almost invisible to the user community. Naturally enough they would be included in the Numerics Annex.

However, there is the question of whether the packages as in 13813 are satisfactory as they are or whether they should be modified in some way. The main purpose of this paper is to describe the possible extensions that are being considered by the Ada Rapporteur Group (ARG) and to invite comments from the user community.

The packages as in 13813

The starting point is the packages as in 13813. Here is the specification of the package for real types.

```
generic
  type Real is digits <>;
package Ada.Numerics.Generic_Real_Arrays is
  pragma Pure(Generic_Real_Arrays);

  type Real_Vector is array (Integer range <>) of
    Real'Base;
  type Real_Matrix is array (Integer range <>,
    Integer range <>) of Real'Base;
```

```
-- Real_Vector arithmetic operations
function "+" (Right: Real_Vector) return Real_Vector;
function "-" (Right: Real_Vector) return Real_Vector;
function "abs" (Right: Real_Vector) return Real_Vector;
function "+" (Left, Right: Real_Vector)
  return Real_Vector;
function "-" (Left, Right: Real_Vector)
  return Real_Vector;

function "*" (Left, Right: Real_Vector)
  return Real'Base;

-- Real_Vector scaling operations
function "*" (Left: Real'Base; Right: Real_Vector)
  return Real_Vector;
function "*" (Left: Real_Vector; Right: Real'Base)
  return Real_Vector;
function "/" (Left: Real_Vector; Right: Real'Base)
  return Real_Vector;

-- Other Real_Vector operations
function Unit_Vector (Index: Integer; Order: Positive;
  First: Integer := 1) return Real_Vector;

-- Real_Matrix arithmetic operations
function "+" (Right: Real_Matrix) return Real_Matrix;
function "-" (Right: Real_Matrix) return Real_Matrix;
function "abs" (Right: Real_Matrix) return Real_Matrix;
function Transpose (X: Real_Matrix)
  return Real_Matrix;

function "+" (Left, Right: Real_Matrix)
  return Real_Matrix;
function "-" (Left, Right: Real_Matrix)
  return Real_Matrix;
function "*" (Left, Right: Real_Matrix)
  return Real_Matrix;

function "*" (Left, Right: Real_Vector)
  return Real_Matrix;

function "*" (Left: Real_Vector; Right: Real_Matrix)
  return Real_Vector;
function "*" (Left: Real_Matrix; Right: Real_Vector)
  return Real_Vector;

-- Real_Matrix scaling operations
function "*" (Left: Real'Base; Right: Real_Matrix)
  return Real_Matrix;
function "*" (Left: Real_Matrix; Right: Real'Base)
  return Real_Matrix;
function "/" (Left: Real_Matrix; Right: Real'Base)
  return Real_Matrix;

-- Other Real_Matrix operations
function Unit_Matrix (Order: Positive;
  First_1, First_2: Integer := 1) return Real_Matrix;
end Ada.Numerics.Generic_Real_Arrays;
```

This is all straightforward and provides operations for adding subtracting and multiplying matrices and vectors by matrices and vectors and also by scalars. These enable all

the usual expressions to be written in a natural form. Note that there are two operations which multiply two vectors. One (the inner product) produces a scalar and the other (the outer product) produces a matrix. Ada conveniently distinguishes them through the overload resolution rules. In tensor notation we might write these operations as

```
x = vi.vi  inner product
aij = vi.vj  outer product
```

We have actually omitted three operations that were in 13813 that did not seem appropriate; they did component by component multiplication, division and exponentiation.

Of course, none of the above are particularly exciting. Any user can code them easily since they simply require a few loops. The main value of 13813 is that it provides a uniform baseline from which other packages can be defined. However, as it stands, it is dull. Moreover, the Rationale for Ada 95 (section G.1.1) says:

"A decision was made to abbreviate the Ada 95 packages by omitting the vector and matrix types and operations. One reason was that such types and operations were largely self-evident, so that little real help would be provided by defining them in the language. Another reason was that a future version of Ada might add enhancements for array manipulation and so it would be inappropriate to lock in such operations prematurely."

Well, there are not going to be any enhancements for array manipulation (slicing and trimming as in Algol 68 was perhaps being contemplated by the writer of that part of the Rationale) and so that excuse for omitting them vanishes.

But they are still boring. About the only possible excitement is that the inner product could be computed in a more accurate way than just as a sum of products. It could for example use extended arithmetic or even built-in hardware. The elderly British reader might remember the special instruction in the stack-based English Electric KDF9 (circa 1960) for just this. Thus the short loop

```
*7; M0M1Q; M0M2Q; ×+F; *J7C1NZS;
```

computes the inner product of the vectors M1 and M2 using the special ×+F double length multiply and add instruction. Gosh it was fast for those days.

The ARG thus concluded that the package as in 13813 is not really worthwhile and so it was decided that some added value should be considered.

Inversion

The obvious omission is the ability to invert a matrix. Anyone can program addition, subtraction and multiplication of matrices but division is not so easy. Of course, thought of as group operations, the computation of the inverse is the required operation. However, in order to solve a single set of linear equations

$$\mathbf{Ax} = \mathbf{y}$$

it is inefficient to compute the inverse \mathbf{A}^{-1} and then to multiply by \mathbf{y} in the obvious way

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$

Accordingly we need separate subprograms for solving sets of linear equations. It is also handy to be able to compute the determinant since it is the measure of whether the matrix is ill-conditioned and thus likely to cause trouble in solving equations.

This leads to the following four additional functions in the package.

```
-- Real_Matrix inversion and related operations
function Solve (A: Real_Matrix; X: Real_Vector)
  return Real_Vector;
function Solve (A, X: Real_Matrix) return Real_Matrix;
function Inverse (A: Real_Matrix) return Real_Matrix;
function Determinant (A: Real_Matrix)
  return Real_Base;
```

One version of Solve applies to a single set of equations whereas the other solves several sets in parallel.

The addition of these operations seems to be the very minimum required to make the package complete in any sense. They also make the package really useful for many simple applications.

What about the BLAS?

There is a well-known set of subprograms for linear algebra known as the Basic Linear Algebra Subprograms or BLAS (www.netlib.org/blas). The basic reference implementation is in FORTRAN 77 and there are implementations or bindings for C, C++ and Java and maybe other languages as well.

Indeed, an important original purpose of 13813 was to provide types which would facilitate the writing of a binding to some or all of the BLAS. However, the BLAS cover a vast range of operations most of which are only of interest to the numerical specialist toiling with problems such as tricky differential equations.

It has been suggested that maybe Ada 2005 should contain a binding to the BLAS. But that would be inappropriate for two reasons. One is simply that the BLAS are not defined by an International Standard (they are naturally evolving) and so do not provide a standard foundation. And the other is that they are huge; they could double the size of the Ada Reference Manual (well perhaps not quite). Indeed I have heard them referred to as the Blasted Bloated BLAS.

Note that even the FORTRAN language standard does not include much linear algebra. Inner product and the consequential matrix multiplication is all there is. Thus FORTRAN recognizes the value of a special treatment for inner product but it does not provide any accuracy requirements in the standard but leaves it to the market place.

So our goal is not to provide in Ada itself the full range of stuff for the numerical specialist but simply some handy facilities for the embedded system or real-time programmer who happens to need to solve a few equations and at the same time to provide the basic types to enable access to the

BLAS and other such material to be programmed in a consistent manner.

Of course nothing prevents the implementer from implementing the proposed Ada subprograms, Solve, Inverse and Determinant by a binding to the appropriate BLAS.

Complex Vectors and Matrices

There is also a generic package for complex vectors and matrices. It follows the same style as the real one and it uses the real one as one of its formal parameters. Indeed the formal parameters are worth mentioning since they use the package parameters introduced in Ada 95. The specification of the package starts as follows.

```
with Ada.Numerics.Generic_Real_Arrays,
    Ada.Numerics.Generic_Complex_Types;
generic
  with package Real_Arrays is new
    Ada.Numerics.Generic_Real_Arrays (<>);
  use Real_Arrays;

  with package Complex_Types is new
    Ada.Numerics.Generic_Complex_Types (Real);
  use Complex_Types;

package Ada.Numerics.Generic_Complex_Arrays is
  pragma Pure(Generic_Complex_Arrays);

  type Complex_Vector is array (Integer range <>) of
    Complex;
  type Complex_Matrix is array (Integer range <>,
    Integer range <>) of Complex;
...

```

Note that the formal packages are instantiations of the real array package and the complex number package (the latter is already in the Ada 95 Numerics Annex). Note how they are constrained to be instantiated with the same underlying floating point type. Thus the type Real used to constrain the instantiation of Generic_Complex_Types is that exported from the instantiation of Generic_Real_Arrays.

The bulk of the package is much as one would expect. As well as the arithmetic operations there are also composition and decomposition operations for building complex vectors out of real ones and so on (these are similar to the corresponding operations on scalars in the complex number package). There are also mixed operations for multiplying a complex matrix by a real matrix (again these are similar to the corresponding scalar operations).

Finally, by analogy with the real package, we propose

```
-- Complex_Matrix inversion and related operations
function Solve (A: Complex_Matrix;
  X: Complex_Vector) return Complex_Vector;
function Solve (A, X: Complex_Matrix)
  return Complex_Matrix;
function Inverse (A: Complex_Matrix)
  return Complex_Matrix;
function Determinant (A : Complex_Matrix)
  return Complex;
```

It does not seem necessary to add mixed real and complex for Solve.

Eigenvalues and vectors

An important intrinsic property of a matrix is its eigensystem. Recall that if a matrix **A** and a vector **x** are such that

$$\mathbf{Ax} = \lambda \mathbf{x}$$

then we say that **x** is an eigenvector of **A** and λ is the corresponding eigenvalue. Thus multiplying an eigenvector by the matrix results in a vector in the same direction but whose length is increased by the eigenvalue.

Eigenvalues and vectors turn up in a host of physical situations such as confidence regions in statistics, moments of inertia, the elasticity tensor and so on. In these common situations the matrix **A** is symmetric and the eigenvalues and vectors are all real.

They are so useful that they merit consideration for the standard. In order not to clutter the main packages it is proposed that there be a child package Eigen for the real symmetric case whose specification is

```
generic
package Ada.Numerics.Generic_Real_Arrays.Eigen is
  pragma Pure(Eigen);

  function Values(A : Real_Matrix) return Real_Vector;

  procedure Values_And_Vectors(A: in Real_Matrix;
    Values: out Real_Vector;
    Vectors: out Real_Matrix);

end Ada.Numerics.Generic_Real_Arrays.Eigen;
```

Thus we can write

```
X := Eigen.Values(A);
```

and this assigns the eigenvalues of the real symmetric matrix **A** to the vector **X**. The procedure Values_And_Vectors returns both eigenvalues and eigenvectors through the out parameters.

It was not felt necessary in this basic package to provide special functionality just to return the maximum eigenvalue or all those in a certain range and so on. Such variations are useful if the matrices are very large and time is at a premium.

If the matrix is not symmetric then Argument_Error is raised. This exception is declared in the root package Ada.Numerics; the reader might recall that it is also raised if a function such as Sqrt is given an incorrect argument. Other typical errors in using the vector and matrix packages are that array bounds do not match properly and this raises Constraint_Error as happens with built-in array operations.

The test for symmetry could be approached in two ways. Matrices such as covariance matrices will automatically be symmetric apart from possible errors introduced in their computation. The onus could be placed on the user to ensure that the matrix is exactly symmetric or alternatively some tolerance level could be passed to the eigen subprograms so that they can perform a reasonable check. Passing a tolerance level adds an irritating parameter, raises the issue of how the tolerance should be expressed and gives the user something to fret over. Moreover, the

subprograms would still have to decide just what values should be used for those pairs of components which are not exactly equal.

The approach adopted therefore is to place the onus on the user to ensure that the matrix is exactly symmetric. This could be done for example by taking the mean of the matrix and its transpose. The test in the subprograms can then simply use predefined equality which is guaranteed to deliver the correct answer.

Mathematically, if the matrix is not symmetric then some of the eigenvalues and vectors may be complex. This is where we enter the slippery slope of completeness. In order to handle nonsymmetric real matrices we introduce a child package `Eigen` of the complex package. This child package also caters for complex matrices which are also subdivided into two categories (the Hermitian matrices whose eigenvalues are always real and the general case where they might be complex). Thus altogether this child package has to cater for three different categories. Its specification is

```
generic
package Ada.Numerics.Generic_Complex_Arrays.Eigen is
  pragma Pure(Eigen);

  -- Eigensystem of a general real matrix

  function Values(A: Real_Matrix)
    return Complex_Vector;

  procedure Values_And_Vectors(A: in Real_Matrix;
    Values: out Complex_Vector;
    Vectors: out Complex_Matrix);

  -- Eigensystem of a Hermitian matrix

  function Values(A: Complex_Matrix)
    return Real_Vector;

  procedure Values_And_Vectors(A: in Complex_Matrix;
    Values: out Real_Vector;
    Vectors: out Complex_Matrix);

  -- Eigensystem of a general complex matrix

  function Values(A: Complex_Matrix)
    return Complex_Vector;

  procedure Values_And_Vectors(A: in Complex_Matrix;
    Values: out Complex_Vector;
    Vectors: out Complex_Matrix);

end Ada.Numerics.Generic_Complex_Arrays.Eigen;
```

Some readers might wonder what a Hermitian matrix is. It is the complex analogue of a real symmetric matrix. A complex matrix is Hermitian if its complex conjugate (that is with the sign of the imaginary components changed throughout) is equal to its transpose (that is with rows and columns interchanged). Hermitian matrices have the important property that their eigenvalues are always real. They have practical application in Quantum Mechanics where the eigenvalues give the possible values of the observed variables.

Again the approach has been taken to place the onus on the user to ensure that the matrix is exactly Hermitian. The test in the package can then be exact - the real components have to be exactly equal and the imaginary components have to be exactly of opposite sign (to guarantee the latter may need a slight tweak to the Ada 95 rules - although it is hard

to conceive of hardware for which floating point negate followed by an equality check would not produce the right answer).

Incidentally, Charles Hermite (1822-1901) was a French mathematician and professor at the Ecole Polytechnique. Among other things he proved that the base e of natural logarithms is transcendental.

It is interesting to observe that the Ada overload resolution rules enable the three cases in this complex package to be distinguished without having to introduce different names for the subprograms.

Well maybe it has got a bit complicated now and outside the domain of need of the everyday programmer. But real symmetric matrices deserve to be provided for and all the others follow for the sake of completeness.

A middle way would be to provide for the real symmetric and Hermitian cases only. This would increase the symmetry between the real and complex packages. Moreover, computing the eigensystem of real symmetric and Hermitian matrices is a well-understood and reliable process. But general matrices with possibly complex eigenvalues can be badly behaved (they can be defective which means that they do not have a complete set of linearly independent eigenvectors anyway) and thus can cause computational difficulties. So provision for these general matrices is maybe not appropriate in a standard.

LU Decomposition

If the same matrix \mathbf{A} is used in many calculations (perhaps for solving several sets of equations, computing the determinant and evaluating the eigensystem) then it is traditional to first convert it to LU form in order to save a certain amount of repetitive calculation.

The idea is that we find a lower triangular matrix \mathbf{L} and an upper triangular matrix \mathbf{U} such that

$$\mathbf{A} = \mathbf{LU}$$

A lower triangular matrix is one whose elements above the diagonal are all zero and an upper triangular matrix is one whose elements below the diagonal are all zero. It is conventional for \mathbf{L} to be chosen such that its diagonal elements are all 1.0. We can then store \mathbf{L} and \mathbf{U} together in the same amount of space as occupied by the original matrix \mathbf{A} . It is also conventional to permute the rows of \mathbf{A} to avoid division by small "pivots" but we can ignore this nicety for the moment.

Having decomposed \mathbf{A} into \mathbf{L} and \mathbf{U} the solution of a set of equations is easy since if

$$\mathbf{Ax} = \mathbf{y}$$

then it follows from the fact that $\mathbf{A}^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1}$ that

$$\mathbf{x} = \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{y}$$

Now computing $\mathbf{z} = \mathbf{L}^{-1}\mathbf{y}$ is easy since it just requires back substitution and then $\mathbf{x} = \mathbf{U}^{-1}\mathbf{z}$ is similarly easy using forward substitution.

The LU decomposed form is also a good starting point for evaluating the determinant and finding eigenvalues and vectors.

We have therefore considered whether to include such facilities in our packages. For simplicity we will first consider the real package. We could introduce a private type with unknown discriminants thus

```
type Real_LU (<=>) is private;
```

and then a function to convert a matrix to LU form

```
function Decompose(A: Real_Matrix) return Real_LU;
```

and then we can provide overloadings of functions such as Solve

```
function Solve (LU : Real_LU; X: Real_Vector)
  return Real_Vector;
function Solve (LU : Real_LU; X : Real_Matrix)
  return Real_Matrix;
function Inverse (LU : Real_LU) return Real_Matrix;
function Determinant (LU : Real_LU) return RealBase;
```

In a similar way we can provide overloadings using the decomposed form in the Eigen child package

```
function Values(LU: Real_LU) return Real_Vector;
procedure Values_And_Vectors(LU: in Real_LU;
  Values: out Real_Vector;
  Vectors: out Real_Matrix);
```

The user can now write a sequence such as

```
with Ada.Numerics.Real_Arrays;
use Ada.Numerics.Real_Arrays;
with Ada.Numerics.Real_Arrays.Eigen;
procedure An_Example is
  N : Integer := ...
  Mat : Real_Array(1 .. N, 1 .. N);
  P, Q, R : Real_Vector(1 .. N);
  X, Y, Z : Real_Vector(1 .. N);

  -- compute some value for Mat
  -- and values for vectors P, Q and R;

  declare
    LUMat : constant Real_LU := Decompose(Mat);

    -- Now compute various properties of Mat

    D : Float;
    Inverse_Mat : Real_Array(1 .. N);
    Mat_Eigenvalues : Real_Array(1 .. N);
    ...
  begin
    D := Determinant(LUMat);
    Inverse_Mat := Invert(LUMat);
    X := Solve(LUMat, P);
    Y := Solve(LUMat, Q);
    Z := Solve(LUMat, R);
    Mat_Eigenvalues := Eigen.Values(LUMat);
    ...
  end;
end An_Example;
```

This will be significantly faster than doing the corresponding computation on the raw matrix. Note incidentally how the private type prevents the user from inadvertently using a decomposed matrix instead of a raw one or vice versa. This is a nasty risk with typical FORTRAN libraries.

In a similar way we can provide facilities for the LU decomposition of complex matrices.

A slight problem arises in the case of the eigenvalues of a general real matrix. We would like to add to the complex Eigen child package the following overloadings for this case

```
function Values(LU: Real_LU) return Complex_Vector;
procedure Values_And_Vectors(LU: in Real_LU;
  Values: out Complex_Vector;
  Vectors: out Complex_Matrix);
```

However, in the body of `Generic_Complex_Arrays.Eigen`, the full type declaration of `Real_LU` is not visible and so the details of the decomposed form are not available. (In all the other cases the visibility rules are such that the details are visible.) In order to make the details accessible for this awkward case we therefore provide specific functions in the real package thus

```
function Lower (LU: Real_LU) return Real_Matrix;
function Upper (LU: Real_LU) return Real_Matrix;
function Permutation (LU: Real_LU)
  return Real_Matrix;
```

and for completeness we could add similar functions to the complex package. Of course there is an argument that the user might find it handy to access the decomposed form anyway.

Note however that if we took the middle way for the eigensystems and only provided for real symmetric and Hermitian matrices then the visibility problem would not arise and so we would not need to provide the Lower, Upper and Permutation functions.

The reader should by now have realised that adding the LU decomposition facilities makes the packages quite a lot larger and possibly creates complexity in the mind of the programmer.

Is LU worthwhile?

The heading of this section might give a professional numerical analyst the shivers. Of course it's worth it! LU has been a traditional programming approach for half a century.

But the underlying reason for using LU is simply to save time. It doesn't of itself improve accuracy at all. Of course when computers were slow, it really mattered. Moreover, it only matters if the same matrix is used more than once.

To get some idea of timing, the solution of a set of n linear equations is an n^3 process. On a machine with nanosecond basic instructions, it might take a microsecond for 10 equations. And for 100 equations it might take a millisecond. But a 100 x 100 matrix is huge and if we are in that situation then better call in the numerical analyst after all. So it is clear that if n is sufficiently large for time to matter, then the matrix is so large that it falls outside our domain of simple applications anyway.

Of course, very large matrices occur in the solution of differential equations and similar problems where the size

is dictated by the number of grid points. But these are specialist applications and not the goal of our packages.

But for the real-time embedded user who might perhaps have a handful of equations to solve relating to some positioning information then even if repeatedly solved within a tight loop, it is hard to imagine that timing would be a problem.

So it seems as if LU is unlikely to be worthwhile for these simple packages.

Choices

We have discussed a number of possibilities starting with the basic 13813. It should also be noted that it might be concluded that none of this material should be included in the Amendment after all. So the list of possibilities is

- 0 Do nothing, not even 13813.
- 1 The basic 13813. that is two packages, one for real and one for complex vectors and matrices.
- 2 As 13813 but including Solve, Inverse and Determinant for both real and complex.
- 3 As (2) but adding the real Eigen child package for real symmetric matrices.
- 4 As (3) but adding the complex Eigen child package for Hermitian matrices.
- 5 As (4) but extending the complex Eigen child package to cover all matrices.
- 6 As (2) plus LU decomposition.
- 7 As (3) plus LU decomposition.
- 8 As (4) plus LU decomposition.
- 9 As (5) plus LU decomposition.

The discussion above seems to make it clear that LU decomposition is not worthwhile. On the other hand the basic 13813 is pretty useless. So it really comes down to a choice between 2, 3, 4 and 5 (and 0 of course). In other words whether we should add the eigenvalue child packages. Eigenvalues for real symmetric matrices are pretty handy. Moreover one's sense of orthogonality makes the choice of eigenvalues for the real symmetric case only seem uncomfortable. But the full eigenvalue facility seems unwise as mentioned earlier. This points to 0, 2 and 4 as being sensible proposals.

My personal vote would be for 4, that is including both the Eigen child packages but excluding the eigenvalues of general matrices and also excluding LU decomposition.

Implementation issues

It is the intention that these packages be included in a new section in the Numerics Annex. This means of course that

they do not have to be supported by every implementation. Moreover the rules regarding the Specialized Annexes allow packages to be omitted if they are felt unnecessary for the vendor's market segment. (Of course such an implementation would not be deemed as conforming to the whole annex .)

This is therefore an advantage of providing the eigenvalue features in distinct child packages.

We have said nothing much about accuracy. The current feeling is that the standard should say something crisp about inner product (better than FORTRAN) but leave the accuracy of the Solve, Inverse, Determinant and eigenvalue subprograms as implementation defined.

Conclusions

This paper has outlined a possible extension to Ada for the 2005 Amendment in the form of additional packages in the Numerics Annex for the manipulation of vectors and matrices.

The ARG has also given brief consideration to whether any further additions could be considered for the Numerics Annex. Criteria are that features must be (1) well understood and mature and not in a state of evolution, (2) actually useful, (3) not so trivial that users can do it themselves, and (4) not so hard as to be a real burden to implementers.

Many areas of numerical analysis are still evolving strongly and clearly not ripe for inclusion in an international standard. It has been suggested that perhaps one area that might be considered is the provision of types and operations for the manipulation of polynomials. Most of these would be trivial as are the operations in basic 13813 but the solution of a polynomial equation is not trivial although the techniques are stable and well understood. This seems to be the only candidate for further consideration.

The author welcomes any feedback on the proposals made here, especially from real programmers who might find the facilities useful.

Acknowledgements

The author is grateful to Brian Ford and Zohair Maany of the Numerical Algorithms Group at Oxford and to Catherine West and Joanne Nothard of Qinetiq at Farnborough as well as to members of the BSI Ada Panel and the Ada Rapporteur Group for thought provoking discussions.

References

- [1] John Barnes (2001), Do You Use Generic Arrays and Vectors? - ISO/IEC 13813: 1998, Ada User Journal, vol 22 no 2, p 124.

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

Fax: +33-1-49-70-05-52

URL: www.act-europe.fr

Aonix

Contact: *Marrienne Tournery*

66/68, Avenue Pierre Brossolette, 92247 Malakoff, France

Tel:

Email : info@aonix.fr

Fax: +33-1-41-48-10-20

URL : 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

Praxis Critical Systems Ltd

Contact: *Rod Chapman*

20 Manvers Street, Bath, BA1 1PX, UK

Tel: +44-1225-823763

Email : sparkinfo@praxis-cs.co.uk

Fax: +44-1225-469006

URL : 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

Fax: +44-1249-467-610

URL : 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

Fax: ++33-1-30-12-09-66

URL: 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

Fax: +44-1260-29-14-49

URL: www.tni-europe.com