

ADA USER JOURNAL

Volume 23
Number 4
December 2002

Contents

	<i>Page</i>
Editorial Policy for <i>Ada User Journal</i>	186
Editorial	187
News	189
Conference Calendar	219
Forthcoming Events	225
Articles	
Pascal Leroy <i>"An Invitation to Ada 2005"</i>	230
Luis Santos <i>"Evolutionary Computation in Ada 95: A Genetic Algorithm Approach"</i>	239
Brian Dobbing, Terry Totten <i>"COTS RTOS Suitability for High Integrity Systems"</i>	245
Ada-Europe 2002 Sponsors	248
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

This issue, which completes volume 23 while the year 2002 draws to a close, has its highlight in an article by Pascal Leroy, the ARG rapporteur, which gives us an informative round-up of the status, prospects and key points of the on-going Ada language revision process. It is our intention, in collaboration with WG9 and the ARG itself, to closely follow the final, public period of the process with articles that illustrate specific aspects and features of the proposals tabled for the revision.

This issue also features the usual comprehensive News and Events sections, which keep us up to date with what goes on within and in the vicinity of the Ada world, and two further technical articles. In one, Luis Santos, from the University of Algarve in Portugal, tells us how Ada can match the demand (and exploit the benefit) of the evolutionary computation theory. In the other, Brian Dobbing, of Praxis Critical Systems, and Terry Totten of QinetiQ, in the UK, discuss the suitability of commercial-off-the-shelf real-time operating systems for use in high integrity avionics applications.

Let me close by wishing all readers, contributors and collaborators of the AUJ and their families all the best for their personal and professional life in the New Year.

*Tullio Vardanega
Padova
December 2002
Email: tullio.vardanega@math.unipd.it*

News

Dirk Craeynest (ed)

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

Contents

	page
Ada-related Events	189
Ada Semantic Interface Specification (ASIS)	190
Ada and Education	190
Ada-related Resources	192
Ada-related Tools	192
Ada-related Products	201
Ada and Linux	205
Ada and Microsoft	205
References to Publications	205
Ada Inside	209
Ada in Context	211

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]

Oct 29 - Ada UK User Group Autumn Conference

*From: John Robinson
John@JohnRobinsonAndAssociates.com
Date: Sat, 7 Sep 2002 15:21:56 +0100
Organization: John Robinson And Associates Ltd
Subject: CNF: Embedded Systems Club and Ada UK User Group Autumn Conference
Newsgroups: comp.lang.ada*

Announcing the Autumn Ada UK User Group conference, in association with the Embedded Systems Club, to be held in Swindon (UK) on October 29th 2002.

Sessions of interest to Ada users include:

"GNAT's Hidden Jewels", a look at the more advanced features of GNAT PRO.

"Correctness by Construction", describing the development of a secure certification authority (CA) for smart cards using the SPARK Ada subset.

"Ada Code Generation Using Model Driven Architecture and the UML", describing work on the F16 Mission Computer using Model Driven Architecture, UML and Ada.

"Ada-UML Profile", and associated panel session, examining the state-of-the-art in Ada code generation from UML and proposing the development of a standardised Ada code generation profile for the UML.

"A Standard Container Library for Ada", describing the issues related to such a project and reporting on the workshop on this topic held recently at Ada Europe.

"Rational Apex Ada and Rational TestRT", a short vendor presentation from Rational on enhanced Ada support.

"Recruiting & Retaining Ada Staff", a birds-of-a-feather session on the issues and problems related to recruitment and retention of Ada programmers. This is session also supports a pre-event survey which has been the subject of another post.

There are also several other vendor presentations from tool vendors who support Ada.

Full details, [...], can be found at <http://www.AdaUK.org.uk>.

[See also same topic in AUJ 23-3 (Sep 2002), pp.125-126. -- dc]

John Robinson, Conference Director,
John@JohnRobinsonAndAssociates.com

*From: John Robinson
John@JohnRobinsonAndAssociates.com
Date: Thu, 24 Oct 2002 00:08:40 +0100
Organization: John Robinson And Associates Ltd
Subject: CNF: Ada UK User Group Autumn conference - last call for delegates
Newsgroups: comp.lang.ada*

The joint Ada UK User Group & Embedded Systems Club Autumn conference is being held in Swindon, UK on Tuesday 29th October 2002.

Late bookings are still being taken at <http://www.AdaUK.org.uk> for this low cost conference and exhibition, which has already attracted over 110 attendees.

The conference features 2 streams of 22 sessions, including several sessions designed to give delegates the opportunity to play an active role in the conference. For example, Silver Software will be facilitating a session exploring the trials and tribulations of recruiting and retaining Ada software engineers while ARTiSAN, Kennedy Carter and others will be examining the potential for a standardised Ada-UML Profile.

Two tutorials on "Test-First Design" (Kevlin Henney, Curbralan) and "Principles of High Availability Embedded Systems Design" (David Kalinsky, OSE) provide a training element to the event, while technical presentations by Praxis, ADI, IPL, ACT Europe, Programming Research and I-Logix help

to make this a conference with significant technical content.

The exhibition has attracted key players in the Ada and embedded systems markets including Wind River, Vsystems, Telelogic, Silver Software, Programming Research, Praxis, PolySpace, OSE, Objektum, LDRA, Kennedy Carter, John Robinson & Associates, IPL, I-Logix, Headway Software, Green Hills Software, First Matrix, ARTiSAN and Aonix.

At the exhibition a number of product vendors will announce and demonstrate new products and/or significant product upgrades, including IPL (Testing) and Telelogic (UML). A number of product announcements will be of particular interest to Ada users. Ada-related product announcements will be made by Rational Software, Headway Software, PolySpace Technologies, First Matrix and OSE.

Both Headway Software and Polyspace Technologies are new to the Ada market, challenging the widespread misconception that the Ada market has stagnated.

The mix of exhibitors, sponsors, speakers and delegates attracted to this event represent a wide spread of interests across the embedded systems industry. [...]

<http://www.EmbeddedSystemsClub.com>

<http://www.AdaUK.org.uk>

To receive information on our 2003 events, register as an associate member via the same websites. Associate membership is free of charge.

Dec 8-12 - ACM SIGAda 2002 Conference

*From: Clyde Roby <roby@ida.org>
Date: Fri, 8 Nov 2002 16:56:05 -0500
Subject: ACM SIGAda 2002 -- Early Registration Dates Extended
To: SIGAda-Announce@acm.org*

SIGAda 2002 Annual International Conference

Holiday Inn Houston/NASA, 1300 NASA Road One, Houston TX

<http://www.acm.org/sigada/conf/sigada2002>

Tutorials Dec.8-9, Conference Dec.10-12, Exhibits Dec.10-11, Workshops Throughout

Keynote Speakers: Robert Dewar (Ada Core Technologies), Charles McKay (University of Houston), Michael McEvelley (Decisive Analytics Corporation).

Invited Speaker: Robert Carey (Lawrence Livermore National Laboratory).

SIGAda 2002 Great Values! Pay-for-One-Tutorial/Get-Charts-For-All (a SIGAda tradition). Free workshops(s) participation with conf. Registration. Career-enhancing tutorials at affordable rates. Grants for Students and Educators. Great hotel across from NASA at a great price (\$73).

Check out the convenient Sunday/Monday tutorial schedule: Software Engineering Topics: Design Patterns, ...; Ada Language Topics: Intro to Ada 95, SPARK, CORBA, ...; IT Security Topics: Intro to the Common Criteria for IT ...; Embedded/Real-Time Topics: Embedded Ada95, Real-Time Ada95, ...

The conference features a single full track devoted to Ada software engineering and education topics as well as embedded systems. In addition, WG9 will conduct workshops to address the five major areas where enhancements to the Ada Language are projected.

Sponsored by ACM SIGAda, In Cooperation With Ada-Europe and ACM SIGAPP, SIGCAS, SIGCSE, SIGPLAN, and SIGSOFT.

Conference Chair: Salih Yurttas (Texas A&M Univ), yurttas@cs.tamu.edu

Program Committee Chair: John McCormick (Univ of Northern Iowa)

Exhibits Chair: Ben Brosgol (Ada Core Technologies), brosgol@gnat.com

Visit our website at

<http://www.acm.org/sigada/conf/sigada2002> for full technical program, tutorial, & workshop descriptions, the registration form, hotel info, and other goodies.

Jun 16-20 - Ada-Europe 2003 Conference

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

Date: 10 Sep 2002 23:22:06 +0200

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

Subject: CfPapers, Reliable Software Technologies, Ada-Europe'2003

Newsgroups:

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

8th International Conference on Reliable Software Technologies - Ada-Europe 2003, 16 - 20 June 2003, Toulouse, France

<http://www.ada-europe.org/conference2003.html>

[...] The 8th International Conference on Reliable Software Technologies (Ada-Europe 2003) will take place in 2003 in Toulouse, France. 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. There will be a special session on avionics and space, including the use of Ada in this realm. [...]

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 17, 18 and 19. Vendors of software products and services should contact the Exhibition Chair Frédéric Dumas, frederic.dumas@c-s.fr, 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-Europe'2003 Publicity Co-Chair

Ada Semantic Interface Specification (ASIS)

ASIS on RedHat Linux

From: Dean Sutherland

<dfsuther@cs.cmu.edu>

Date: Tue, 22 Oct 2002 10:03:18 -0400

Subject: ASIS for Gnat on RedHat

To: SIGAda-ASIS@acm.org

I'm having trouble tracking down a version of ASIS that is compatible with the version of Gnat now shipping with every new RedHat Linux system. I'd really appreciate any pointers you folks may be able to provide.

From: Robert Dewar <dewar@gnat.com>

Date: Tue, 22 Oct 2002 10:49:44 -0400

Subject: Re: ASIS for Gnat on RedHat

To: SIGAda-ASIS@acm.org

There is currently no version of ASIS that matches the version that Redhat is shipping. ACT will supply the necessary sources at some point in the future, but this is very low priority for us so do not hold your breath.

We would advise sticking to 3.14p. The version Redhat is shipping has a significant number of known errors and is bleeding edge stuff that you probably want to stay away from if you are not in hobbyist mode.

From: Bonnin Hugues

<hugues.bonmin@thales-avionics.com>

Date: Fri, 25 Oct 2002 09:22:48 +0200

Organization: Thales-avionics

Subject: Re: ASIS for Gnat on RedHat

To: SIGAda-ASIS@acm.org

Is that due to the fact that Redhat is shipping GCC, which contains GNAT since 3.2, or is due only to a Redhat choice to ship a specific GNAT? In other words, does it mean that GCC-GNAT couldn't be synchronized with any ASIS delivery, and that only GNAT 3.xpp are synchronized with, or not?

From: Robert Dewar <dewar@gnat.com>

Date: Sun, 10 Nov 2002 09:40:38 -0500

Subject: Re: ASIS for Gnat on RedHat

To: SIGAda-ASIS@acm.org

[...] we know that Red Hat is shipping some kind of GNAT as part of GCC 3.2, but that's an untested version, and we have no idea how they built it or packaged it, or exactly which sources they used etc. There is as far as we (ACT) know, no version of ASIS sources that is compatible with the set of sources in GCC 3.2. It's on the list of things to do to upload a set of compatible ASIS sources and GNAT sources to the FSF gnu.org site, but

a) this set of sources won't be compatible with GCC 3.2, but rather with the GCC 3.3 snapshots.

b) although this is on our list of things to do, it is unfunded work, so it is hard to give any specific deadline (it gets pushed back by any funded work).

Note that the real issue here is that GCC 3.2 GNAT is still not in good enough shape for ACT to support a corresponding GNAT Pro version. We make the latest cutting edge sources of GNAT available for hobbyists etc, but they are not suitable for serious use and there simply is no corresponding set of ASIS sources yet, and likely won't be until we qualify and release a GCC 3 compatible version of GNAT.

Robert Dewar, Ada Core Technologies

Ada and Education

Resources to Learn Ada

From: Preben Randhol <randhol@pvv.org>

Date: Wed, 4 Sep 2002 10:50:20 +0000 (UTC)

Organization: Norwegian university of science and technology

Subject: Re: New to Ada

Newsgroups: comp.lang.ada

> I've never used Ada but I like to check new things out. C++ and C appeal to me Java not so much. [...]

Look at:

<http://www.adaic.com/whyada/WhyAda.html>

To learn Ada95 go to:

<http://www.adapower.com/learn/>.

Here you can find free on-line books. I have added a list of links you can visit below. [...]

Short Intro I:

<http://www.cl.cam.ac.uk/~mgk25/ada.html>

Short Intro II:

<http://www.adaic.com/whyada/>

Main Ada site:

<http://www.adapower.com/>

Online Ada 95 book:

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

Online Ada 95 book II:

<http://burks.bton.ac.uk/burks/language/ada/ada95.pdf>

Short Ada tutorial:

<http://goanna.cs.rmit.edu.au/~dale/ada/aln.html>

A Linux and Ada book:

<http://www.vaxxine.com/pegasoft/homes/book.html>

Other books:

<http://www.adapower.com/learn/>

Ada 95 book reviews:

<http://www.seas.gwu.edu/~mfeldman/ada95books.html>

ISO Reference Manual:

<http://www.adapower.com/rm95/>

ISO Rational Manual:

<http://www.adapower.com/rationale/>

Quality and Style Guide:

<http://www.adaic.com/docs/95style/html/cover.html>

GNU Ada Compiler(GNAT):

<http://www.gnat.com/>

GNAT for Linux, Dos, NetBSD, OS/2, SCO:

<http://www.gnuada.org/>

GNAT for Windows:

<http://home.trouwweb.nl/Jerry/>

GTKAda ToolKit (GTK+):

<http://libre.act-europe.fr/GtkAda/>

GNADE Project:

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

Some software projects:

<http://freshmeat.net/browse/163/>
and

http://sourceforge.net/softwaremap/trove_list.php?form_cat=163

Course and Training Material in French

[Translated from French: -- dc]

From: Daniel Feneuille <feneuille@univ-aix.fr>

Date: Sat, 21 Sep 2002 09:45:32 +0200

Organization: IUT Aix

Subject: [ada-france] Cours, TD, TP Ada

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

On the occasion of the new teaching year I have actualised my material for the Ada courses I teach since 1988 at the IUT of Aix [IUT = Institut Universitaire de Technologie -- dc] and which have been

available on the Net. I thank all those who sent me their comments and I hope to have taken into account their proposals.

You can find these new documents (3 zip files) both at ACT and at the IUT d'Aix.

http://libre.act-europe.fr/french_courses
(to consult and download)

ftp://paprika.iut.univ-aix.fr/pub/cours_ada_DF
(only to download)

Getting Started with Distributed Ada

From: Adrian Hoe

<mailbox@adrianhoe.com>

Date: Tue, 24 Sep 2002 01:13:45 +0800

Subject: Distributed programming in heterogeneous platforms

Newsgroups: comp.lang.ada

I am having a project which is going to be developed using gnat and glade. This is a distributed computing involved heterogeneous platforms.

For a start, a small heterogeneous platforms will be setup with two Sun Solaris workstations and a dual cpu Intel Xeon workstation running Linux.

Before bumping my head into the project, I like to gather as much information about distributed programming in heterogeneous platforms.

Can anyone provide me pointers, experience, websites, sample codes and etc. for reference?

From: Joe Simon <jb.simon@lmco.com>

Date: Mon, 21 Oct 2002 18:04:22 -0400

Organization: Lockheed Martin Corporation

Subject: Distributed Ada (Annex E)

Newsgroups: comp.lang.ada

I have been working for many years on implementing distributed systems for various uses. Every one of them has had specialized data communications schemes worked out for them. I was just reading a paper on distributed systems utilizing Ada 95 and Annex E, it said that the Ada distributed processing model made creating distributed systems in Ada "Painless".

I have Ada 95 and would like to evaluate for myself how "painless" it is. When I get on the net to search for information, I find references to software, such as "Glade" from ACT, or "Adept" from Texas A & M.

However, I can't seem to find any literature that provides an "Ada distributed Systems for the clueless". Is all I need to put together a small distributed system test (the equivalent to "Hello World", but displayed on another node) is GNAT, or do I need one of these other packages?

From: Dale Stanbrough

<dale@cs.rmit.edu.au>

Date: Tue, 22 Oct 2002 14:47:40 +1000

Organization: RMIT

Subject: Re: Distributed Ada (Annex E)

Newsgroups: comp.lang.ada

You need to install gnat and glade. Do your normal development in gnat, then a final (few?) compiles with glade, which distributes everything according to the distribution file.

It really is painless. I had an assignment for first year students which used it for a distributed chat/multiroom system.

Couldn't have been simpler.

From: Samuel Tardieu <sam@rfc1149.net>

Date: Tue, 22 Oct 2002 09:54:29 +0200

Subject: Re: Distributed Ada (Annex E)

Newsgroups: comp.lang.ada

Get GNAT and GLADE from

<ftp://cs.nyu.edu/pub/gnat/>
and read GLADE manual, which contains a tutorial.

From: Preben Randhol <randhol@pvv.org>

Date: Tue, 22 Oct 2002 08:00:09 +0000

(UTC)

Organization: Norwegian university of science and technology

Subject: Re: Distributed Ada (Annex E)

Newsgroups: comp.lang.ada

And read the Programming Distributed Systems introduction you find at this location:

http://libre.act-europe.fr/Software_Matters/main.html

Skip all that is known and later it talks about Glade if I remember correctly.

From: David Marceau

<davidmarceau@sympatico.ca>

Date: Sat, 02 Nov 2002 20:37:01 -0500

Subject: Re: Distributed Ada (Annex E)

Newsgroups: comp.lang.ada

Please also check out the Ada Web Server(aws) sources and its

documentation section 3.6

Communication. It is certainly worthwhile since you might even find it to be simpler to use than GLADE :)

<http://libre.act-europe.fr/aws/>

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: colbert@abssw.com (Ed Colbert)

Date: 23 Oct 2002 15:40:21 -0700

Subject: Public Ada 95 Class During Week of 16 December 2002 in Carlsbad CA

Newsgroups: comp.lang.ada

Absolute Software will be holding a public Ada 95 class on 16-20 December 2002 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 classes on object-oriented

methods and other object-oriented languages.) [...]

*From: Joyce L. Tokar
<tokar@attglobal.net>*

*Date: Thu, 24 Oct 2002 09:59:30 -0700
Subject: One-Day Public Introduction to
Ada 95 Training Course
To: team-ada@acm.org*

DDC-I is offering a one-day Introduction to Ada 95 training course on 13 Nov 2002 in Phoenix AZ. This course is being offered in conjunction with a two-day public DO-178B training course being conducted by TekSci in Phoenix, AZ on 14-15 Nov 2002. You can find a full description of the course on our website, www.ddci.com -- click through to the main page, left column.

Joyce Tokar, Vice President, DDC, Inc.,
400 N 5th St, Suite 1050, Phoenix, AZ
85004, 602-275-7172, 602-252-6054
(FAX), jlt@ddci.com

Ada-related Resources

Open Directory Project

*From: Samuel Tardieu <sam@rfc1149.net>
Date: Tue, 03 Sep 2002 15:58:50 +0200
Subject: Open Directory Project
Newsgroups: fr.comp.lang.ada*

[Translated from French: -- dc]

The "Open Directory Project" (ODP) is the largest existing directory on the WWW which classifies the resources manually, thanks to volunteer editors.

The ODP has an "Ada" section in French: <http://dmoz.org/World/Fran%e7ais/Informatique/Programmation/Langages/Ada/> which currently contains 15 links.

If you know Ada resources in French that are not yet present, don't hesitate to use the link "Proposer un site" in order to have them added.

Ada Articles on New Software Forum

*From: Jim Rogers
<jimmaureenrogers@worldnet.att.net>
Date: Sat, 14 Sep 2002 15:33:55 GMT
Subject: New Software Forum
Newsgroups: comp.lang.ada*

Yesterday I became aware of a new software forum starting up at <http://www.crystalcode.com/codemage/MainMenu/Home/Welcome.php> I have had some conversations with the web master at this site. I find that they are open to articles and discussions concerning Ada. I plan to write some Ada articles for this site, and possibly help them maintain links to existing high quality Ada web sites.

I post this so that anyone else who wants to help me expose the wider development world to Ada can help out.

*From: Daniel Dudley <dudley@online.no>
Date: Sun, 15 Sep 2002 16:57:41 GMT
Subject: Re: New Software Forum
Newsgroups: comp.lang.ada*

IMHO, this is a good idea, Jim. Exposing Ada on non-Ada-specific sites will surely open the eyes of many programmers.

BTW, I found your Concurrent Programming article to be well-written and enlightening. Thanks.

[See also "Introducing Ada on CodeMages Software Forum" further in this AUJ issue. -- dc]

Another Ada Resources Page

*From: Peter Hermann
<ica2ph@csv.ica.uni-stuttgart.de>
Date: Wed, 9 Oct 2002 09:04:43 +0000
(UTC)*

*Organization: Comp.Center (RUS), U of
Stuttgart, FRG*

*Subject: resources_on_ada
Newsgroups: comp.lang.ada*

http://www.csv.ica.uni-stuttgart.de/homes/ph/resources_on_ada.html

I spent some time refreshing it.

Peter Hermann, (49)0711-685-3611,
fax3758, Pfaffenwaldring 27, Raum 114,
D-70569 Stuttgart, Uni
Computeranwendungen

Ada-related Tools

Charles - Container Library

*From: Matthew Heaney
<mheaney@on2.com>
Date: Wed, 18 Sep 2002 10:58:46 -0400
Subject: Re: periodicity
Newsgroups: comp.lang.ada*

[Extracts from several messages. -- dc]

I have indeed written a library that may be of interest to Ada users.

Charles is a container library, modeled closely on the C++ STL.

You can download the library from my home page:

<http://home.earthlink.net/~matthewjheaney/> [...]

I make drops about once a week, so you may wish to check in periodically to get the latest version. [...]

The Charles container library also has lists (both singly- and doubly-linked), for homogeneous elements. [...] All the containers support both active and passive iteration. Both lists support bidirectional iteration (although for the singly-linked list this isn't terribly efficient, as you might imagine).

[...] Charles, modeled on the C++ STL, [...] has all the container types in that library (vectors, deque, lists, sets, multisets, maps, and multimaps). [...]

*From: mheaney@on2.com (Matthew Heaney)
Date: 22 Oct 2002 07:11:37 -0700
Subject: Re: Beginner problem: variable array size
Newsgroups: comp.lang.ada*

[In response to a question on how to start using Charles when you're not used to the C++ STL? -- dc]

I have written documentation for the vector container, which is both a tutorial and a reference manual. You may be able to use this to get started.

<http://home.earthlink.net/~matthewjheaney/charles/charles-vectors-unbounded.html>

There's also a new release (20021021) of the library, containing changes to the vector package [and better error checking].

<http://home.earthlink.net/~matthewjheaney/charles/> [...]

Drop me a line to let me know whether the documentation is adequate.

PragmARC - PragmAda Reusable Components

*From: Jeffrey Carter <jrcarter@acm.org>
Date: Wed, 25 Sep 2002 01:17:38 GMT
Subject: New release of the PragmAda Reusable Components
Newsgroups: comp.lang.ada*

A new release of the PragmARCs is now available from

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

[See also same topic in AUJ 22-4 (Dec 2001), p.198. -- dc]

The main features of this release are the elimination of the restriction on instantiating data structures with scalar elements and the addition of context information to the iterators. The mirror at www.adapower.com will hopefully be upgraded soon.

Jeffrey R. Carter, PragmAda Software Engineering

Work on Ada Issue 302 - Container Libraries for Ada 200Y

*From: Randy Brukardt
<randy@rrsoftware.com>
Date: Thu, 19 Sep 2002 14:42:11 -0500
Subject: Re: status of Ada STL?
Newsgroups: comp.lang.ada*

> I've seen comments in Ada Letters and elsewhere that seem to indicate that there is some official recognition of the need to have *some* kind of library(ies).

There is an open AI for container libraries for Ada 200Y.

[See <http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-00302.TXT> -- dc]

The problem is, everyone is willing to write an interesting library, but there is much less enthusiasm for the hard work it takes to make something suitable for standardization. (Compare the documentation for the typical library with the RM for Ada.Strings.Fixed, and you'll see what I mean). Then, compound that with getting everyone to agree on the requirements and the solution... [...]

From: Randy Brukardt

<Randy@RRSoftware.Com>

Date: Wed, 2 Oct 2002 18:25:12 -0500

Subject: Re: [Ada-Comment] Update to AI-302: unprotected bounded lists

To: 'Ada-Comment List' <ada-comment@ada-auth.org>

[...] We discussed container libraries (AI-302) briefly in Vienna. No one objected to the general idea.

[A high level discussion on desirable packages] was done quite a few meetings back. The ARG essentially decided that it did not have enough information to decide on which packages are appropriate to add. And we definitely do not want to be in the business of designing packages. Rather, we decided to ask the community to propose packages. This was in the call for APIs (<http://www.adaic.org/news/call4apis.html>). [...]

Randy Brukardt, ARG Editor

From: Martin Dowie <martin.dowie@no-sp-am.btopenworld.com>

Date: Sat, 28 Sep 2002 15:18:17 +0000 (UTC)

Subject: Re: status of Ada STL?

Newsgroups: comp.lang.ada

Well, there is definitely the ASCL Working Group set up (see Matt's message near the start of this thread for a link to the work he has been doing on this).

From: Jeffrey Carter <jrcarter@acm.org>

Date: Sun, 29 Sep 2002 19:02:17 GMT

Subject: Re: status of Ada STL?

Newsgroups: comp.lang.ada

It appears that there will only be a common data-structures library supported by most vendors if the ARG receives an API proposal for one.

That's a lot of work someone will have to do to get it into the form of ARM sections.

If the ARG receives only one such proposal, that will by default form the basis of the standard library. The best thing for Ada would be for people to prepare such proposals. With multiple proposals we are more likely to get a decent standard library. Debating societies such as we have seen here have yet to accomplish anything useful.

From: Ehud Lamm <ehudla@openu.ac.il>

Date: Thu, 3 Oct 2002 16:31:45 +0200

Subject: Re: [Ada-Comment] Update to AI-302: unprotected bounded lists

To: 'Ada-Comment List' <ada-comment@ada-auth.org>

> [...] a submission of a serious proposal requires RM-style wording, and that's a lot of work.

A few of us are working toward that goal. The group started following the Ada-Europe workshop concerning a standard container library for Ada. I created a mailing list for our discussions, so as not to disturb the discussions on Ada-Comment. When we have a decent AI to propose, we will of course submit it.

If anyone is interested in joining the mailing list, let me know.

From: Matthew Heaney

<mheaney@on2.com>

Date: Wed, 2 Oct 2002 18:17:36 -0400

Subject: Re: status of Ada STL?

Newsgroups: comp.lang.ada

If you're interested in an Ada95 port of the C++ STL, then you can try the Charles container library:

<http://home.earthlink.net/~matthewjheaney/charles/>

[And from another message: -- dc]

I plan on submitting my own proposal for AI-302, using the subsystem name "ada.containers".

From: Jeffrey Carter <jrcarter@acm.org>

Date: Wed, 09 Oct 2002 11:57:34 -0700

Subject: Re: [Ada-Comment] Update to AI-302: unprotected unbounded bags

To: Ada-Comment List <ada-comment@ada-auth.org>

The components I am proposing are the data-structure components from the PragmAda Reusable Components with some things given different names (based on advice from Randy Brukardt). The PragmARCs have existed for many years, first in Ada 83 and now in Ada 95. They have been extensively tested through use on a number of real world projects. The discussion of each component refers to the corresponding PragmARC as the reference implementation.

It would be great if the ARG would review existing libraries and choose one for standardization. That was what I was hoping would happen as a result of my original message that started this AI. Tell me that the PragmARCs are it as long as I make certain changes "wrt naming etc." and write the boring documentation and I'll be very happy.

However, Randy assures me that such a thing will not happen, so I am formalizing my proposal that the PragmARCs be used as the basis for standard components.

Ada on ARM-based bare board

From: Tucker Taft <stt@sofcheck.com>

Date: Thu, 29 Aug 2002 12:03:51 -0400

Organization: SofCheck, Inc.

Subject: Re: Ada 95 for an ARM-based bare board?

Newsgroups: comp.lang.ada

Green Hills Software supports both Ada and ARM. You should certainly check with them whether they currently have an Ada offering certified for the ARM. Even if they don't, they might be able to create something. Start at: <http://www.ghs.com>

Note also that SofCheck, Inc. has a validated Ada 95 compiler that uses optimized ANSI C as its intermediate representation. This has been successfully used for relatively quick ports to new architectures which have a decent C compiler. If you are interested in pursuing this, please contact us.

Tucker Taft, President, SofCheck, Inc.

(SofCheck now owns and maintains the AdaMagic(tm) technology developed by Intermetrics, Inc., aka AverStar, AverCom/Titan)

From: Robert A Duff

<bobduff@shell01.TheWorld.com>

Date: Fri, 30 Aug 2002 14:19:25 GMT

Subject: Re: Ada 95 for an ARM-based bare board?

Newsgroups: comp.lang.ada

> But one general question, what about run-time? We have a bare board with no OS planned. To my limited knowledge to translate from Ada to ARM is not the main problem here.

The run-time system is written in Ada. It needs a small amount of configuration for the target. If you have an RTOS and want Ada tasks mapped to RTOS threads, it requires more substantial work (which we also have done in one case). But you have no OS, so it's pretty easy.

Bob Duff, bobduff@sofcheck.com, Vice President, SofCheck, Inc.

GNAT IDEs

From: jim hopper

<hopperj@macconnect.com>

Date: Mon, 02 Sep 2002 18:19:48 -0400

Subject: Re: The Ada experience.

Newsgroups: comp.lang.ada

[On complaints that the free IDEs for GNAT are bad: -- dc]

Depends upon your system. The integration that we have achieved between GNAT and Apple's Project Builder IDE and Interface Builder (GUI Builder) in MacOS X 10.2 which was just released is VERY nice and very much on par with the C/C++ stuff. We use the same tools, debuggers, etc. We have some things to work out yet, but its the best environment I have ever worked in for development so far.

I also thought the SGI tools were very nice for GNAT.

*From: Preben Randhol <randhol@pvv.org>
Date: Wed, 4 Sep 2002 10:40:13 +0000
(UTC)
Organization: Norwegian university of
science and technology
Subject: Re: The Ada experience.
Newsgroups: comp.lang.ada*

Nowadays the wrapping is the most important I guess. If the content is bad or buggy or a health risk doesn't matter as long as it has a nice box flashy cover and cool stickers on it.

But speaking of GNAT [...] read this:

http://www.gnat.com/texts/news/news_gps.htm
and

http://www.gnat.com/texts/products/GPS_tour.pdf
for a tour with screenshots. Hopefully there will be a public version too.

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

Gnatfe - Front End to Multiple Gcc and Gnat Installations

*From: Simon Wright
<simon@pushface.org>
Date: 20 Oct 2002 17:38:13 +0100
Subject: Re: Problems by GNAT installation
on Linux Mandrake OS
Newsgroups: comp.lang.ada*

[On avoiding problems with multiple gcc installations on one machine. -- dc]

Or [...] see <http://www.pushface.org/gnatfe/> for a way of having several GNAT installations at the same time and choosing which to use on the basis of an environment variable.

Free Intel Reverse Compiler/Disassembler

*From: marlinsmeadow@aol.com
(MarlinsMeadow)
Date: 29 Oct 2002 15:24:11 GMT
Subject: Free Intel Reverse
Compiler/Disassembler available
Newsgroups: comp.lang.ada*

A free Intel Reverse Compiler/Disassembler is available from the website below. The purpose of which is to decompiler raw Intel machine bytes back into machine source language. It runs on Windows but can reverse compile any binary so long as it is Intel code e.g. Win, Linux, QNX, etc.

There is also information about our new Ada project on the Website.

<http://members.aol.com/marlinsmeadow/index.htm>

David Moore

IEEE 754 Numerics Model and Ada

*From: Juergen F H Winkler
<jwinkler@acm.org>*

*Date: Thu, 26 Sep 2002 10:33:37 +0200
Subject: [Ada-Comment] Proposal for new
numerics model*

To: ada-comment@ada-auth.org

*Topic: IEEE 754 numerics model;
reference: RM95-G.2 [...]*

IEEE 754 is supported by most microprocessors in use today. It has more useful features than are required by the strict mode of RM95.

Examples of such features are: different rounding modes; infinities and NaNs; tight accuracy requirement for sqrt (Intel processors e.g. implement additional functions). [...]

Ada would be the first language to let the user use the IEEE 754 features of the processor. Up to now they can only be used in assembler.

It would therefore be useful if Ada contained a numerics model with all the properties of IEEE 754. [...]

Prof. Dr. Juergen F H Winkler, Institut fuer Informatik,

Friedrich-Schiller-Universitaet, Jena, Germany, winkler@informatik.uni-jena.de, <http://psc.informatik.uni-jena.de>

*From: dewar@gnat.com (Robert Dewar)
Date: Thu, 26 Sep 2002 14:24:42 -0400
(EDT)*

*Subject: Re: [Ada-Comment] Proposal for
new numerics model*

To: ada-comment@ada-auth.org

[Extracted from several messages. -- dc]

Regarding IEEE fpt, a lot of work has been done in this area. You should familiarize yourself with the work of the NRG in this area, and you should probably also read Sam Figueroa's thesis. This is a PhD thesis from the department of CS at New York University, 1999, Robert Dewar, advisor. This same advice applies to anyone else who wants to discuss this issue technically.

Note that Sam's thesis contains a completely worked out proposal in this area (IEEE754 support in Ada).

*From: John Halleck
<John.Halleck@utah.edu>
Date: Thu, 26 Sep 2002 13:34:52 -0600
(MDT)*

*Subject: Re: [Ada-Comment] Proposal for
new numerics model*

*To: Ada-Comment List <ada-
comment@ada-auth.org>*

http://www.cs.nyu.edu/csweb/Research/Theses/figueroa_sam.pdf

for the pdf version. Back up one for other options.

Simple JPEG Library

*From: Patrice Freydriere <frett27@free.fr>
Date: Sun, 23 Jun 2002 20:49:31 +0200
Subject: Simple JPEG Lib
Newsgroups: fr.comp.lang.ada*

[Translated from French: -- dc]

A binding to the JPEG library which allows reading and writing of JPEG files is available at the address <http://pfreydiere.free.fr>

Interested in feedback!

GtkAda Tutorial and Reference Manual

*From: Adrian Knoth
<adi@drcomp.erfurt.thur.de>
Date: 16 Sep 2002 13:09:25 GMT
Organization: Modern Electronics
Subject: Re: gtkada
Newsgroups: comp.lang.ada*

> Does anybody know where i could find a tutorial or a reference manual about gtkada?

There is a reference-manual for GtkAda which comes along with the library itself.

You might also want to have a closer look at glade, a GUI-builder for Gtk (and even GtkAda), those giving you much help in developing GUIs.

There are some examples for GtkAda in the examples-directory.

After all, using GtkAda is really nice and straight-forward. You can even learn it by reading the include-files (in most cases, *.ads is sufficient, they are well-documented and because of strict typing the API is easy to understand for programmers)

[And from another message: -- dc]

> Be sure to understand the way you pack widgets in Gtk+/GtkAda. One do not use fixed positions as for example Delphi does.

Yes. I'd like to put emphases on this. Having those unfixed widgets give you much freedom in resizing the window. On the other side you'll have to use paddings if you want some space between objects.

*From: David C. Hoos
<david.c.hoos.sr@ada95.com>
Date: Mon, 16 Sep 2002 08:03:36 -0500
Subject: Re: gtkada
Newsgroups: comp.lang.ada*

<http://libre.act-europe.fr/GtkAda/GtkAda-tutorial/GtkAda-tutorial.htm>

http://libre.act-europe.fr/GtkAda/docs/gtkada_ug.html

http://libre.act-europe.fr/GtkAda/docs/gtkada_rm_toc.html

TAP - Thick Ada-Prolog bindings

*From: Alexandre E. Kopilovitch
<aek@vib.usr.pu.ru>
Date: Wed, 4 Sep 2002 01:19:34 +0400
(MSD)
Subject: ANN: Release 0.2 of TAP (Thick
Ada-Prolog) bindings
Newsgroups: comp.lang.ada*

New release (0.2) of TAP (Thick Ada-Prolog) bindings replaced previous one at <http://www.tarkvara.com/tap>

[See also same topic in AUJ 23-3 (Sep 2002), p.134. -- dc]

This is a bugfix release, 2 bugs (both in peripheral parts of the bindings) were fixed, and some internal code was slightly changed for (hopefully) better readability (See Release Notes for more detailed list of changes). No new features are provided with this release.

[...] Send error reports, suggestions, etc. to: aek@acm.org or aek@vib.usr.pu.ru

Alexander Kopilovitch, Saint-Petersburg, Russia

Xlib and X11R6 Bindings

From: Caffeine Junky
<nospam@hotmail.com>

Date: Thu, 12 Sep 2002 23:42:27 GMT
Subject: Xlib and X11R6 bindings
Newsgroups: comp.lang.ada

Recently I've been spending some time learning to use the Xlib that comes with XFree86 (version 4.1.0).

I know there's an Ada binding for X11R5, but I haven't found one for X11R6. I don't have a problem with importing the C libs into my Ada program, but I was curious if there exists an Ada95 binding to X11R6.

Or if there isn't a binding, if anyone would be interested in such a thing?

Frankly, I don't find Xlib very difficult to use at all. However, it would be much simpler for me if there was an Ada95 binding to Xlib.

Nonetheless I wouldn't go so far as creating an entirely new widget library. (There are tons of those already. Qt, GTK/GTKAda, Athena, Motif, etc...) However I might consider creating a few widgets at a somewhat more abstract level than just as a binding to Xlib so as to take advantage of some of Ada's more useful features. (GLADE distributed objects, Protected Objects, the Safety Annex for those who might have a use for such a thing.)

If I was just creating it for myself, I wouldn't care. But since I plan on making it available to others (if it turns out to be of decent quality) I'm wondering what things I should keep in mind when designing the binding.

Of course I'm assuming that there's a need for such a thing. (I'm not aware of any bindings to X11R6). [...] Any advice from the more experienced among us?

From: Vadim Godunko
<vgodunko@vipmail.ru>

Date: Fri, 13 Sep 2002 08:27:38 +0400
Subject: Re: Xlib and X11R6 bindings
Newsgroups: comp.lang.ada

See <http://home.arcor.de/hfvogt/programming.html>

This is a X11R5/R6 and Motif/OpenMotif Ada bindings. Also is interesting to add nonexistent types/functions to this bindings.

[See also "Motif Binding" in AUJ 23-2 (Jun 2002), p.74, and "X11/Motif Bindings for Gnat 3.12p" in AUJ 21-1 (Apr 2000), pp.18-19. -- dc]

Sockets and Streams

From: Pascal Obry <p.obry@wanadoo.fr>
Date: 15 Oct 2002 18:35:49 +0200
Subject: Re: Starting with socket and stream
Newsgroups: comp.lang.ada

> I'd like to study more deeply the socket implementation in Ada and build some examples using the streams, anyone know some good source for starting with? thanx in advance

Look at AWS sources: <http://libre.act-europe.fr>, this is certainly not the simplest code to start with though...

From: tmoran@acm.org
Date: Tue, 15 Oct 2002 17:46:15 GMT
Subject: Re: Starting with socket and stream
Newsgroups: comp.lang.ada

There is no "the" socket implementation in Ada, there are several. I presume you've looked for "socket" on www.adapower.com. There's also Claw.sockets, whose implementation you can see by downloading the Claw source, and a search for "web" on adapower leads to my Finder (a link tracer) and Smplsrvr (a simple web server).

Streams are a rather general and powerful tool, which can be used in many ways, but with significant work.

From: Samuel Tardieu <sam@rfc1149.net>
Date: Tue, 15 Oct 2002 21:18:59 +0200
Subject: Re: Starting with socket and stream
Newsgroups: comp.lang.ada

You can have a look at AdaSockets (<http://www.rfc1149.net/devel/adsockets>) which lets you use sockets from Ada in raw, line-oriented and stream-oriented mode.

From: Marc A. Criley
<mcq95@earthlink.net>
Date: Wed, 16 Oct 2002 11:01:08 GMT
Organization: Quadrus Corporation
Subject: Re: Starting with socket and stream
Newsgroups: comp.lang.ada

See my article "A Socket Manifestation of Streams" (www.acm.org/sigada/ada_letters/june2001/socket_streams.pdf)

Marc A. Criley, Quadrus Corporation, www.quadruscorp.com

From: Bernd.Specht@gmx.com (Bernd Specht)
Date: Thu, 24 Oct 2002 15:07:12 +0000 (UTC)
Subject: Re: Udp server instructions with adasocket?
Newsgroups: comp.lang.ada

[...], maybe you can find something helpful in some other socket-packages (I personally use (windows only): <http://alfred.hilscher.bei.t-online.de/sophi.zip>)

APQ - PostgreSQL Ada95 Binding

From: Warren W. Gay VE3WWG
<ve3wwg@cogeco.ca>
Date: Sat, 21 Sep 2002 21:41:26 -0400
Subject: PostgreSQL APQ 1.91 Thick Binding now has Win32 Support
Newsgroups: comp.lang.ada

[See also same topic in AUJ 23-3 (Sep 2002), pp.134-135. -- dc]

APQ is a thick binding for Ada95 (GNAT) to the PostgreSQL database.

The most recent changes include: Win32 clients are supported; additional API functionality; bug fixes; binary Win32 release is available.

See <http://home.cogeco.ca/~ve3wwg/software.html> for details.

From: Warren W. Gay VE3WWG
<ve3wwg@cogeco.ca>
Date: Sun, 29 Sep 2002 16:04:37 -0400
Subject: APQ: undefined PQescapeString Work-Around for older PostgreSQL releases
Newsgroups: comp.lang.ada

For some of you that are using older releases of PostgreSQL prior to 7.2.1 and wanting to use APQ, a work-around has been posted to the website that leaves your PostgreSQL software unchanged.

The problem is fixed by using the extracted new C source code for PostgreSQL's PQescapeString function, and adding it to your built libapq.a library (the source for this C module is also provided for your convenience). This solves the APQ linking problem while leaving your existing PostgreSQL software unaltered.

It is unknown how far back in PostgreSQL releases this works, but it should work for any relatively recent vintage.

The work-around is provided at: <http://home.cogeco.ca/~ve3wwg/pqesc.html>

APQ is a thick binding to the PostgreSQL database client library libpq, for Ada95 programs. APQ supports FreeBSD/Linux/UNIX and Win32 client access to PostgreSQL databases. [...]

GNADE - GNAT Ada 95 Database Development Environment

From: Michael Erdmann
<Michael.Erdmann@snafu.de>
Date: Sat, 28 Sep 2002 18:07:54 +0200
Subject: Release of GNat Database Development Environment 1.3.7d
Newsgroups: comp.lang.ada

[See also same topic in AUJ 23-3 (Sep 2002), p.134. -- dc]

GNU Ada Database Environment (GNADE) Version 1.3.7d

This release of the GNADE project features: Thin Ada 95 bindings to the ODBC interface; Embedded ISO/92 SQL Translator generating code on top of the ODBC bindings, the translator supports dynamic and static SQL code; Native bindings to Postgres, MySQL and Oracle; Installation support for MySQL, Postgres and Mimer Database products; Documentation in html, postscript and PDF format.

For more information on the project please refer to:

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

The source code distribution for the platforms Windows, Linux and Solaris is available via:

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

From: Michael Erdmann
<michael.erdmann@snafu.de>
Date: Fri, 18 Oct 2002 06:52:47 +0200
Subject: Release of GNat Database Environment 1.4.0
Newsgroups: comp.lang.ada

The GNADE project (gnade.sourceforge.net) provides tools and packages for the integration of data base products into Ada 95 applications.

This release contains bug fixes, and the following features: ODBC interface; Embedded SQL preprocessor for Ada 95; Oracle Call Interface; Bindings for MySQL; Bindings for PostgreSQL.

The packages are available via: [see above -- dc]

Lex and Yacc for Ada

From: JMG
<jeanmichel.guerville@freesbee.fr>
Date: Tue, 8 Oct 2002 15:48:35 +0200
Subject: Lex et yacc pour Ada
Newsgroups: fr.comp.lang.ada

[Extracts translated from French. See also "Aflex and Ayacc Distribution" in AUJ 21-1 (Apr 2000), p.19. -- dc]

Where can one find a lex and yacc (or equivalent) for Ada?

The goal is to realize a static code analyzer [...]

Jean-Michel Guerville, quality engineer

From: Gautier
<gautier_niouzes@hotmail.com>
Date: Tue, 08 Oct 2002 18:29:21 +0000
Subject: Re: Lex et yacc pour Ada
Newsgroups: fr.comp.lang.ada

This might help you: there's an ayacc and aflex in "newp2ada.zip" on the page <http://www.mysunrise.ch/users/gdm/gsoft.htm>

From: Pascal Obry <p.obry@wanadoo.fr>
Date: 08 Oct 2002 22:45:28 +0200
Subject: Re: Lex et yacc pour Ada
Newsgroups: fr.comp.lang.ada

I have a port of aflex/ayacc for GNAT on my web page. [...]

<http://perso.wanadoo.fr/pascal.obry>

Compiler Construction Tools for Ada

From: Piotr Zgorecki
<pioter@terramail.pl>
Date: Mon, 21 Oct 2002 10:11:53 +0000 (UTC)
Subject: Compiler construction tools for Ada
Newsgroups: comp.lang.ada

Does anybody know of any compiler construction tools for Ada? I found only an outdated port of yacc/lex. I'm interested especially in parser/tree parser generators.

From: Colin Paul Gloster
<Colin_Paul_Gloster@acm.org>
Date: Mon, 21 Oct 2002 12:12:15 +0100
Organization: Dublin City University (DCU)
Subject: Re: Compiler construction tools for Ada
Newsgroups: comp.lang.ada

Lex:
<http://www.adaic.org/standards/95lrm/lexer9x.1>

YACC:
<http://www.adaic.org/standards/95lrm/grammar9x.y>

Java Compiler Compiler:
<http://www.cobase.cs.ucla.edu/pub/javacc/ada9x.jj>
<http://www.webgain.com/download/javacc/details.html/>
<http://www.sun.com/software/communitysource/j2ee/readme.html>

NewJacc / Purdue JavaTree Builder / JavaCC:
<http://www.csee.wvu.edu/~callahan/software.html>

If you want to implement it in Ada, see Tucker Taft's fairly recent efforts in teaching compiler implementation in Ada.

From: Georg Bauhaus <sb463ba@11-hrz.uni-duisburg.de>
Date: Mon, 21 Oct 2002 14:10:14 +0000 (UTC)
Subject: Re: Compiler construction tools for Ada
Newsgroups: comp.lang.ada

See also Cocktail, which has some support for Ada.

<http://www.first.gmd.de/cocktail/>

A search for Cocktail in comp.lang.ada on google will reveal more.

[See also "Parser Generators - Cocktail Toolbox" in AUJ 20-2 (Jul 1999), p.108. - - dc]

From: Michal Nowak <vinnie@inetia.pl>
Date: Mon, 21 Oct 2002 22:37:50 +0200
Subject: Re: Compiler construction tools for Ada
Newsgroups: comp.lang.ada

Here are some useful links from what I have on CD:

Ada and Java Lexer:

<http://home.t-online.de/home/Christ-Utsch.Grein/Ada/Lexer.html>

AFlex & AYacc for GNAT (may be really outdated):

<http://www-users.cs.york.ac.uk/~bernat/resources.html>

AFlexx & AYacc (other implementation):

<http://www.gnuada.org/rpms313p.html>

<http://mast.unican.es/aflex-ayacc-gnat.tar.gz>

AdaGOOP - Ada Generator of Object-Oriented Parsers

<http://wuarchive.wustl.edu/languages/ada/usafa/adagoop/>

<ftp://ftp.usafa.af.mil/pub/dfcs/carlisle/usafa/adagoop/index.html>

[See also "AdaGOOP - Ada Generator of Object-Oriented Parsers" and "More Parser Generators for Ada" in AUJ 23-1 (Mar 2002), p.15. -- dc]

From: Stephen Leake
<stephen.a.leake.1@gsfc.nasa.gov>
Date: 22 Oct 2002 17:51:50 -0400
Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)
Subject: Re: Compiler construction tools for Ada
Newsgroups: comp.lang.ada

For a run-time generated parser, see OpenToken:

<http://www.telepath.com/~dennison/Ted/OpenToken/OpenToken.html>

I like it :).

[See also "OpenToken 3.0b - Token Analysis and Parsing Package" in AUJ 21-3 (Oct 2000), pp.165-166. -- dc]

From: David Marceau
<davidmarceau@sympatico.ca>
Date: Sat, 02 Nov 2002 20:21:21 -0500
Subject: Re: Compiler construction tools for Ada
Newsgroups: comp.lang.ada

Here are a couple of very cool pieces of compiler construction code I discovered in the CLA forum:

1) Check out the sources for ada2nqc translator for programming Lego Mindstorm robots :)

<http://www.usafa.af.mil/dfcs/adamindstorms1.htm>

<ftp://ftp.usafa.af.mil/pub/dfcs/fagin/AdaMindstorsSource.zip>

[See also "Lego Mindstorms" in AUJ 23-3 (Sep 2002), pp.127-128. -- dc]

Sure it's cool what the result does but what's even cooler is the technology for building the ada2nqc :) As you requested ada2nqc uses compiler construction tools for Ada ... specifically AdaGOOP, scayacc and scaflex.

FYI AdaGOOP/scayacc/scaflex may be found here:

<ftp://ftp.usafa.af.mil/pub/dfcs/carlisle/usafa/adagop/index.html>

http://unicoi.kennesaw.edu/ase/ase02_02/tools/sca_tcdsk/scatchb1u.zip

2) Check out the sources for adasubst/adadep

<http://www.adalog.fr/compo2.htm#Semtools>

[See also "Adasubst & Adadep - New Free Semantic Tools for Ada" in AUJ 23-3 (Sep 2002), p.126. -- dc]

As you requested adasubst/adadep uses a compiler construction tool for Ada: ASIS, <http://www.acm.org/sigs/sigada/WG/asiswg/>

If I understand correctly, ASIS is being recommended as the way of doing these things from now on if you want to massage Ada code. [...]

From: Ira Baxter

ibaxter@semdesigns.com

Date: Sun, 3 Nov 2002 15:36:10 -0600

Subject: Re: Compiler construction tools for Ada

Newsgroups: comp.lang.ada

If you mean, "that are for (processing) Ada", see the DMS Software Reengineering Toolkit, with Ada83 and Ada95 front end parsers, tree builders, and prettyprinters. You can also write source-to-source transformations, and/or run attribute evaluations to carry out analyses. See

<http://www.semdesigns.com/Products/DMS/DMSToolkit.html>.

Ira D. Baxter, Ph.D., CTO Semantic Designs, Inc.

[See also "Semantic Designs – Custom Translation Tools" further in this issue. -- dc]

Tools for Graphical Design of Real-Time Systems in Ada

From: paniaguaivan@hotmail.com (Ivan Paniagua)

Date: 21 Aug 2002 05:31:16 -0700

Subject: Application for graphical design of Real-time Systems in Ada

Newsgroups: comp.lang.ada

I'm developing an application for graphical design of Real-time Systems that generate Ada code.

I would like to know if there is a similar application. Can anybody help me? If

there is somebody interested in my work I will be happy to talk to.

[And from another message: -- dc]

[...] I will explain what my application does. It's a graphical interface to design real time systems in a top-down model based in ADARTS model. You must to specify the devices your system has to interact to in a first phase and then you define your system in detail in four more phases. In the last phase you can generate an Ada skeleton. I could send you some screens.

This application is our (we are two students) thesis necessary to end our university education. We want to compare this project with other ones to evaluate it.

From: Michal Nowak <vinnie@inetia.pl>

Date: Thu, 22 Aug 2002 00:03:31 +0200

Subject: Re: Application for graphical design of Real-time Systems in Ada

Newsgroups: comp.lang.ada

I'm aware about one, STOOD. Some information may be found at:

<http://www.tni.fr/tni/offre/stood/index.eng.html>

I saw it at one of my study friends, when we were doing our real-time classes project. I have not seen the generated code, but the diagrams looked quite nice. [...]

There is another solution. There exists a nice drawing tool called Dia (free, available under Linux and Widows, some search on google should bring you the homepage). I think it should be possible to extend it with new stencils. As I saw, it saves the output in XML, so using some processing it should be possible to generate Ada code from it.

[See also "Dia & dia2code - Open Source Ada UML Tools" in AUJ 23-3 (Sep 2002), p.138. -- dc]

From: "Dr. Michael Paus" <paus@ib-paus.com>

Date: Thu, 22 Aug 2002 11:26:00 +0200

Subject: Re: Application for graphical design of Real-time Systems in Ada

Newsgroups: comp.lang.ada

Have a look at this.

http://www.ilogix.com/products/rhapsody/rhap_in_ada.cfm

[See also "I-Logix Launches Rhapsody in Ada" in AUJ 23-2 (Jun 2002), pp.75-76. -- dc]

From: jim hopper

hopperj@macconnect.com

Date: Thu, 22 Aug 2002 21:19:25 -0400

Subject: Re: Application for graphical design of Real-time Systems in Ada

Newsgroups: comp.lang.ada

Check out <http://www.eng.auburn.edu/grasp/>

Real-Time Kernels

From: Per Jakobsen <pdj@dontwork.dk>

Date: Thu, 3 Oct 2002 15:43:43 +0200

Subject: Re: RTL for GNAT?

Newsgroups: comp.lang.ada

> [...] research projects that are writing GNAT-specific kernels, that provide the minimum needed for full Ada. See recent issues of Ada Letters at http://www.acm.org/sigada/ada_letters/ [...]

After a search-session I have found the following environments (or actually minimal OS's):

- RTEMS <http://www.rtems.com/>

- OpenRavenscar <http://www.openravenscar.org/>

- MaRTE <http://marte.unican.es/>

- BOSS

<http://www.first.gmd.de/~sergio/public/ada2002-v4.pdf>

- ORK <http://polaris.dit.upm.es/~ork/>

Many of these seem to refer to something called the Ravenscar Profile:

<ftp://ftp.openravenscar.org/openravenscar/ravenscar00.pdf> [...]

Cheddar - Real-Time Scheduling Simulator

From: Frank Singhoff <singhoff@beru.univ-brest.fr>

Date: 18 Oct 2002 16:34:02 GMT

Organization: Universite de Bretagne Occidentale

Subject: ANNOUNCE: Cheddar 1.2p0

Newsgroups: comp.lang.ada

The LIM/EA2215 team is pleased to announce a new release of Cheddar, a free real time scheduling simulator. Cheddar is a scheduling framework designed for educational purpose and for quick prototyping of real time scheduling algorithms. Cheddar is composed of two independent parts: a graphical editor used to describe a real time application/system, and a library which includes classical real time scheduling/feasibility algorithms/tests.

[See also same topic in AUJ 23-3 (Sep 2002), p.138. -- dc]

The current release is now 1-2p0. Cheddar is distributed under the GNU GPL license. Source code, binaries and documentation can be downloaded from: <http://beru.univ-brest.fr/~singhoff/cheddar>

Cheddar is written in Ada with GtkAda for the graphical editor part.

Cheddar runs on Solaris, Linux and win32 boxes and should run on every GNAT/GTKAda supported platforms (see ACT web site for details).

With Cheddar, you can:

- Draw scheduling with classical real time schedulers (Rate Monotonic, Deadline Monotonic, schedulers, Least Laxity First and Earliest Deadline First, POSIX.4 queuing policies).

- Check scheduling feasibility with: Response times (with DM and RM), Processor utilization (with EDF, LLF, RM and DM), Scheduling for a given base period.

- Support for aperiodic and periodic tasks (with jitter and deadline > period).

The new features are:

- A small framework is provided to design/test multiprocessor schedulers. In the current release, a scheduler with precedencies is built with this framework.

- SCHED_FIFO / SCHED_RR / SCHED_OTHERS Posix 4 scheduler.

- Shared resources support (scheduling and blocking time analysis). Supported protocols: PIP, PCP.

- A small user's guide is now provided.

- Tools to express and do simulations with task precedencies.

- Tindell end to end response time computation.

- Chetto and Blazewicz task parameter modification algorithms to take care of task precedencies.

- Tools to compute bounds on buffers shared by periodic tasks.

- Many fixed bugs from the previous release.

Work in progress:

- Write a user's guide of the framework.

- Response time for EDF and LLF schedulers.

- SRP support.

- Message scheduling.

Feel free to contact us for help or bugs report.

Best Regards,

The LIM1/EA2215 team

MAST - Modeling and Analysis Suite for Real-Time Applications

From: Michael Gonzalez
<mgh@unican.es>

Date: Thu, 31 Oct 2002 11:29:39 +0100
Organization: Universidad de Cantabria
Subject: MAST 1.2

MAST 1.2 is now available from:
<http://mast.unican.es/>

MAST is an open source set of tools that enables modeling real-time applications and performing timing analysis of those applications. The MAST model can be used in a UML design environment to design real-time applications, representing all the real-time behavior and requirements together with the design information, and allowing an automatic schedulability analysis.

[See also same topic in AUJ 22-1 (Mar 2001), pp.10-11. -- dc]

The new version introduces support for analysis of linear systems with offsets, delays, or rate divisors.

This version adds the following tools: Classic_RM_Analysis, Varying_Priorities_Analysis.

This version introduces a new format for the results, and a GUI tool to view those results (gmastresults) [...]

Michael Gonzalez Harbour, Dpto. de Electronica y Computadores, Universidad de Cantabria, Avda. de los Castros s/n, E-39005 Santander, Spain, Phone +34-942-201483, Fax +34-942-201402

The AdaOS Project

From: nickroberts@blueyonder.co.uk (Nick Roberts)

Date: Thu, 29 Aug 2002 22:24:16 GMT
Subject: Re: Ada-inspired OS/Language
Newsgroups: comp.lang.ada

[...] By the way, there is a simple overview of AdaOS now available at <http://www.adaos.net> with a pretty diagram if anybody is interested.

From: nickroberts@blueyonder.co.uk (Nick Roberts)

Date: Sat, 31 Aug 2002 03:38:51 GMT
Subject: Re: Ada OS : towards an Ada
Micro-kernel

Newsgroups: comp.lang.ada

[...] I am about to publish my own design (called 'Bachar'), which solves the various problems that need to be solved (including a few that neither L4 nor anyone else that I'm aware of have even addressed). More info soon.

From: nickroberts@blueyonder.co.uk (Nick Roberts)

Date: Sat, 31 Aug 2002 03:38:51 GMT
Subject: Re: Ada-inspired OS/Language
Newsgroups: comp.lang.ada

> It looks interesting. Does this mean that development on AdaOS is ramping up?

I'm not sure about "ramping up"! Maybe stepping up a gear. I can only hope that as people see the vast technical superiority (and I'm not being ironic, for once) of AdaOS, they will be attracted. But that will surely be a slow process for the next few years.

> Here's one thing I think is an excellent hook: The fact that you are aiming to make it a distributed system. ... That's something kind of unique.

Yes, I think it is. I am not very familiar with Andrew, but my impression is that in fact no-one has produced a truly distributed OS yet (that can shuffle processes between workstations arbitrarily, without any major restrictions).

> Perhaps the more it aims at distribution and load balancing the more of a niche it will find in industry. If it makes

scaling up a system easy by just plugging in more inexpensive computers, it could really catch on.

I believe AdaOS really will make it easy to just add on computers. But this is not the only potential hook. It seems AdaOS may be the first general OS to offer proper security facilities (TCSEC B1 or B2). Do any others offer full dual labeling?

> Especially to the extent that it can exploit Ada tasking as a distribution mechanism ...

I'm glad you have seen that point. I think it is a very significant one. There is currently a hot market in Internet servers, for e-business and many other purposes. These still all work on software that is very primitive (e.g. vulnerable to buffer overrun vulnerabilities), and I see this as one potential ouvre for AdaOS. [...]

From: nickroberts@blueyonder.co.uk (Nick Roberts)

Date: Thu, 05 Sep 2002 03:16:21 GMT
Subject: Re: Ada-inspired OS/Language
Newsgroups: comp.lang.ada

[On the suggestion to do a web search for Amoeba by Tanenbaum, which the poster claimed also allowed process "shuffling": --dc]

I'm sure I tried this (and got relatively little information), but I will try again! ... I tried and got loads of stuff. Great! Much reading matter.

[Georg Bauhaus <sb463ba@11-hrz.uni-duisburg.de> later provided the following URL: -- dc]

<http://www.cs.vu.nl/~ast/cs/dist.html>

From: Rod Haper
<rhaper@houston.rr.com>

Date: Thu, 12 Sep 2002 02:17:10 GMT
Organization: Road Runner - Texas
Subject: Re: Ada-inspired OS/Language
Newsgroups: comp.lang.ada

If you haven't already, you might also want to have a look at Inferno for ideas about a distributed OS architecture, applications, security, programming language support, etc.

<http://www.vitanuova.com/inferno/index.html>
[and also:] [concepts.html](http://www.vitanuova.com/concepts.html) [papers.html](http://www.vitanuova.com/papers.html) [papers/bltj.html](http://www.vitanuova.com/papers/bltj.html)

From: Richard Riehle
<richard@adaworks.com>

Date: Fri, 06 Sep 2002 10:31:26 -0700
Organization: AdaWorks Software
Engineering
Subject: Re: Ada-inspired OS/Language
Newsgroups: comp.lang.ada

The architecture [of AdaOS at <http://www.adaos.net> -- dc] looks pretty good. Hope you can bring it to completion.

One comment. I am concerned about calling it AdaOS. At present, anything with the name Ada in it is likely to be

ignored by the larger audience you would hope to gain.

The published product would do well to stand on its own, support languages besides Ada, and have a name that makes Ada less of an explicit issue. In time, people concerned with the technology will realize that it is written in Ada, while those who simply want to get some work done will use it without any knowledge of its underlying implementation.

Meanwhile, if this project gets to the point where we can see it becoming a reality, some of us ought to get busy writing applications for it. The only way it will become popular is if real people use it. Those real people will need spreadsheets, word processors, etc.

From: nickroberts@blueyonder.co.uk (Nick Roberts)

Date: Sat, 07 Sep 2002 00:23:18 GMT

Subject: Re: Ada-inspired OS/Language Newsgroups: comp.lang.ada

[On the name AdaOS: -- dc]

I rather agree with you. AdaOS was originally named 'Alguinntar', which was perhaps not the most inspiring name! Shortly after the project was formed (in 1999), the members (then about 15) had a vote on the name, and the overwhelming majority voted for 'AdaOS'.

In fact, there are some advantages. Occasionally someone will ask "Is that AdaOS as in the Ada language I've heard of somewhere?," and this naturally gives me the perfect *oeuvre* for my (now polished ☺ 'sales' spiel about Ada.

In fact, nearer the time, what I might do is to invent various 'marketing' names for different releases of AdaOS. How many names does Linux have now?

I intend to implement a judicious selection of other languages on AdaOS. For example, SQL3 or OQL for the database engine, and a C compiler. Maybe a JVM.

[On the need to write applications for AdaOS: -- dc]

Absolutely. AdaOS will be an object-oriented OS. This will provide a framework for componentware, which I am sure is the key to getting sophisticated application software out there without going through the classic heavy programming cycle first.

I am not certain, but if AdaOS is based on CORBA 3, the opportunity will be there for the direct import of COM components. Now we all know how reliable typical COM components tend to be ☺ but this could be handy.

From: nickroberts@blueyonder.co.uk (Nick Roberts)

Date: Sat, 07 Sep 2002 15:18:31 GMT

Subject: Re: Ada-inspired OS/Language Newsgroups: comp.lang.ada

of module (English and French), sources of the software (English), the complete

> [...], far more important than the name is that there be some usable, working piece of it in existence.

Yes I agree with this idea too. I have started a mini-project on SourceForge just for the microkernel, called "Bachar". I shall try to get this component done ASAP, and it will stand alone and could be usable by a lot of people for a variety of projects.

I am trying to square the circle of releasing a usable item as early as possible, but at the same time ensuring that it will be of the highest quality.

AdaBrowse 3.0 with XML support

From: Thomas Wolf

<t_wolf@angelfire.com>

Date: Fri, 30 Aug 2002 14:14:09 +0200

Subject: ANN: AdaBrowse 3.0 with XML support

Newsgroups: comp.lang.ada

I've published version 3.0 of AdaBrowse.

It's available at the URL

http://home.tiscalinet.ch/t_wolf/tw/ada95/adabrowse/ in source form and as a pre-built executable for Win NT/2k and GNAT 3.14p. AdaBrowse is distributed under the terms of the GPL.

AdaBrowse is known to work on Win NT/2k and Linux. It should work on other Unices, too.

(For those who don't know it yet:

AdaBrowse is an HTML/XML documentation generator for Ada 95 library unit specifications.)

[See also "AdaBrowse - A Javadoc for Ada 95" in AUJ 23-3 (Sep 2002), pp.136-137. -- dc]

Changes from the previous version (2.13):

1. AdaBrowse 3.0 now can also generate XML output. The XML file contains all the structure, syntax-coloring, and cross-reference information present in the HTML, but is far better suited for further processing. A DTD for the generated XML is included in the distribution.

This makes it possible for anyone to use any odd off-the-shelf XML parser toolkit (such as XML/Ada) to parse the XML files generated by AdaBrowse and produce whatever format is desired from it -- docbook, or texinfo, or anything else. Maybe even differently formatted HTML, if you happen not to like the HTML generated by AdaBrowse.

This feature saves you the trouble of having to write your own Ada 95 parser to collect all the information; and it saves me the trouble of providing built-in formatters for any odd format out there.

2. Some improvements concerning cross-references to implicitly inherited

documentation of the project (only in

subprograms and enumeration literals, and cross-references to items from instantiations of nested generics.

3. The HTML generator now uses both the line and the column number to produce cross-references. Former versions of AdaBrowse used only the line number.

This means that HTML files generated by AdaBrowse 3.0 are not compatible with files generated by earlier versions. If compatibility is needed (say, because for some reason you cannot re-generate the documentation with V3.0), use the "-l" command line option, which makes V3.0 use only the line number, too.

The XML generated by AdaBrowse always contains both the line and the column number.

AdaBrowse 3.0 has passed all my regression tests; the new XML generation has been tested by verifying the generated XML files to comply to the XML 1.0 DTD included in the distribution using Xerces 2.0.0.

For information on how to submit bug reports or enhancement proposals, see the User's Guide included in the distribution.

AdaDoc - Documentation tool for Ada Package Specifications

From: Julien Burdy <dev@newdeal.ch>

Date: Sat, 31 Aug 2002 17:12:09 +0200

Subject: ANN: AdaDoc 2.0

Newsgroups: comp.lang.ada

We are announcing the release of AdaDoc version 2.0. It had been completely rewritten to be more extensible and remove the limitations of AdaDoc v1.x.

[See also "AdaDoc - Html Generator for Ada Package Specifications" in AUJ 22-4 (Dec 2001), pp.200-201. -- dc]

AdaDoc is a tool for developers using the Ada95 programming language.

Its goal is to create documentation in different formats from a package specification. [...]

The program analyzes the specification (by controlling its syntax), then writes a document with the desired format. The format of exit depends on the selected module. AdaDoc writes an XML file (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 (French), a mailing list to be informed of the new release.

AdaDoc is under the GPL.

The Ultimate SI Units Cracker

From: Grein, Christoph
<christoph.grein@eurocopter.com>
Date: Thu, 12 Sep 2002 10:11:43 +0200
(MET DST)
Subject: Announcement: The Ultimate SI Units Cracker
Newsgroups: comp.lang.ada

The ultimate SI Units cracker is there!

There has always been a demand to be able to compute with physical items where correctness of dimensions is checked, but to me, all previous attempts seemed unsatisfactory (and there were many) – now it's there, in full generality, including arbitrary powers and roots.

Yes, it's run-time consuming since the dimension is an attribute. No, it's not precluded from being used for hard real-time systems. Yes, it can be applied under hard real-time conditions if it's done correctly.

Yes, it's easy to switch off dimensions - only pure numerics remains.

See <http://home.T-Online.de/home/Christ-Usch.Grein/Ada/SI.html>

It has been released under GMGPL. Enjoy...

Christoph Grein, Member of Ada Germany, <http://www.ada-deutschland.de>, Christ-Usch.Grein@T-Online.de

From: Jeffrey Creem
<jeff@thecreems.com>
Date: Thu, 12 Sep 2002 10:50:55 GMT
Subject: Re: Announcement: The Ultimate SI Units Cracker
Newsgroups: comp.lang.ada

Sounds like it might be interesting to look at.

[...] note that the page seems to make the somewhat incorrect assumption that hard real-time implies "fast". Hard real-time just means you can never miss a deadline (where soft real-time means that you can sometimes miss some deadlines but not often). While these terms vary from author to author, it has been my experience that these are pretty close to the generally accepted definitions.

I believe you really should be saying that there might be a concern about your approach (which you believe you have solved) for "time critical" applications. Or, one could say that your approach is still ok in applications in which computational efficiency is important.

My offline MPEG encoding I do at home is by no means hard real-time (it is a hobby, perhaps if I was getting paid to do this it would be soft real-time :) But it is time critical because it is computationally

expensive and slow (with respect to how long humans want to wait for the data).

From: Grein, Christoph
<christoph.grein@eurocopter.com>
Date: Mon, 14 Oct 2002 13:12:17 +0200
(MET DST)
Subject: Update: The Ultimate SI Units Checker
Newsgroups: comp.lang.ada

The ultimate SI Units checker deals with physical items in full generality, including arbitrary powers and roots, checking the correctness of dimensions in all expressions:

```
Current_Density := (4.0/9.0) * Eps0 *
Sqrt (2.0 * Electron_Charge /
Electron_Mass) *
Voltage**(3/2) / Distance**2;
```

[...] Temperature scales like Celsius and Fahrenheit have been added (the corresponding SI unit is Kelvin) with conversion functions taking into account whether temperatures or temperature differences are converted. (For differences, e.g. 1K = 1C, but not for temperatures themselves.) [...]

Handling Physical Dimensions in Ada

From: Grein, Christoph
<christoph.grein@eurocopter.com>
Date: Wed, 23 Oct 2002 08:11:43 +0200
(MET DST)
Subject: Poll: Units of Measure (Physical Dimensions)
Newsgroups: comp.lang.ada

Some time ago, I released an SI units checker, a set of Ada packages that can handle physical dimensions in full generality.

Now I know there are many such methods around, so I think it valuable to have one place where they all are enumerated so that anyone interested can choose the optimal method for their problem at hand.

So I would like to ask anyone who has built such a method to send me in a short characterization. To give you an idea what kind of information I expect, I've set up a template at

<http://home.T-Online.de/home/Christ-Usch.Grein/Ada/Dimension.html>

I promise you that I'll put your text unchanged into this page. And before I'm going to release it publicly (I intend to ask David Botton to put it into AdaPower), you'll be given a note that all input is ready. Then you can check again if you'd like to add or change something. I'm only going to release the page after I've got the go from all contributors.

Please send your input to: Christ-Usch.Grein@T-Online.de

I'm looking forward to your valuable input.

Auto_Text_IO & SAL

From: Stephen Leake
<stephen.a.leake.1@gsfc.nasa.gov>
Date: 30 Sep 2002 10:29:03 -0400
Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)
Subject: Auto_Text_IO & SAL release
Newsgroups: comp.lang.ada

Due to user interest, I've posted a new release of Auto_Text_IO and SAL on my web site: <http://users.erols.com/leakstan/Stephe/>

[Auto_Text_IO generates Text_IO routines Put and Get, using named notation, when given an Ada package containing types. See also "Auto_Text_IO ASIS Application" in AUJ 22-4 (Dec 2001), pp.206-207, and "SAL - Stephe's Ada Library" in AUJ 23-1 (Mar 2002), p.9. -- dc]

Ada-related Products

Aonix - ObjectAda for Windows 7.2.2 Patch Update

From: owner-intel-objectada@sf.aonix.com
Date: Tue, 29 Oct 2002 18:58:32 -0800
Subject: Intel-OA: New ObjectAda update
To: intel-objectada@sf.aonix.com

A new patch update to ObjectAda for Windows 7.2.2 (1102) is now available. The update download file and the Release Notes are available at

<http://www.aonix.com/content/support/ada/patches/objectada.html>

Please see the Release Notes for more information.

Aonix Ada Support, adasupport@aonix.com, 1-800-972-6649

DDC-I - JTAG Debugging for SCORE PowerPC

URL:
http://www.ddci.com/news_vol3num8.shtml
Subject: DDC-I Online News, September 2002, Vol. 3 Issue 8

JTAG Debugging for SCORE PowerPC, By Karl Rehmer

JTAG (Joint Test Action Group pronounced "jay-tag") is an IEEE specification (IEEE 1149.1). It was originally implemented to allow testing of all the pin connections of a chip and its interconnections to other chips on the circuit board. It is a serial protocol and chips on the board may be daisy-chained together. The chain may be very long. There is no specification stating any inclusion of resources for software debug nor is there a prohibition.

JTAG debugging provides the advantage of having the debugger communicate directly with the target hardware, avoiding the use of a debug monitor on the target.

Because of this, the environment in which your program runs when it is debugged using JTAG debugging is closer to the actual environment in which it will run in production. There is no debug monitor that is in actual control of the target board.

DDC-I's SCORE multi-language debugger for the PowerPC provides support for JTAG debugging, communicating with devices provided by Macraigor Systems. The supported devices for the NT hosted debugger are Wiggler, Raven, and mpDemon. For the Solaris hosted debugger, the mpDemon is supported.

The Macraigor web site (www.macraigor.com) has a link to an interesting article "The Zen of BDM" written by Criag A. Haller of Macraigor Systems. This article describes the historical approach to debugging and how the advantages of BDM (background debug mode) debugging overcome some of the limitations of the traditional methods. JTAG debugging is a kind of background debug mode debugging. This article even provides some helpful information for those designing their own custom boards with a JTAG interface. [...]

DDC-I - Improved Interface SCORE GUI & Third-Party SCORECast Tools

URL:

http://www.ddci.com/news_vol3num8.shtml

Subject: DDC-I Online News, September 2002, Vol. 3 Issue 8

Tech Talk: Improved Interface Between SCORE GUI and Third-party SCORECast Tools, By Lisa Jett

With Release 2.3 of SCORE, a new and improved interface has been implemented between the SCORE GUI and the SCORECast (module testing) tools. When the AdaCast or CCast tools are invoked, SCORE automatically generates a configuration file that the SCORECast tools read during startup that contain the definition of environment variables key for the compile, link, debug, and load steps used by the SCORECast tools to test a particular module.

Now users define communication and build options all within the SCORE Tool Options dialog and that information gets automatically communicated to the SCORECast tools, eliminating the need for specification of the options again in the SCORECast tools. [...]

DDC-I - DACS Network Bridge Added to DACS Programming Environment

URL:

http://www.ddci.com/news_vol3num10.shtml

Subject: DDC-I Online News, November 2002, Vol. 3 Issue 10

DDC-I's DACS Network Bridge Delivers Valuable Developer Connectivity by Enabling Remote Management for Multiple Targets

DDC-I announces the addition of DACS Network Bridge to their proven DACS programming environment for WinNT hosts. Available for DACS-80x86, DDC-I's Ada Compiler Systems for 80x86 and Pentium processors, the Ethernet-based network bridge addresses the increasing need among developers to maximize geographically diverse programming resources while keeping project costs to a minimum.

"Since cost reduction is the driving force in embedded systems design, the ability of the DACS Network Bridge to allow a single programmer to manage multiple targets from a single remote site using existing network resources can deliver significant cost benefits straight to a projects bottom line," explains DDC-I DACS Product Champion Thorkil Rasmussen.

Integrated with the current DACS Windows cross toolset, the host tools communicate via a network to a "bridge" PC, in an engineering lab for example. The debugger and loader connections to the targets are managed by software on the bridge PC which directs the messages over high-speed connections to the appropriate targets. Switching targets requires only a user-generated software command from the debugger.

In addition to empowering remote programmers, such software switchable targeting reduces the risk of debugging hardware connection issues when swapping targets. A configuration file on the bridge PC maps serial ports and communication options and specifies network sockets. Target connections can be reserved for specific users, while simple queries report connection status.

"As projects become larger and budgets grow tighter, the DACS Network Bridge is yet another example of how DDC-I is working to create genuinely beneficial productivity tools that address the real needs of the embedded systems programming community," concludes Rasmussen.

PegaSoft - TextTools Console GUI & TIA Tiny IDE for Ada

From: PegaSoft Canada <adalin-1@tiamet.vaxxine.com>

Date: Mon, 9 Sep 2002 12:00:32 -0400 (EDT)

Subject: TextTools and TIA projects updated
To: GNAT Discussion List
<gnatlist@lyris.seas.gwu.edu>

Slightly updated versions of my TextTools console windowing package and TIA (Tiny IDE for Ada) have been posted at <http://www.pegasoft.ca>

[See also "PegaSoft - TextTools 1.0.5 Console Development Package" in AUJ 23-3 (Sep 2002), p.145, and "PegaSoft - TIA 0.7.3 Tiny IDE for Ada & TextTools 1.0.4 Console GUI" in AUJ 22-3 (Sep 2001), p.146. -- dc]

Ken O. Burtch, PegaSoft, Jordan Station, ON, Canada, <http://www.pegasoft.ca>, ken@pegasoft.ca

PegaSoft - BUSH Business Shell

From: Ken O. Burtch

<kburtch@sympatico.ca>

Date: Wed, 11 Sep 2002 20:39:48 -0400

Subject: ANNOUNCE: BUSH 0.9 release
Newsgroups: comp.lang.ada

BUSH, the PegaSoft Business Shell, is a powerful Linux/Unix shell for designing secure, reliable shell scripts that can be later compiled as a fast executable programs. It can also be used as an interactive login shell, generate Java Virtual Machine applications, and open network sockets. BUSH is a robust and readable alternative to BASH, CSH, and (to a certain extent) Python and Perl.

[See also "BUSH 0.8 - AdaScript Shell" in AUJ 22-4 (Dec 2001), pp.205-206. -- dc]

The business shell now features backquoted commands, filename completion, customizable prompt, 4 new built-in packages (28 function CGI/cookie package, calendar, source_info, lock_files). New script debugging mode (-break) handles the insertion of breakpoints (pragma inspection_point) and debugging statements (pragma debug). SIGINT (Control-C) can break out to a command prompt at any time. Does greatly enhanced and include a hackers guide. Usual bug fixes.

Homepage: <http://www.pegasoft.ca/bush.html>

[And from another message: -- dc]

BUSH is written almost entirely with Ada. There is a small amount of C for portable UNIX signal handling.

From: Ken O. Burtch

<kburtch@sympatico.ca>

Date: Wed, 02 Oct 2002 10:00:28 -0400

Subject: ANNOUNCE: BUSH 0.9.1
Newsgroups: comp.lang.ada

BUSH 0.9.1 has been released at <http://www.pegasoft.ca/bush.html>. It contains some fixes to the command line features.

If you are interested in a Windows port, I've talked with an experienced Ada/Windows developer. He has looked through the sources and knows what needs to be done. He estimates that a conversion would take two weeks full-time so he doesn't want to undertake it by himself.

All he needs is people to help him with the conversion. If you're interested, or know someone who is, let me know. My email address is on the PegaSoft site.

Rational - Apex 4.2.0 for Windows NT

From: Greg Bek <gab@Rational.Com>
Date: Mon, 2 Sep 2002 18:47:48 -0700
Subject: Rational Apex NT, version 4.2.0 is available by FTP
To: Apex Announcements <apex-announcements@Rational.Com>

[In the URLs below, substitute <rel> by ftp://ftp.rational.com/public/apex/releases/win -- dc]

Product: Rational Apex NT

Version: 4.2.0

Platform: Windows/NT

URL: <rel>/apex_nt.4.2.0-self_extract

This release is pending Generally Available (GA) status as it goes through the final steps of the manufacturing process. We anticipate that this will be complete within the next 30 days. Once this release reaches GA status, it will be available for shipping. Until then, it is being provided on this FTP server for immediate access. Follow the URL above for Rational Apex NT self-extracting exe download.

Release Notes: <rel>/apex_nt.4.2.0-self_extract/apex_nt_relnote.html

Rational - Apex 4.2.0b for IBM AIX, Compaq Tru64, HP HP-UX, SGI IRIX

From: Greg Bek <gab@Rational.Com>
Subject: Rational Apex 4.2.0b for IBM AIX, Compaq Tru64, HP HP-UX, SGI IRIX is available by FTP
To: Apex Announcements <apex-announcements@Rational.Com>

[Extracted from postings on Thu, 26 Sep 2002. In the URLs below, substitute <rel> by <apex>/releases, <doc> by <apex>/documents/unix, <apex> by ftp://ftp.rational.com/public/apex -- dc]

Product: Rational Apex

Version: 4.2.0b

Platform: IBM AIX (aix), Compaq Tru64 (axp), HP HP-UX (hp) & SGI IRIX (sg32)

URL: <rel>/<platform>/apex.4.2.0b

This release is pending Generally Available (GA) status [see above -- dc]. Follow this link for Rational Apex download and installation instructions. [...]

<ftp>/standard.msgs/install_instructions.html

Rational Apex 4.2.0b Release Note:

<doc>/release_note.4.2.0b.dir/release_noteTOC.html

<doc>/release_note.4.2.0b.ps.Z

Rational Apex 4.2.0b Install Guide:

<doc>/install_guide.4.2.0b.dir/igTOC.html

<doc>/install_guide.4.2.0b.ps.Z

Rational - Ada Analyzer 4.2.1 for Solaris, HP-UX, Tru64, AIX, IRIX

From: Greg Bek <gab@Rational.Com>
Date: Fri, 11 Oct 2002 08:44:05 +0930
Subject: Ada Analyzer 4.2.1 for Solaris, HP-UX, Tru64, AIX, and IRIX is available by FTP

To: Apex Announcements <apex-announcements@Rational.Com>

[In the URLs below, substitute <rel> by <analyzer>/releases, <doc> by <analyzer>/documents/unix, <analyzer> by ftp://ftp.rational.com/public/ada_analyzer -- dc]

Product: Ada Analyzer.

Version: 4.2.1.

Platform: Sun Solaris (sol), HP HP-UX (hp), IBM AIX (aix), Compaq Tru64 (axp) & SGI IRIX (sg32).

URL: <rel>/<platform>/ada_analyzer.4.2.1

This is the Generally Available (GA) release. Follow this link for Ada Analyzer download and installation instructions. [...]
<ftp>/standard.msgs/install_instructions.html

Ada Analyzer 4.2.1 Release Note:

<doc>/release_note.4.2.1.dir/AA_Release_Note.4.2.1.html

<doc>/release_note.4.2.1.ps.Z

Rational - Apex Embedded for Rational Exec 4.2.0 for Sun Solaris to Intel Architecture Family

From: Glenn, Eddie <cav@Rational.Com>
Date: Mon, 21 Oct 2002 13:26:15 -0700
Subject: Rational Apex Embedded for Rational Exec 4.2.0 for Sun Solaris to Intel Architecture Family is available by FTP

To: 'apex-announcements@rational.com' <apex-announcements@Rational.Com>

[In the URLs below, substitute <rel> by <cross>/releases, <doc> by <cross>/documents/unix, <cross> by ftp://ftp.rational.com/public/apex_cross -- dc]

Product: Rational Apex Embedded for Rational Exec

Version: 4.2.0

Platform: Sun Solaris to Intel Architecture (x86) Family

URL: <rel>/sol/i386/rt/4.2.0

This release is pending Generally Available (GA) status [see above -- dc]. Follow this link for Rational Apex for Rational Exec download and installation instructions. [...]
<ftp>/standard.msgs/install_instructions.html

Rational Apex Exec 4.2.0 Release Note for Intel Architecture:

<doc>/Rexec/relnote.4.2.dir/rexec_relnote_386/rexec_release_noteTOC.html

<doc>/Rexec/relnote.rexec.386.4.2.ps.Z

Semantic Designs - Custom Translation Tools

From: Ira Baxter <idbaxter@semdesigns.com>
Date: Wed, 18 Sep 2002 14:50:17 -0500
Subject: Re: C++ to Ada translator?
Newsgroups: comp.lang.ada

> I'm looking for a C++ to Ada translator. Can anybody help me?

We build custom translation tools as special cases of our generalized compiler technology used for analysis and enhancement of large scale software systems. See <http://www.semdesigns.com/Products/Services/LegacyMigration.html>. We have Ada and C++ modules for our tools, although we have not paired these before.

Another poster suggested that the "nature of the C++ code" would be lost. From a purist black and white point of view, probably true.

From a practical point of view, perhaps not. C++ classes have decent analogs in Ada95, and so the basic translation seems feasible.

Lots of icky details invariably show up. I'd guess type casts in C++ would be gruesome when converted to Ada, and pointer arithmetic truly ugly. Translating macros to something sensible is hard, but there's less macro stuff in C++ code than in C programs. Nasty assumptions about the nature of "strings" (zero terminated in C++) will also show up, probably as a String package coded in Ada to simulate Ada semantics, etc. The resulting code would be ... somewhat readable.

(We've had pretty reasonable results with other language translations, such as JOVIAL to C, where the basic language concepts are quite similar. We do better than you might expect, copying comments, etc).

I'm sure this will start a flame fest over readability. I'm not going to get into that, because the result depends on what a translator actually does. The issue of the OP, if he really needs translation, may be the driving factor. And I suspect the Ada community would much rather see C++ programs moving to Ada than the other way round, even if the answer isn't ideal :- }

And these translators, even with our foundation tools and available front-end language modules, still aren't easy or cheap to produce.

Just a hell of a lot more practical, if you insist on needing a translation, than lots of other approaches.

Ira D. Baxter, Ph.D., CTO Semantic Designs, www.semdesigns.com

From: Ira Baxter

<idbaxter@semdesigns.com>

Date: Thu, 19 Sep 2002 08:40:53 -0500

Subject: Re: C++ to Ada translator?

Newsgroups: comp.lang.ada

> I meant templates and multiple inheritance when I talked about losing the C++ nature in the translated code.

Templates if used in their full generality would be pretty hard to translate.

Templates used simply might translate to various kinds of generics. Multiple inheritance can be faked with delegation. Not pretty, but workable.

The real question for the OP is how much of the zany parts of C++ does he use, in what kind of volume?

If there's only a little bit of the odd stuff, then the "nature" could be preserved probably reasonably.

I'll repeat the other comment: often such translations are politically motivated (after all, there are perfectly good, er, usable, C++ compilers, why switch?) If that's really the case, the output doesn't have to be wonderful. It only has to be not terrible. Yes, the downstream engineers suffer. But that always happens when politics intervenes.

From: Ira Baxter

<idbaxter@semdesigns.com>

Date: Thu, 19 Sep 2002 18:28:54 -0500

Subject: Re: C++ to Ada translator?

Newsgroups: comp.lang.ada

> I sometimes translate Spanish to English and vice versa. I always use one or more auto-translating programs first. The output is totally unacceptable [...] but it gets me to the final result faster than if I did the first pass myself. I expect similar from a C/C++/Java to Ada translator.

Translating natural languages is much harder than translating artificial computer languages. Natural languages contain ambiguity, inconsistency (maybe, maybe not), incredible assumptions about context (this whole newsgroup thread), broken sentences, misspellings, dialects, ... duh. (-:). no wonder they're hard to translate. And if you use that as a baseline, yes, I can see how you'd arrive at your conclusion.

Artificial languages tend not to have most of this junk. Individual constructs in one language can often be simulated by a modest number of constructs from another. A trival "expansive" translation

will then lead to code bloat, but not "wrong answers". We can argue about readable.

Coupling "naive" translation with post-optimization can strip away much of the generality that the "number of constructs" bring, by taking into account constraints imposed by the local code. With that, you can get code that is arguably readable (as opposed to unarguably unreadable).

See the example in the slides found at <http://www.semdesigns.com/Products/Services/LegacyMigration.html>.

This is JOVIAL to C; in fact, if one looks closely, it will be a bit more unreadable for a C purist because the customer asked us to retain the JOVIAL type names as macros for defining variables, to ease the education curve of his existing JOVIAL team. All those macro names are trivially replaceable by their content, e.g. "I" by "int".

I'll agree that the translated code is not as pretty or nice as that written by an experienced engineer. I've seen much worse code from some so-called "professional" engineers.

From: Ira Baxter

<idbaxter@semdesigns.com>

Date: Sat, 21 Sep 2002 10:54:14 -0500

Subject: Re: C++ to Ada translator?

Newsgroups: comp.lang.ada

> My point (and I think you supported it) was that, although the output of either kind of translator is not acceptable for delivery either one can provide a valuable, time-saving head start.

For many translators I've seen, I'll agree with this. I think, however, this has to do with the quality of translator. And I will immodestly say that things are changing for the better; our JOVIAL-to-C customer seems to be happy enough. He has no specific plans to do a major cleanup of the translated code.

There are often major plans to modify translated programs after the fact. This is driven by the need to add functionality. For the JOVIAL case, the original code ran on 16 bit machines, and they had hardly enough room to turn around. Running in a 32 bit C environment, now they plan to add lots of new functionality.

Overview of Ada Test Tools

From: coolpages@excite.com

Date: 10 Sep 2002 12:59:29 -0700

Subject: Ada test tools

Newsgroups: comp.lang.ada

Looking into some test tools like AdaCAST (from VectorCAST) and also LDRA. Unit testing, MC/DC, integration level tests, coverage analysis, auto test driver a plus.

What is the current state of test tools using primarily Ada?

Suggestions?

From: R. Tim Coslet

<R_Tim_Coslet@pacbell.net>

Date: Wed, 11 Sep 2002 04:27:46 GMT

Subject: Re: Ada test tools

Newsgroups: comp.lang.ada

I've used AdaCAST quite a bit. Overall it seems to work very well for CUT level testing. [...]

From: Patricia Langle

<patricia.langle@aonix.fr>

Date: Thu, 12 Sep 2002 18:30:13 +0200

Subject: Re: Ada test tools

Newsgroups: comp.lang.ada

As Vector Software partner, Aonix France has now a good experience of using VectorCAST/Ada on big projects. Our professional services and our customers have successfully tested Ada applications. The last one was for SIL4 certification (highest critical level in ground transportation systems). Our teams appreciate the tools (user friendly) and the efficiency of Vector Software customer support.

Just have a look on their Web site www.vectorcast.com, you could download all the documentation you need.

From: KSW <bitchsession@yahoo.com>

Date: Mon, 30 Sep 2002 23:58:54 GMT

Subject: Re: Ada test tools

Newsgroups: comp.lang.ada

I would be happy to review our unit testing and code coverage product suite with you. Feel free to contact me.

Kyle Wilbur, Vector Software, ksw@vectors.com

From: John Kern <jkern3@visteon.com>

Date: Wed, 11 Sep 2002 11:03:20 -0400

Organization: Visteon Corporation

Subject: Re: Ada test tools

Newsgroups: comp.lang.ada

How about TestRT from Rational Corporation.

<http://www.rational.com/products/testrt/index.jsp>

From: Peter Amey <peter.amey@praxis-cs.co.uk>

Date: Wed, 11 Sep 2002 17:00:18 +0100

Organization: Praxis Critical Systems

Subject: Re: Ada test tools

Newsgroups: comp.lang.ada

AdaTest from IPL

(<http://www.iplbath.com/products/tools/pt600.shtml>)

From: Simon Wright

<simon@pushface.org>

Date: 11 Sep 2002 21:33:14 +0100

Subject: Re: Ada test tools

Newsgroups: comp.lang.ada

We are using AUnit from ACT, it doesn't do all the fancy stuff (MC/DC etc) but it is working very well.

Also gcov for coverage; not always completely straightforward, especially with generics, but still useful.

Ada and Linux

Safety-Critical Ada Software on Linux

URL:

<http://www.linuxjournal.com/article.php?sid=5965>

Subject: *Testing Safety-Critical Software with AdaTEST*

[In AUJ 23-3 (Sep 2002), p.151, Tucker Taft referred to an article in the online Linux Journal. For the full text, see URL above. Some extracts: -- dc]

"Testing Safety-Critical Software with AdaTEST", posted on Friday, July 26, 2002 by Thomas Østerlie and Ståle Dahl. [...]

The increased adoption of embedded Linux within the general consumer electronics market gives rise to new areas of application development for embedded Linux outside the usual realm of PDAs and mobile phones.

Industries such as avionics, railway signaling, process control and medicine are all users of embedded systems. Common to them all is a need for safety-critical software. Safety-critical software is a class of systems whose failure may cause injury or death to human beings. In addition to real-time requirements, including proper control over timing and scheduling, such systems have absolute demands regarding correctness of behavior. [...]

Strict formal methods are applied in developing safety-critical software. Counted among these methods are various forms of testing.

Testing is performed to eliminate possible bugs and to ensure correctness of behavior. The requirements for developing safety-critical systems are so strict that even tools used in the development process must comply with minimum requirements for formal methodology. One such testing tool AdaTEST, from the British company IPL. [...]

However, a pertinent question arises: AdaTEST is designed for testing software written in Ada; with the power of C at hand, why bother with programming Ada for the Linux platform? [...]

Ada and Linux aren't a necessarily obvious combination. As several free and/or commercial real-time Linux implementations already are available on the market, the infrastructure for developing safety-critical Linux systems is in place. Unlike general purpose languages, say C and Java, hard real-time requirements are inherent in the Ada core language's tasking model. The task is an Ada-language construct equivalent to the operating system's thread. Due to its

strong typing, we can be confident that Ada programs contain few surprises -- a perfect match for developing safety-critical software. Ada has therefore become a de facto standard for industries like avionics and railway signaling.

As for embedded platforms, Ada was originally developed by the US Department of Defense for use in embedded system applications. It is therefore a perfect match for the future's embedded, safety-critical Linux solutions.

But how does Ada mix with Linux? In fact, it mixes quite well. The GNU Ada tool chain (GNAT) is an Ada front-end to gcc, tying Ada closely with the operating system. With standard facilities to import C functions, Ada allows for metal-near programming by importing any C functions, including system calls if need be. [...]

Despite its commercial license, AdaTEST comes with out-of-the-box support for GNAT, which makes it interesting for developing Linux software. [...] One could argue that good open-source alternatives for testing already exists (like AUnit for Ada), so why buy something when you can get roughly the same functionality for free? Well, sometimes you can't. If you are required to document full statement coverage and/or full decision coverage (as in DO-178B), you may be out of luck with other programs. [...]

Ada and Microsoft

Ada for .NET Effort in Progress

From: *Craig Carey <research@ijs.co.nz>*
 Date: *Tue, 20 Aug 2002 06:44:09 +1200*
 Subject: *Re: JGNAT 'Ada 95 to bytecode' compiling project ends*
 To: *team-ada@acm.org*

Carlisle Martin C Dr USAFA/DFCS wrote:

> FYI, at SIGAda 2002, I will be presenting a very similar project [...] which compiles Ada to .NET, and is largely based on the JGNAT sources.

8-12 December 2002, Houston, Texas

<http://www.acm.org/sigs/sigada/conf/sigada2002/>

<http://www.microsoft.com/net/> [...]

[Read more in the next AUJ issue about the first release of A#, a port of Ada to the Microsoft .NET Platform. -- dc]

Status of GCC 3 on Windows

From: *Jerry van Dijk <jvandyk@attglobal.net>*
 Date: *16 Oct 2002 17:53:30 +0200*
 Subject: *Re: Why does this happen?*
 Newsgroups: *comp.lang.ada*

[...] Yes, the gcc GNAT 3.1 and 3.2 that come with the mingw system are not very useful. However, after the freezing of the gcc 3.2 sources, a big update was done to gcc GNAT, and it is now a LOT better.

If you want to try for yourself, goto

<http://prdownloads.sf.net/mingw>

and download [gnatgcc-3.3-exp-20021006-notes.txt.gz](http://prdownloads.sourceforge.net/mingw/gnatgcc-3.3-exp-20021006-notes.txt.gz), [gnatgcc-3.3-exp-20021006-src.tar.gz](http://prdownloads.sourceforge.net/mingw/gnatgcc-3.3-exp-20021006-src.tar.gz) if you need the sources), and follow the instructions in the notes file. REALLY as it is not installed under the current mingw distribution.

Note: there is currently no such thing as a gcc 3.3, and this is strictly an experimental build. Although it is, IMO, much better than the 3.2 GNAT version on Mingw, an ACVC run still shows some problems.

[And from another message: -- dc]

[...] Note that by better, I mean that it functions better on Windows (less blowups), not that there are additional frontend bugs resolved.

3.14p is, IMHO, still the better compiler.

Jerry van Dijk, Leiden, Holland,
users.ncrvnet.nl/gmvdijk

References to Publications

"Ada, Interfaces and the Listener Paradigm" on line

From: *Jean-Pierre Rosen <rosen@adalog.fr>*
 Date: *Tue, 20 Aug 2002 10:50:34 +0200*
 Organization: *Adalog*
 Subject: *AE2002 paper available*
 Newsgroups: *comp.lang.ada*

By popular demand, my paper on "Ada, Interfaces and the Listener Paradigm" is now available from Adalog's publications page,

<http://www.adalog.fr/publica2.htm>

[Direct URL is <http://www.adalog.fr/publicat/ada-interfaces.pdf> - dc]

On-line GNU Ada Run-Time Book

From: *Javier Miranda <jmiranda@iuma.ulpgc.es>*
 Date: *Tue, 3 Sep 2002 18:46:47 +0100 (WET DST)*
 Organization: *ENST, France*
 Subject: *ANNOUNCE: Free GNU Ada Run-Time Book (version 0.7)*
 Newsgroups: *comp.lang.ada*

The upgraded version of my book on the GNAT Run-Time is already available. The book is now free under the terms of the GNU Free Documentation License,

Version 1.1 or any later published by the Free Software Foundation.

[See also same topic in AUJ 23-3 (Sep 2002), p.150-151. -- dc]

Title: A Detailed Description of the GNU Ada Run-Time.

Version: Beta 0.7.

Last Update: September, 3, 2002.

Author: Javier Miranda.

Access:

<http://www.iuma.ulpgc.es/users/jmiranda>

Chapters:

1. The GNAT Project
2. Task Types and Objects (upgraded)
3. The Rendezvous (upgraded)
4. Protected Objects (upgraded)
5. Time and clocks (new chapter)
6. Interrupts
7. Exceptions (new chapter)
8. Abort and Asynchronous Transfer of Control (new chapter)
9. Bibliography.

Goal: To provide to the Ada community an On-line book which can be used for research and for teaching.

Brief Description: HTML Document which describes the behaviour of the Run-Time of the free GNU Ada compiler (GNAT).

Its main features are:

It is free. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation.

The html version of the document is linked with the html version of the GNAT 3.14p run-time sources. This allows the reader to verify the contents of the book by clicking the html reference to the source file where each concept is handled by the Run-Time.

All the html files of the GNAT sources are also linked. This enables the reader a quick access to the point of declaration of the Run-Time variables and types, to follow the calls to subprograms, etc. and to go backward.

It is linked with the Annotated Ada Reference Manual (Technical Corrigendum 1). Any reference to the AARM has a link to the corresponding section.

Most of the papers referenced in the book are free accessible from the NEC CiteSeer Scientific Literature Digital Library (<http://citeseer.nj.nec.com/cs>)

Javier Miranda, University of Las Palmas de Gran Canaria, Canary Islands, Spain

From: Javier Miranda
<jmiranda@iuma.ulpgc.es>

Date: Fri, 6 Sep 2002 11:44:58 +0100
(WET DST)

Subject: ANNOUNCE: Free GNU Ada Run-Time Book (version 0.7)

I have upgraded the server [...] with a new file which allows to install the html sources of the book locally (in your preferred local directory) or remotely (in your preferred web page).

From: Javier Miranda
<javiermirandag@netscape.net>

Date: Tue, 10 Sep 2002 21:04:00 +0000
Organization: Universidad de Las Palmas de Gran Canaria

Subject: Re: ANNOUNCE: Free GNU Ada Run-Time Book (version 0.7)

Newsgroups: comp.lang.ada

I have upgraded the server [...] with the LaTeX sources (and figures) of the book. The full link is:

<http://www.iuma.ulpgc.es/users/jmiranda/gnat-rts/gnat-rts-freebook-sources-0.7.tgz>

Therefore, the current distribution contains: PS, PDF, DVI files (the full book ready to be printed); html sources (with one script which facilitates the local or remote installation of the html sources); LaTeX sources (which facilitates the collaboration to enhance the quality of the book).

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: Tue, 3 Sep 2002 14:25:36 -0700
(MST)

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

To: I9DK Sept 2002 Online News
<jcdk@ddci.com>

DDC-I Online News, September 2002, Volume 3, Number 8. A monthly news update dedicated to DDC-I customers & registered subscribers.

JTAG Debugging for SCORE PowerPC. Debug without costly on-board monitors or extraneous communication lines.

Ada95 Training & DO-178B Training

Tech Talk: Improved Interface Between SCORE GUI and SCORECast. Easy access to module testing tools using the SCORE IDE.

For the complete newsletter, go to http://www.ddci.com/news_vol3num8.shtml

From: JC <jcdk@ddci.com>

Date: Tue, 1 Oct 2002 17:48:16 -0700
(MST)

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

To: J9DK Oct 2002 Online News
<jcdk@ddci.com>

DDC-I Online News, October 2002, Volume 3, Number 9.

DACS Keeps JSOW Maintenance On Target at Raytheon. Read how DDC-I product champs helped this customer.

Third Party Update: Ada Information Clearinghouse. A helpful source for all kinds of Ada information.

Tech Talk: Using Path Commands in SCORE. The article shows how path commands provide flexibility in the location of source code

For the complete newsletter, go to http://www.ddci.com/news_vol3num9.shtml

From: JC <jcdk@ddci.com>

Date: Tue, 5 Nov 2002 16:56:56 -0700
(MST)

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

To: L9DK Nov 2002 Online News
<jcdk@ddci.com>

DDC-I Online News, November 2002, Volume 3, Number 10

FAA Certification - Tips from the Experts.

If you have a DO-178B requirement, don't forget your checklists!

DACS Network Bridge - New from DDC-I. Delivers valuable developer connectivity by enabling remote management for multiple targets.

Tech Talk: Floating Point Support for DACS-80x86. Floating point instructions... provided by either the hardware or floating point emulation software.

For the complete newsletter, go to http://www.ddci.com/news_vol3num10.shtml

Proceedings Ada-Europe 2002 Conference

From: Dirk Craeynest

<Dirk.Craeynest@cs.kuleuven.ac.be>

Date: Sun, 22 Sep 2002 23:54:24 +0200

Subject: Reliable Software Technologies - Ada-Europe 2002, Proceedings

To: team-ada@acm.org

(Sent per request of the Program Co-chairs. -- dc)

Reliable Software Technologies - Ada-Europe 2002, Proceedings

To the participants in the conference,

To the authors of contribution,

To all friends of Ad,

Thanks to all of you, our annual conference held this year in Vienna was very successful. As usual, the proceedings were published in the famous Lecture Notes of Computer Science of Springer-Verlag. To refer to these proceedings, e.g. when you want to quote an article published there, please use the following reference:

Johann Blieberger, Alfred Strohmeier (Eds.); Reliable Software Technologies - Ada-Europe 2002, Proceedings, Lecture Notes in Computer Science, vol. 2361, Springer-Verlag, 2002, 7th Ada-Europe International Conference on Reliable Software Technologies, Vienna, Austria, June 17-21, 2002.

For more about the volume, please see the web site of Springer:
<http://link.springer.de/link/service/series/0558/tocs/t2361.htm>

Looking into the future, the call for papers for our next conference is already available. For more, see:

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

Best regards,

Johann Blieberger and Alfred Strohmeier, program co-chairs

Introducing Ada on CodeMages Software Forum

From: CodeMage
<codemage@crystalcode.com>
Date: Tue, 24 Sep 2002 09:36:21 GMT
Organization: The CodeMages Community
Subject: [ARTICLE] Introducing the Ada programming language
Newsgroups: comp.programming

[See also "Ada Articles on New Software Forum" earlier in this AUJ issue. -- dc]

Jim Rogers has written a thorough introduction to the Ada programming language for the CodeMages Community web site, illustrating Ada's syntax and features with a lot of example code and by comparing Ada with better known languages like C++ and Java.

If you are interested to know more about this fascinating language then read the complete article in the Coding | Ada section of the CodeMages Community web site at:

<http://www.crystalcode.com/codemage/goto.php?url=IntroducingAda>

We would love to get some feedback on this article (and others) in the form of an e-mail to codemage@crystalcode.com (which will be forwarded to the author) or in the form of a discussion with the author and other members on the CodeMages Community Forum.

Also, our brand new developers community needs more serious contributors like Jim who can write high quality technical articles about software development or web design techniques. For more information and other power and safety of this language before we all drown in a C of errors.

Jerry Petrey, Senior Principal Systems Engineer, Raytheon Missile Systems

contribution possibilities please write to codemage@crystalcode.com or visit the CodeMages Community web site at

<http://www.crystalcode.com/codemage>

Thanks for your time,

The CodeMages Community

From: gswork@mailcity.com
Date: 24 Sep 2002 09:34:56 -0700
Subject: Re: [ARTICLE] Introducing the Ada programming language
Newsgroups: comp.programming

If I have the name right then Jim Rogers has contributed quite a bit on comp.programming, and some interesting code in the 'all language programming' threads of a couple of months back.

The article is well written, covering a lot of ground with clarity, it makes my interest in Ada even greater.

From: Jim Rogers
<jimmaureenrogers@worldnet.att.net>
Date: Tue, 24 Sep 2002 16:48:33 GMT
Subject: Re: [ARTICLE] Introducing the Ada programming language
Newsgroups: comp.programming

Yes. I am the same person. I want to emphasize that all opinions in my articles are my own and do not necessarily represent the opinions of any organization. In other words, whatever I said is my fault. Blame me and not the web site :-)

From: CodeMage
<codemage@crystalcode.com>
Date: Mon, 30 Sep 2002 22:55:45 GMT
Organization: The CodeMages Community
Subject: Re: [ARTICLE] Introducing the Ada programming language
Newsgroups: comp.lang.ada

[...] We have presently implemented a number of improvements to the appearance of the site in general and to the appearance of the technical articles in particular. Every technical article now has a plain version which doesn't use CSS and allows the reader to modify the text size or colors in any way that s/he wants.

The excellent "Introducing Ada" article written by Jim Rogers may now be viewed in the standard formatted version (<http://www.crystalcode.com/codemage/goto.php?url=IntroducingAda>) or in the plain version (click on the "Plain version" option at the top of the article) for improved readability. [...]

From: Preben Randhol <randhol@pvv.org>
Date: Tue, 1 Oct 2002 09:39:34 +0000 (UTC)
Organization: Norwegian university of science and technology
Subject: Re: [ARTICLE] Introducing the Ada programming language
Newsgroups: comp.lang.ada

From: Richard Riehle
<richard@adaworks.com>
Date: Fri, 04 Oct 2002 23:31:27 -0700

Could you later when you have time make this page valid:

<http://www.crystalcode.com/codemage/MainMenu/Coding/Ada/>

It would be nice if it had an overview of the stuff concerning Ada. I mean it is easier to direct people to a static address on say a web-page.

From: CodeMage
<codemage@crystalcode.com>
Date: Tue, 01 Oct 2002 23:02:55 GMT
Organization: The CodeMages Community
Subject: Re: [ARTICLE] Introducing the Ada programming language
Newsgroups: comp.lang.ada

Done... Phew, you are a demanding reader! ☺ Thanks to you there are now at least 3 ways to get an overview of the Ada articles on the site.

From: CodeMage
<codemage@crystalcode.com>
Date: Thu, 10 Oct 2002 23:12:03 GMT
Organization: The CodeMages Community
Subject: Re: [ARTICLE] Introducing the Ada programming language
Newsgroups: comp.lang.ada

A PDF version (raw and zipped) is now available for some of the articles on the CodeMages Community site, including "Introducing Ada" by Jim Rogers.

Download (or view) it by selecting the right option at the top of the article on the site, or directly from:
<http://crystalcode.com/codemage/goto.php?url=PDFAda>

Ada Distilled On-line Book

From: Jerry Petrey
<jdpetrey@raytheon.com>
Date: Thu, 03 Oct 2002 08:10:22 -0700
Organization: Raytheon Company
Subject: Re: [ARTICLE] Introducing the Ada programming language
Newsgroups: comp.lang.ada

Richard Riehle wrote:

> If you want a PDF document that describes Ada, you could download my little booklet, Ada Distilled, from www.adaic.org or www.adapower.com.

Jim Rogers wrote:

> Yes, Ada Distilled is much more complete than my relatively small article on the CodeMages site.

Your article is outstanding, Jim! It is a great, concise introduction to Ada.

Richard's is also extremely well done. I applaud both of you for helping to promote the language, especially at a time when so many outsiders think the language is dead. Hopefully, efforts like these will help more people C the beauty,

Organization: AdaWorks Software Engineering
Subject: Re: [ARTICLE] Introducing the Ada programming language

Newsgroups: *comp.lang.ada*

As a point of information, I am updating Ada Distilled (and correcting a few small errors that have been brought to my attention by diligent readers) with some additional examples. I hope to have the new version posted to adapower and adaic.org sometime in the November time frame.

Meanwhile, my thanks to the many people who have been sending me notes, comments, and suggestions for the on-going updates. I am still going to strive to keep the book to around 100 pages.

Ada Inside

USA - DACS Keeps JSOW Maintenance on Target at Raytheon

URL:

http://www.ddci.com/news_vol3num9.shtm

Subject: *DDC-I Online News, October 2002, Vol. 3 Issue 9*

DACS Keeps JSOW Maintenance On Target at Raytheon

Raytheon leverages legacy code from initial GEU development at Texas Instruments to maintain optimal system performance at minimum cost.

Using software originally developed by Texas Instruments, engineers at Raytheon maintaining the Guidance Electronics Unit (GEU) of the Joint Stand-Off Weapon (JSOW) one of the most valuable medium-range tactical weapons in the United States high-tech arsenal have come to respect DDC-I's DACS programming tools for safety-critical, real-time embedded systems.

An autonomous, air-launched glide weapon designed to attack a variety of ground targets from standoff range, the JSOW moniker actually represents a family of delivery vehicles sharing a common platform and variants with unique missions and payloads. A joint Navy and Air Force program, integration is ongoing for several aircraft, including the F/A-18, F-16, B-2, B-52, while also planned for a host of others in current and future inventories.

"The GEU is the operational core of the JSOW, responsible for guidance, navigation and control of the munition from launch to strike," explains Guy Yeager, a principal software engineer and the software team lead for the JSOW/GEU program at Raytheon. "Every system on the weapon in some way, shape or form communicates with the GEU."

According to Yeager, the GEU is literally the mind of the JSOW, taking inertial measurements from the IMU and performing a GPS correction to arrive at a

"nav solution." This data is fed to guidance and control, where necessary adjustments are determined and the data converted into autopilot commands, which drive the control system and set trim on the flight surfaces.

The current Raytheon JSOW program emerged from the Advanced Interdiction Weapon System (AIWS), a Navy program initiated in the mid-eighties. From the beginning, affordability was a critical element, and as a direct result the JSOW program has been aggressively managed within a strict unit-cost threshold established early in the program.

Required mission capabilities for the JSOW also created significant challenges, particularly with respect to the aggressive cost goals, but a persistent focus has driven the adoption of progressive acquisition strategies involving a combination of competition and cooperation.

Unlike some DoD programs that have used an "affordability" label to satisfy critics, the strict JSOW goals drove a number of key design and production decisions to produce components like the GEU at low cost.

"Where the main development of the GEU software was originally done at TI Systems, we're in the maintenance phase now," continues Yeager. "What we're doing is porting the old assembly code to Ada for a 486 processor using DDC-I's DACS cross-compiler."

A strong advocate of Ada, especially for safety-critical applications, Yeager and his team know first-hand how features like robust exception handling, multi-tasking capability and built-in run-time contribute to cost-effective software programming for real-time embedded systems.

"Ada's strong typing and other characteristics seem very limiting to programmers accustomed to languages like C++, but if it's for a safety-critical system I don't want creativity. Ada is much more self-regulating in that respect. Even if you know what you're doing, it has intelligent checks integrated into the language that ultimately produce fewer mistakes in the code," he adds.

Hosted on a UNIX platform, only the DACS cross-compiler is currently in use. However, already in the process of upgrading to the latest version of DACS to eliminate the need to use a second assembler to work with the legacy code, he explains that DDC-I chose to upgrade their DACS toolset specifically to recognize the extensions familiar to their other assembler and make it easier and less expensive for his team to manipulate their legacy code. In addition, they look forward to gaining debug capabilities for their target processor. [...]

The Netherlands - Military Application on iPAQ with JGNAT

From: *Rob Veenker <veenker@xs4all.nl>*

Date: *Sun, 13 Oct 2002 14:43:29 +0200*

Subject: *Re: PDA Ada*

Newsgroups: *comp.lang.ada*

> My colleagues and I are longing for an Ada compiler for many years now. Sadly, we have to develop PDA application using C and Java. Yucks! We want Ada for PDAs!

The JGNAT solution also applies to Pocket PC devices such as the iPAQ. We are about to finish a military application that runs on a ruggedized iPAQ. It uses a military radio for data communications and this protocol is 100 % Ada. Only the GUI part is Java. Because the protocol was implemented in Ada, We could use the same code on a Windows NT box :-)

[See also "Netherlands - JVM on PDA to Drive Communications Radio" in AUJ 22-1 (Mar 2001), p.19. -- dc]

From: *Wes Groleau*

<wesgroleau@despammed.com>

Date: *Mon, 14 Oct 2002 10:33:45 -0500*

Organization: *Raytheon Company*

Subject: *Re: PDA Ada*

Newsgroups: *comp.lang.ada*

> Are you funding ACT for supporting JGNAT. The last I heard is that this product is not supported anymore due to lack of customer interest. The current version also does not run anymore on recent Java versions.

However, there are other Ada to J-code compilers. And the GMGPL would allow you to use the Java lib bindings to the J2ME (IF for some reason the other vendor's bindings were not adequate).

From: *Rob Veenker <veenker@xs4all.nl>*

Date: *Tue, 15 Oct 2002 13:43:17 +0200*

Subject: *Re: PDA Ada*

Newsgroups: *comp.lang.ada*

The radio concerned is of the PR4G family (aka FM9000) by Thales. It uses a protocol called SYCOMORE, used for remote control of the radio itself. This protocol however also allows (limited) data communication. On the iPAQ a SYCOMORE protocol stack written in Ada was implemented using JGNAT.

To allow messaging, we also wrote message acknowledgement, dispatching and marshaling functionality (all in Ada as well). The result is a communication 'jar' (Java archive) that is used by a Java GUI.

We are indeed using the supported version of JGNAT and are aware of the JVM problems. The JVM's available on the iPAQ have not shown this problem (Insignia/Jeode and NSI/CrEme). Whether this becomes a problem is still unknown.

Also the Java standard for small devices (Personal Java) differs from the JDK. The

JVM on the iPAQ is not outdated, it's just different (sort of a subset).

Since the protocol is implemented in Ada there are alternatives like going native on the iPAQ using VxWorks/GNAT :-)

*From: Rob Veenker <veenker@xs4all.nl>
Date: Mon, 21 Oct 2002 08:33:01 +0200
Subject: Re: PDA Ada
Newsgroups: comp.lang.ada*

> I was just wondering how you implemented the messaging in Ada so that it can be compiled by JGNAT. The message standards I know (e.g. Link 16 J-Messages) are heavily bit oriented and people tend to implement this via Ada records with lots of representation clauses. But as far as I know JGNAT does not support representation clauses (correct me if I am wrong). So how did you do that?

Most of the low-level stuff is done using byte arrays and logical AND's and OR's ☺

The records are really not that complicated but I admit that representation clauses would make code better readable and maintainable. At least Ada does have hexadecimal notation and bitwise operations.

USA - Software Controlled Radio

*From: Mike Brenner <mikeb@mitre.org>
Date: Wed, 23 Oct 2002 09:14:15 -0400
Subject: another link for the software
To: team-ada@acm.org*

Another link for the Software Controlled Radio (one of the subcontractors is delivering their software in Ada) that has more information [on the Joint Tactical Radio System (JTRS) -- dc]:

http://www.herbb.hanscom.af.mil/esc_opps.asp?rfp=R459

*From: Victor Giddings
<victor.giddings@ois.com>
Date: Wed, 23 Oct 2002 11:14:26 -0400
Subject: Re: another link for the software
To: team-ada@acm.org*

JTRS is not the only story in Software Defined Radio, although it is "pathfinder". More information, especially about commercial interest, can be found at <http://www.sdrforum.org/>.

JTRS is multi-language and Ada is included. The interfaces for the common software infrastructure is defined by CORBA IDL.

Victor Giddings, Senior Product Engineer, Objective Interface Systems

Europe - Objektum Wins Major Eurofighter Training Contract

*From: Jackie Bissell
<jackie@realitypr.co.uk>*

*Date: Thu, 24 Oct 2002 09:43:57 +0100
Subject: Press Release - Croydon based company to play a major role in the maintenance of the Eurofighter
To: Jackie Bissell <jackie@realitypr.co.uk>*

PRESS RELEASE

Training and consultancy expert Objektum fights off the competition to land a major contract with EADS and the Luftwaffe

Croydon based company to play a major role in the maintenance of the Eurofighter

Date: 24 October 2002. Objektum (www.objektum.com) the training and consultancy expert, has been appointed by the European defence giant EADS to deliver a series of intensive training courses and workshops to the Luftwaffe, designed to provide its software team with the skills they need to maintain its fleet of Eurofighters.

Objektum is an independent provider of tailored training, consultancy and mentoring, specialising in Object Oriented, Real-Time and Embedded systems for the defence and aerospace industry. It will be using its expertise in the Eurofighter's technology to deliver specialist training to the Luftwaffe's engineers.

Training will focus primarily on the complex subsystems as well as the engineering skills required to develop the aircraft's software, such as CORE, HOOD and Ada. This is a fundamental part of a major five-year project, starting at the end of the year, to provide the German air force with the skills necessary to independently maintain and operate its fleet of aircraft.

"The right tools and technical expertise will enable the Luftwaffe to support its fleet of Eurofighter aircraft in the National Support Centre. With the help of Objektum we are on the way to training the first crew of Eurofighter System Engineers. The extensive knowledge of Eurofighter technologies and the training experience of Objektum is a fundamental part of the project." Said Sven Koenig, EADS Training Manager.

"We have been working on a number of projects involving the development of the Eurofighter over the years and are delighted to offer our expertise on the subsystems and software technologies used in this aircraft." Said Derek Russell, Technical Director of Objektum. "This appointment is testament to the skills, knowledge and expertise of our team in the subsystems and software technologies used by the defence and aerospace industry and we are looking forward to training the Luftwaffe." Added Andrew Bissell, Sales Director of Objektum.

[See URL for more info on Objektum: -- dc] www.objektum.com

For more information and photography contact: Andy Brown, Reality PR, 020 8658 5916, andy@realitypr.co.uk

Indirect Information on Ada Usage

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

*From: tmoran@acm.org
Date: Thu, 15 Aug 2002 17:39:20 GMT
Subject: Re: Information Systems Annex
Newsgroups: comp.lang.ada*

Some years ago I did a system for an outfit that sold historical stock and commodity price data. Some things were in pennies, some (like stocks) in eighths, some in quarters, etc. IIRC we defined a fixed point type with the LCM of those things as its 'delta and then it was a simple type conversion to move the odd prices in/out of that fixed point type.

*From: Bonnin Hugues
<hugues.bonin@thales-avionics.com>
Date: Mon, 19 Aug 2002 15:33:16 +0200
Organization: Thales-avionics
Subject: pb. with big object code
To: GNAT Discussion List
<gnatlist@lyris.seas.gwu.edu>*

I have a very big Ada unit to compile (one Library Unit in 2500 files, and 400 KLOC), and the compiler (GNAT 3.13p or 3.14p under NT) has problems to generate the object [...]

*From: Daniel Altbach
<daltbach@bluewin.ch>
Date: Tue, 3 Sep 2002 16:12:43 +0200
Subject: Job Offer*

[...] Several Ada Programmers - Belgium

For a project based near Brussels related to Railway Signaling Systems we are searching Ada Programmers. [...] Job: 3 years.

Function: production of detailed design, software and documentation; implementation (design, code and test); analysis of change requests; participation in the validation of the design; documentation and execution of system tests.

Profile: university level; 3 years experience in the development of application software; knowledge of Ada language, experience in Real Time is an asset; experience of software development in UNIX environment; any experience with functional analysis methodology is an asset. [...]

*From: bdesth@free.fr (Laotseu)
Date: 16 Sep 2002 00:27:39 -0700
Subject: Re: Compileur Ada
Newsgroups: fr.comp.lang.ada*

[Extracts translated from French: -- dc]

Two weeks ago, I got curious about Ada. Approximately 45 minutes later, I had GNAT and AdaGIDE installed, two or

three tutorials, and I was busy writing the classic "Hello World"...

Google is your friend : <http://www.google.fr>

From: postman@jobscareer.be

<postman@jobscareer.be>

Date: Fri, 04 Oct 2002 08:08:14 +0200

Subject: Your weekly jobscareer.be postman

Ada Software Engineers, Brussels [...]

You will be integrated in a strategic project including architectural & detailed design, development of the application, programming, testing and writing of the design documentation. [...] Knowledge of a programming language preferably Ada 83-95 or C++. [...]

From: Pascal Leroy

<pleroy@rational.com>

Date: Wed, 9 Oct 2002 02:38:10 +0200

Organization: Rational Software Corp.

Subject: Re: [Ada-Comment] Update to AI-302: unprotected unbounded bags

To: Ada-Comment List <ada-comment@ada-auth.org>

[...] Many of our customers use one flavor or another of the Booch components, but then our customers are surely atypical in this respect, because for many years we actively promoted the Booch components.

These components may not be perfect, but at least we know that they have been intensively used by many projects for many years.

From: Pascal Obry <p.obry@wanadoo.fr>

Date: Wed, 9 Oct 2002 15:18:53 +0200

Subject: Re: [Ada-Comment] Update to AI-302: unprotected unbounded bags

To: Ada-Comment List <ada-comment@ada-auth.org>

[...] Maybe I'm biased because I'm in the IT field. (Yes I use Ada for non safety critical stuff mostly for Web development :) [...]

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

<http://perso.wanadoo.fr/pascal.obry>

From: Pierre Habraken

<Pierre.Habraken@imag.fr>

Date: Wed, 09 Oct 2002 11:36:58 +0200

Organization: Universite Joseph Fourier

Subject: Re: your command request

To: GNAT Discussion List

<gnatlist@lyris.seas.gwu.edu>

The aim of what I want to do is to provide our students (undergraduate level) with a Posix programming environment (including Ada) that they can easily install and use at home on their own machine. [...]

We use Ada as the main language for teaching algorithmics, data structures and software design. [...]

Pierre Habraken, IMAG-LSR, Saint Martin D'Heres

From: Roger Racine

<rracine@draper.com>

Date: Wed, 23 Oct 2002 10:49:56 -0400

Subject: GNAT 3.14p on Windows

To: GNAT Discussion List

<gnatlist@lyris.seas.gwu.edu>

[...] I am helping teach a course over at a local college (MIT). I gave the students a reasonably large program (at least compared to what they are used to), with a couple of missing pieces given to them as null procedures. Their term assignment (among other things) is to fill in the missing pieces. [...]

The program has a very complex elaboration order (I need to use the -gnatE compiler option, with pragma Elaborate in various packages, to avoid Program Error). I ported this program from Ada83 (it compiled and ran originally using both DEC VAX and Alys Ada compilers). [...]

Roger Racine, Draper Laboratory, Cambridge, MA, USA

From: Manuel Collado

<mcollado@lml.ls.fi.upm.es>

Date: Fri, 18 Oct 2002 10:24:34 +0200

Subject: Porting from Modula-2 to Ada
Newsgroups: comp.lang.ada

We are porting some legacy Modula-2/C code to Ada. The code uses low-level facilities from Modula-2 and C. We would like to port it to clean Ada, without interfacing to C code. In addition we would like to keep the old interface unchanged (if possible).

At present we are looking for ways to manage memory and binary files as arrays of bytes. [...]

From: Vadim Godunko

<vgodunko@vipmail.ru>

Date: Fri, 11 Oct 2002 09:46:30 +0400

Subject: Re: PDA Ada

Newsgroups: comp.lang.ada

> Has anyone besides ACT ever ported the gnat runtime to any target?

Some time ago I port gnat to x86/FreeBSD. If your system is POSIX compatible this is simple process.

Vadim Godunko, North-Caucasian Area Air Traffic Control Center

Ada in Context

C++ Cognoscenti Consider Method(Object) Notation Superior

From: Hyman Rosen <hyrosen@mail.com>

Date: Wed, 14 Aug 2002 16:35:48 -0400

Organization: KBC Financial Products

Subject: OO dot notation

Newsgroups: comp.lang.ada

For all those who have been arguing that Ada would be much more successful if it allowed the object.method() syntax, it might be interesting to check out some of the goings on in

comp.lang.c++.moderated, in the 'Guru of the Week #84' threads.

Many of the C++ cognoscenti are coming to the conclusion that the method(object) notation is superior, and indeed that as few functions as possible should be made methods of a class.

From: Pat Rogers

<progers@classwide.com>

Date: Thu, 15 Aug 2002 04:03:26 GMT

Subject: Re: OO dot notation

Newsgroups: comp.lang.ada

Interesting article (I only read the first). A few comments...

1) I take your meaning in the above, I think. You're saying that the distinguished receiver syntax is not so big a selling point because there are good reasons (at least, according to the article) not to make every function a member of a class and in those cases the notation will not be available anyway. Right?

As you know, the Ada syntax doesn't affect in any way whether or not a procedure or function is a "member function". It's all a matter of where the subprograms are declared and with what parameter and result types. But the article is not discussing syntax anyway.

2) I would argue that the article is really about abstract data type programming. ADT's have, of course, been around for a long time and have always been fundamental to Ada. So the design choices described by the article and the position taken by the author are not new to the Ada crowd. (I hope that does not come across as condescending; it is not meant to be.) Booch's first Ada (83) book discussed this years ago, when Ada only had ADTs and a limited form of inheritance: you only made subprograms primitive if they required direct access to the representation.

3) The design decision certainly applies to OOP objects as well as to ADT objects, since they are the same thing in so many ways. In Ada, this is especially so because the language has explicit class-wide types and class-wide subprograms. The decision of what subprograms should be class-wide instead of primitive to a specific type is based on the criterion described in the article: can we reasonably write this subprogram in terms of the existing primitive operations (using dynamic dispatching)? If so, we do indeed make a much more flexible subprogram and increase encapsulation without loss of performance.

4) OK, I cannot entirely resist the temptation to say that the article indicates there may yet be hope for those class-oriented language guys! :-)

Thanks for pointing out the thread!

Patrick Rogers, <http://www.classwide.com>

From: Jim Rogers

<jimmaureenrogers@worldnet.att.net>
Date: Fri, 16 Aug 2002 19:44:19 GMT
Subject: Re: OO dot notation
Newsgroups: comp.lang.ada

I had been following this thread a while back. It seems that one of the big issues in C++ is the use of functions inherited from a template parameter. If the function is part of a class the object.method() syntax is required while the method(object) syntax is required if the function is not part of a class. This causes genuine problems for C++. The cognoscenti mentioned above have determined that the method(object) syntax is the most general, and this leads to the conclusion that functions should not be made members of a class unless absolutely necessary.

My guess is that the C++ language will not be changed for this. Instead, people will simply be advised to limit their use of functions defined within a class.

I assume that this tends to weaken a design's OO nature. It will result in a lot of basically global functions. This can result in weaker encapsulation of data within classes, and many other such problems.

From: Warren W. Gay VE3WWG
<ve3wwg@cogeco.ca>
Date: Fri, 16 Aug 2002 12:00:39 -0400
Subject: Re: OO dot notation
Newsgroups: comp.lang.ada

> It seems to me that the world isn't "mature" for Ada yet like Unix wasn't mature for the masses years ago until the first affordable powerful PCs came up. Maybe in the future Ada will enjoy a "revival" when people demand more powerful languages and discover the advantages of Ada.

I like this answer. Additionally, people used to cite the compiler speed and cost. Now with faster CPUs and GNAT being freely available, both of these objections are now irrelevant [...]

Now Ada must wait for the C/C++ world to humble themselves to the fact that Ada was ahead of its time in most ways ☺

A Business Case for Ada

From: John R. Strohm
<strohm@airmail.net>
Date: Sat, 17 Aug 2002 09:53:15 -0500
Subject: Re: On accounting and engineering
Newsgroups: comp.lang.ada

> The major issue is \$\$\$. Accountants are more interested in RoI (Return of Investment) [...]. When I proposed to my company to switch to Ada many years ago, the first question was "What will be the RoI and benefits over development cost if we switch to Ada?"

That's where you whip out a few PowerPoint charts. Start with the one headed "Pratt&Whitney", with the two bullets "2X increase in programmer

productivity" and "4X DECREASE in defect density".

> Other issues will be like the cost of training and development and maintenance cost which cannot be estimated as the company has not yet established/experienced any track records using the new tool (Ada, could be).

There, you whip out the chart, straight from AdaIC, that points out that competent programmers typically need two weeks of training and they are up and running in Ada. [...]

[For both issues see also "On Languages, Productivity and Quality" in AUJ 22-4 (Dec 2001), pp.222-224. -- dc]

Ada as your Personal Assistant

From: Juha Valimaki
<juha.valimaki@xdsloy.com>
Date: Mon, 19 Aug 2002 23:53:56 +0300
Subject: Re: Would you use Ada more if...
Newsgroups: comp.lang.ada

Well, I would like to choose Ada... I have been using mostly C/C++ so far, but some time ago I discovered Ada, and I like it. It's just a language and there are many other factors that affect productivity, so Ada itself may not be the silver bullet, but it would help. I like to think that Ada is like my personal assistant that can point out mistakes and guide me. All the effort that has gone to designing the language and tools is available every time I compile my code. It's like having an extra worker :-)

Ada vs. C#

From: Jim Rogers
<jimmaureenrogers@worldnet.att.net>
Date: Tue, 03 Sep 2002 03:31:24 GMT
Subject: Re: Alternatives to ailing Ada.
Newsgroups: comp.lang.ada

> Now there is Visual Studio.NET. C# totally blows Ada away.

Visual Studio.NET is an environment, not a language. C# has some features of Ada. It has a lot of features of Java.

I would not look to C# for efficiency. Speaking of "out of the box" experiences, look at the C# concept of "boxing". This, in summary, is the tendency of C# to convert primitive types to instances of a class implicitly. That is, you have no control over the timing of this dynamic allocation. To a naive programmer this makes C# look easier to use. It certainly removes important levels of control and efficiency from the programmer.

The C# threading model is a strange combination of Java threads, WIN32 threads, and a very questionable implementation of a monitor.

C# still does not allow you to define your own primitive types like Ada. C# has no built in interface to any other language except through the .NET interface. C# must have the .NET virtual machine to run, ruling out any RTOS.

None of this blows Ada away.

Microsoft continues to state that .NET will be implemented on additional operating systems. So far that is just hype. .NET, and by association, C#, are prisoners of Microsoft operating systems. Ada is proven to work well across a wide range of operating systems.

On the Value of Strong Typing

From: nma124@hotmail.com (steve_H)
Date: 12 Oct 2002 18:18:39 -0700
Subject: Who said strong typing is a benefit?
Newsgroups: comp.lang.ada

Since many Ada programmers mention as one of Ada benefits is its strong typing, I thought some here might find the following 'interesting' to read.

I guess this is an argument of '4GL' vs '3GL'. The following seems to imply that not having to declare variables and having the ability to mix any variable with any in an expression is a 'benefit'. I am not sure I agree with this, on the other hand, one can not deny that one can write code faster in matlab than in a strongly typed language such as Ada specially for scientific applications where arrays and matrices are heavily used. [...]

http://www.mathworks.com/company/digest/sept02/accel_matlab.pdf

"MATLAB Type Handling. An important benefit of MATLAB is that users do not have to declare variables to be of certain data types, as is required with 3GLs. In MATLAB, any variable can be assigned a value of any type, and that type can be changed implicitly at will because of an assignment to a new value of a different type. As a result, the MATLAB interpreter is prepared to deal with the most complicated data types (such as an ndimensional array of complex doubles) and is capable of performing operations no matter what the actual data types turn out to be at run-time"

From: Jim Rogers
<jimmaureenrogers@worldnet.att.net>
Date: Sun, 13 Oct 2002 02:53:55 GMT
Subject: Re: Who said strong typing is a benefit?
Newsgroups: comp.lang.ada

Please be sure you do not confuse dynamically typed languages with weakly typed languages. Ada is a statically typed, strongly typed language. Perl is a dynamically typed, weakly typed language. Lisp is a dynamically typed, strongly typed language.

The benefits of each of these approaches is a matter of taste or need. Perl is most useful when development speed for text manipulation problems is your major concern. Lisp is beneficial for general data grouping and parsing problems. Ada is a beneficial when you must be able to determine code correctness before running the program.

I have contracted for a not to be named telecom company that treated all software as a disposable commodity. They were more interested in redoing bad programs than writing them correctly the first time. They believed they needed to get *something* to market, and that was more important than getting *quality* to the market. If that is your goal then Ada offers you very little. If your goal is to get a correct, high quality product to market quickly then Ada provides some genuine benefits.

From: Richard Riehle

<richard@adaworks.com>

Date: Sat, 12 Oct 2002 20:24:17 -0700

Organization: AdaWorks Software Engineering

Subject: Re: Who said strong typing is a benefit?

Newsgroups: comp.lang.ada

Although Ada is designed to be type-safe, this is not a problem for anyone who has no need of type-safety. It is so easy to circumvent type-safety, if you really want to do that. We start with a default of type-safe and simply relax that default as much as we wish.

More important, it is rare for anyone to have difficulties with Ada's type model. Any practical programmer understands the notion of type and can write code accordingly. So what is the problem, then?

More programmers have trouble with the Ada visibility rules than any other single aspect of the language. The type model supports the visibility rules. With this observation, I include the accessibility rules, which are simply a variation on visibility. In my experience, once programmers understand the visibility rules and learn to use them to their advantage, instead of fighting them at every turn, they become comfortable with Ada and begin to enjoy using it. The visibility rules are a benefit of the language design, not a curse, but it is hard to see that in the early stages of using the language.

Ada as a Real-Time Language

From: Thierry Lelegard

<thierry.lelegard@canal-plus.fr>

Date: Wed, 28 Aug 2002 17:44:26 +0200

Organization: CANAL+ Technologies

Subject: Re: Ada as a real time language

Newsgroups: comp.lang.ada,comp.realtime

> I have heard that the Ada Run Time doesn't handle load that well. If you have to many tasks or to much input/output the Runtime gives in so to speak. In that regard you should be better of with a RTOS. I have not any experience myself with large Ada programs. Anybody out there who could comment?

There is no such thing as "THE" Ada Run Time. There are as many implementations of an Ada Run Time as compiler/platform combinations.

Using DEC Ada (Ada83) on OpenVMS systems, we had applications using up to 6000 tasks in one single process, with a high I/O rate, which run quite smoothly. You can achieve this with a robust and versatile O/S and a perfect integration of the Ada Run Time with the underlying O/S.

From: Jerry Petrey

<jdpetrey@raytheon.com>

Date: Wed, 28 Aug 2002 17:47:58 -0700

Organization: Raytheon Company

Subject: Re: Ada as a real time language

Newsgroups: comp.lang.ada,comp.realtime

We use DDC-I Ada with its runtime on a 72 Mhz 80486 in the missile I work on. It has about 32 tasks and handles all our navigation, guidance and control without any problems.

Jerry Petrey, Senior Principal Systems Engineer, Raytheon Missile Systems

From: dennison@telepath.com (Ted Dennison)

Date: 28 Aug 2002 10:27:47 -0700

Subject: Re: Ada as a real time language

Newsgroups: comp.lang.ada

> I was just wondering about the real-time quality in the Ada language. Is it possible to write a good hard real-time system without a RTOS, or is it necessary to use a RTOS to achieve good real-time properties?

We wrote a military-grade flight simulator, including its real-time scheduler, using no OS calls whatsoever (with the exception of one call to set the RTC frequency). Thus was under vxWorks, but all synchronization and scheduling was done using standard Ada calls, and thus should be portable to any system that supports the real-time annex (and can cycle fast enough). We ran the same code under NT with no trouble, other than the trouble one would expect when using a non-real-time OS.

Its certainly easier to do this kind of thing in Ada than it is using any other language.

Ada IDEs vs. VisualC++

From: dennison@telepath.com (Ted Dennison)

Date: 6 Sep 2002 23:59:41 -0700

Subject: Re: The Ada experience.

Newsgroups: comp.lang.ada

> I have never used the pro version of GNAT. [...] However, the free version with the various free IDEs is abysmal. It feels like something out of the sixties.

[See also "GNAT IDEs" earlier in this AUJ issue. -- dc]

I'm guessing you are comparing it to VisualC++, no?

I suppose it depends on your perspective. For me, developing using VisualC++ (version 6.0) feels like playing with a toy compared to the Gnat toolkit. The editor is nice looking, but not nearly functional enough. The compiler doesn't implement the language properly (or even very well). Template support is a particular disaster. It has gawd-awful error messages, which causes the user way more work than should be necessary to figure out what they did wrong (its not uncommon for something simple like a missing semicolon to pull an error somewhere deep in a standard library header file). It comes with a semi-functional non-standard version of the STL. To top it off, its chock full of code-generation bugs, which act as little day-wasting land mines for you to periodically step on. And good luck trying to get Bill G. to fix them for you, or getting the sources to fix them yourself.

The code builder is too moronic to figure out recompilation dependencies without manual user intervention (which is just begging for errors). Maintaining all those dependencies manually in a large project is simply impossible.

But I'll grant you, its a nice *shiny* toy. :-)

From: Marin David Condic

<mcondic@acm.org>

Date: Sat, 07 Sep 2002 09:55:28 -0400

Subject: Re: The Ada experience.

Newsgroups: comp.lang.ada

Your complaints about VisualC++ are fair, but there are two perspectives to look at. MSVC++ is very feature-rich and well integrated. Gnat as a compiler is a command-line oriented thing with some add-on tools that might make it a bit more of an IDE. (Waiting to see what GPS provides.) In this sense, I can understand the complaint about it looking like something from the '60s in comparison to MSVC++.

The other perspective is that even though MSVC++ is a feature-rich, well integrated set of tools, the execution thereof can be quite poor. Throw on top of it the naturally difficult syntax/semantics of C++ and all the traps that implies plus the awkward and butt-ugly cricrap that MSVC++ sticks into the code so it can connect up to the GUI building thing and you've got a royal mess. It can make you yearn for the '60s where all you wanted to do was develop for a command line. :-)

*From: Marc A. Criley
<mcq95@earthlink.net>
Date: Mon, 09 Sep 2002 11:41:57 GMT
Organization: Quadrus Corporation
Subject: Re: The Ada experience.
Newsgroups: comp.lang.ada*

Ditto.

I've been working in the VC++ environment for a year and a half now, and I continue to bang my head up against it. (Though no small part of that is due to the idiosyncrasies of Windows(tm) as well.)

I'm not stupid, and I've worked on a variety of platforms and IDEs over the last nearly 20 years (Univac, VAX, Concurrent mainframe, various Unices) and never had the ongoing struggle I experience with Windows and VC++. It's not the C++ that hits me, it's the (for lack of a better phrase) Windows paradigm.

I was talking to my wife about this the other day, and she observed that when programming Windows and VC++ it doesn't matter much how smart you are, or skilled, or what you've studied; the only thing that matters is what experience you have with that environment, so that you can draw on that to work around, or avoid, problems.

*From: tmoran@acm.org
Date: Mon, 09 Sep 2002 17:36:55 GMT
Subject: Re: The Ada experience.
Newsgroups: comp.lang.ada*

When we started developing CLAW (Class Library for Ada on Windows), part of the intent was to make Windows easier for Ada programmers by presenting a surface allowing use of Ada techniques and paradigms. Nowadays I realize the (painfully gained) knowledge of Windows and workarounds embedded inside the CLAW routines is every bit as important.

Best PR for Ada: Tell the World about its Successes

*From: Richard Riehle
<richard@adaworks.com>
Date: Mon, 19 Aug 2002 10:10:01 -0700
Organization: AdaWorks Software Engineering
Subject: Re: Would you use Ada more if...
Newsgroups: comp.lang.ada*

[...] At present, there is no significant effort within our industry to raise the level of visibility for Ada. [...] The best way to get an Ada article published is to write about something else and "Oh, by the way, we did this in Ada." Even the large number of stories written about the Boeing 777 rarely mention that the language used was Ada. However, when an Ariane V explodes, everyone is quick to note that it was programmed in Ada. We really need a more effective effort in PR. [...]

ACT is doing its part by providing cost-effective options for entering the world of Ada. [...]

*From: Marin D. Condic
<not.valid@acm.org>
Date: Mon, 19 Aug 2002 13:50:30 -0400
Subject: Re: Would you use Ada more if...
Newsgroups: comp.lang.ada*

[...] I am convinced that better PR is (a) not likely to happen and (b) not likely to change anything. We in the Ada community tend to stand around saying "Look, its a really great language..." - which may be true, but is not nearly so compelling as "Here's this spiffy thing I built with Ada...." or "Here's how I made my first million dollars with Ada..." (The two possibly being closely related.)

You observe that articles about other things with an "Oh, by the way, it was done with Ada..." line is right on track. People want to see that something cool was done and will be less concerned about the language. If what was done was truly cool, they'll pick up the code and start fiddling around with it and you'll find new converts to the language that way. I doubt people want to read dozens of articles about how cool *any* language is. They'd rather read an article on how someone implemented some tricky math thing or clever navigation thing or internet web thing and if they discover that it was done with Ada, they'll have an incentive to learn something about the language.

Likewise, if a software product is built and a successful business is established selling it (or related services & support - it amounts to the same thing) that's the sort of thing that people will take notice of. If Ada is *really* better at building mousetraps, then it would seem that a little entrepreneurial activity along those lines will be rewarded. When someone can stand up and say "We're making more money than our competitors and one of the big reasons for that is our software quality due to Ada..." its going to get noticed and imitated. When someone dreams up a new electronic gadget and programs it in Ada, it will help create that critical mass. When someone develops a new business system and programs it in Ada, they'll create more Ada jobs and more interest in the language will result. The way for Ada to become a market force is for the Ada-philes to construct a market that uses Ada.

When that happens, the PR will be a natural outflow. Waiting for some organization to start a PR campaign is probably fruitless, but creating success will breed more success and with it will come the favorable PR.

Marin David Condic, Senior Software Engineer, Pace Micro Technology Americas, www.pacemicro.com

On Free Software and Software Economics

*From: dewar@gnat.com (Robert Dewar)
Date: 22 Aug 2002 03:28:47 -0700
Subject: Re: Ada 95 for an ARM-based bare board?
Newsgroups: comp.lang.ada*

[On remarks that ACT doesn't provide a specific GNAT port: -- dc]

Just because Ada Core does not market something does not necessarily mean it can't be marketed.

After all, no doubt one can make money selling Halloween costumes, but Ada Core Technologies is not in that business last time I looked :-). And more seriously, we can't even expect to cover all viable aspects of the commercial Ada market. Successful companies specialize in what they are good at. We consider our strength to be in providing high level support for mission critical projects, since that's where we see the major business opportunity. That's a totally different business from going after a mass market. Even if the mass market is viable, going after such a market is not something that fits our expertise and business model.

This kind of specialization is definitely a key to success. 2001-2002 was a difficult period for many (almost all?) high tech companies, but Ada Core Technologies has prospered during that period. We have experienced steady growth for this period in both companies (Ada Core Technologies and ACT/Europe) and we have a dedicated group of people, who may not be getting rich with stock options, but they are paid respectable and steady salaries :-).

I have run into a few people who are puzzled by our financial model, because they don't see how the founders expect to make big money out of the company. It's a sad sign of the times that people assume this is the motive for founding a company, and the founders, who are not getting Enron-style-rich, but are also paid respectable and steady salaries are quite content with the situation thank you :-)

Once again, the nice thing about Free Software is that we don't lock up the market by stopping anyone else from using or exploiting our technology. If someone thinks they can be successful selling an inexpensive ARM port of GNAT to the mass market, then I say go for it!

Robert Dewar, Ada Core Technologies

*From: dewar@gnat.com (Robert Dewar)
Date: 23 Aug 2002 18:58:06 -0700
Subject: Re: Software Economics was Re: Ada 95 for an ARM-based [...]
Newsgroups: comp.lang.ada*

[On the remark that providing tools for free "would decrease the value of intellectual property": -- dc]

[...] First point, ACT is definitely in favor of maximizing the value of its property. A company the size of ours (Ada Core and ACT/Europe between them have over 40 full time employees) needs a considerable level of income to support our activities and the ongoing development of GNAT and its surrounding tools.

Second point, the only difference between Microsoft and ACT is the license we use. We both sell licensed copyrighted software.

We use the GPL as our license because we think it maximizes the value of the product to our customers. We think restrictive licenses such as those used by Microsoft [...] are bad business because they don't provide the customers with what they need.

We think it's good business to provide the customers what they need. And the result of providing customers what they need is that we prosper. So we are using the GPL *because* it allows us to support our activities.

[...] the absurd statement that GNAT is a monopoly. This shows a level of unawareness of the market that is truly staggering. In fact there is fierce competition in the Ada market, as anyone who is in fact involved in the market as a vendor or a procurer of Ada products knows perfectly well. [...]

Robert Dewar, Ada Core Technologies

Date: 23 Aug 2002 19:19:07 -0700

Subject: Re: Ada 95 for an ARM-based bare board?

Newsgroups: comp.lang.ada

[On complaints that "ACT's business model discriminates small development teams". -- dc]

What an odd word to use. We don't discriminate against anyone, we simply don't serve all segments of the market. We succeed because we don't try to be all things to all people. Of course all users don't fit in the same business model! What get's companies in trouble is when they try to serve multiple markets at the same time and the markets get mixed up (consider the airlines trying to sell expensive seats to businesses and the same exact seats cheap to tourists).

If there is a viable business in providing low cost products to small development teams, then someone should by all means step in, but please don't expect Ada Core Technologies to serve all needs of the Ada community. We can't and we won't try.

We are Ada enthusiasts, and we immodestly believe that the continued viability of Ada Core Technologies, now with a complement of over 30 highly experienced engineers dedicated to the continued support and improvement of Ada, is critical to the continued success of Ada.

We are thus very conservative in how we approach the market, and so far this conservative approach has paid off with steady (but modest and manageable) growth, through a period in which other less conservative high-tech companies have seen a huge boom-and-bust cycle.

We have increased revenues in the last year, and we have a comfortable but modest buffer in our bank account. We don't have zillions in stock options, and I am sorry to report that the CEO and other founders have not been able to cash out for millions of dollars, no doubt making us a failure in the eyes of some, but we are quite happy (and proud) to be succeeding in our own style!

Robert Dewar, Ada Core Technologies

From: dewar@gnat.com (Robert Dewar)

Date: 23 Aug 2002 19:24:38 -0700

Subject: Re: Ada 95 for an ARM-based bare board?

Newsgroups: comp.lang.ada

[On support for very small teams. -- dc]

One interesting piece of experience here is that when we started we used to have prices for 1 or 2 seats. We abandoned that. Why? Because it was eating us alive. These tiny projects used far far more of our support services than larger projects. Some people complain that they have to pay the same for one developer as five, but in fact if we priced according to historical data, we should charge more for one person than five.

That's actually not so surprising, one person working on their own is indeed all on their own, and thus they need help wherever they can get it, whereas even a small team will be able to help one another over many simple problems.

It's just maybe that your imagination is not good enough, and that in fact you would find that a support contract would easily justify its cost. Many of our customers *are* very small developers, and find our support services extremely useful and very much worth the cost.

This is of course something you have to figure out for yourself. If the GNAT Pro product is not for you, have a look at offerings from other Ada vendors. You may well be able to find less expensive products that will meet your particular needs. There is a lively competition in the Ada market. It is not at all the case that GNAT is the only game in town :-)

Robert Dewar, Ada Core Technologies

From: dewar@gnat.com (Robert Dewar)

Date: 24 Aug 2002 06:01:48 -0700

Subject: Re: Software Economics

Newsgroups: comp.lang.ada

> There is no economic difference between the devastating effect on competition of a zero priced browser and a zero priced Ada compiler.

There is a no comparison between these situations. In one case, a company is making its product available at no cost to drive out a competitor by subsidizing the effort with other products (Internet Explorer). In the case of ACT, we absolutely definitely do NOT make our product (which is a fully supported Ada compiler, with guaranteed licensing suitable to the application area) available at no cost. In fact many of the threads on CLA [...] complain that GNAT Pro is too expensive!

> the second was done because of government support and ideology

[...] The reason that we make versions of GNAT freely available has nothing to do with government support (which disappeared 7 years ago) or with ideology. The reason we do this is because we think it is the best strategy for the continued health of Ada, and thus for the continued success of Ada Core Technologies. So it's simply good business sense!

When Ada first came out, there certainly was no freely available version of Ada, and several companies tried to market low cost products. They all failed. So that's a data point that seems completely contradictory to your thesis.

Second observation. There has always been a high quality free compiler for C, and these days for C++ as well. Do you think this has "devastated the market for these compiler technologies?" Of course not, to claim this would be absurd. The reason that we think it is critical to have a freely available high quality Ada compiler around is that it sparks interest in students, hobbyists, and individual engineers who want to experiment. If no one knows about Ada, then who is going to buy *any* Ada technology? - answer no one.

Yes, occasionally companies may use the public version of GNAT for mission critical projects. We find that a dubious decision, since the use of unsupported software with no assurance of correct licensing is rather risky. Few companies are willing to take this risk in practice. We don't see that as having a significant impact on the commercial market for Ada. [...]

[On "an inexpensive supported Ada compiler" for the mass market: -- dc]

My own analysis: Inexpensive mass market products are possible only if there is a mass market. I don't think there is such a market for Ada tools in the current climate. No one has ever succeeded with this approach yet, and I would not expect them to have succeeded.

The scale of things is just not right.

If you charge \$1000 for a product, then you need to sell several thousand of them a year to support a reasonable development effort, but that's still far too expensive for many hobbyists. Now if you

reduce the price to \$50, you have to sell ten's or hundred's of thousands of copies a year, and I just don't think the market begins to be there.

Furthermore, if you do try to use the mass market model, then you simply can't provide any kind of reasonable support, and our experience is that serious Ada projects really appreciate and need (and can afford) good support.

Robert Dewar, Ada Core Technologies

From: dewar@gnat.com (Robert Dewar)

Date: 27 Aug 2002 18:56:42 -0700

Subject: Re: Ada 95 for an ARM-based bare board?

Newsgroups: comp.lang.ada

> For me it seems you are afraid that low end solutions would eat up your revenue

Nope, that has nothing to do with it, it is just that this end of the market is not what we specialize in, and we believe our success (our sales are substantially up this year) is significantly due to the fact that we keep a narrow well defined focus.

> and that Ada would be used more in general purpose programs that don't focus on mission critical code

No, we don't believe that at all! If you believe it why not start a business aiming at these users, since it sounds like you believe it would be profitable. [...]

Robert Dewar, Ada Core Technologies

Mosemann's Keynote Speech at STC'2002

From: AdaIC--Editorial Webmaster

<webmaster@adaic.com>

Date: Wed, 05 Jun 2002 22:10:32 -0400

Subject: [AdaIC] Mosemann Speech and Interview [...]

To: AdaIC News List

<announce@adaic.com>

The AdaIC News has added two items to its website.

First, the speech by Lloyd K. Mosemann II, senior vice president of SAIC and retired Air Force deputy assistant secretary, that he gave at this year's Software Technology Conference, is now on the web at:

<http://www.adaic.org/news/mosemann.html>

[See also "CrossTalk - Keynote STC Speech Lloyd K. Mosemann" in AUJ 23-3 (Sep 2002), p.151. -- dc]

Second, we interviewed Mr. Mosemann to see how his views on Ada have weathered his entering the commercial sector:

<http://www.adaic.org/news/mose-int.html>

[Ann Brandon <abrandon@sover.net> added: -- dc]

He had some fine things to say about Ada, and also about SPARKAda and Rational. Though he retired as Air Force deputy assistant secretary a while ago and is now

a senior VP at SAIC, his opinions about Ada and software development process haven't changed.

URL:

<http://www.adaic.org/news/mosemann.html>

Subject: AdaIC Article - Lloyd K.

Mosemann: You Get What You Pay For

Did We Lose Our Religion?

Lloyd K. Mosemann II, Software Technology Conference, April 29, 2002

[...] Back in 1990 I declared that the 1980s were a lost decade from the perspective of software development progress. The question I posed was: "Will there be a breakthrough in the 1990's?" I went on to say: "It won't happen automatically; people are too satisfied with unsatisfactory ways. [...]"

In 1994, I closed this annual Software Technology Conference with the observation that the underlying need within the defense community is for predictability. "[...] What our senior managers and DoD leaders want most from us, is to deliver on our promises. They want systems that are on-time, within budget, that satisfy user requirements, and are reliable."

The Question I should like to briefly explore this morning is: "Where are we today, and where will we be tomorrow?" Did we lose our religion?

Why do I use the metaphor of "religion"? Primarily because "religion" is the traditional example of "faith-based" behavior - that is, behavior that is based on a belief-system rather than on externally imposed rules such as the "law of gravity" or "she that has the gold, rules". [...]

Sadly, I still see the world of software being governed by religious-like belief systems rather than objective appraisal. When I left the Pentagon six years ago I described some of what was happening as "bumper sticker" management, and the situation has not changed for the better. I sometimes have the feeling that the "blind are leading the blind", or as shown here, the leadership is blissfully ignorant of the direction in which they are headed. [...]

Many people used to tell me that DOD needed to get on the .com bandwagon - 'cause those folks develop and deliver software fast. It is true that the Internet and the availability of web browsers has fundamentally changed the environment within which even mission critical software is being developed. But, instead of adapting proven software methods, the software research community has all but dropped its concerns with formal methods, peer reviews, cleanroom processes and other reliability techniques, including the language known as Ada that was designed to promote reliability. [...], invention of new user interfaces, of new distributed

computing architectures, of new (more flexible and less reliable) programming languages have been given top priority. The goals of reliable performance and predictable development costs have been largely ignored. [...]

Let me talk briefly about some subjects that have been notably missing from the plenary sessions of this Conference over the past few years: Software Engineering, Product Line Development, Formal Methods Programming, and Predictability. [...]

The fundamental ingredient in a "software engineering" approach is the design of a robust software architecture. By architecture [I refer] to the inherent design of the software itself - the identity of modules and their relationships, including in particular the infrastructure, control and data interfaces that permit software components to operate as a system. [...] All too often the DOD excuse for not requiring an architectural evaluation is that "requirements change rapidly -- can't afford to be locked into a specific architecture". Wrong - that is the very reason that attention should be given to architecture, so that changes to requirements can be accommodated easily. [...]

Interestingly, when one thinks about "best commercial practice", there are two very different worlds out there. There is the Government contractor world; and, there is the real commercial world - banks, insurance companies, UPS and FedEx, Eckerd Drug, and DisneyWorld. These companies develop their own software using the best available software tools such as Rational's software development environment. They don't pick the cheapest tools. They don't rely on COTS or outside software developers - their software is their business, and they consider that it provides them a competitive advantage, and they want to control it, and they use the best tools available regardless of cost. [...]

I work for one, but let me tell you that, although government contractors are commercial organizations, they do not have an identifiable "best commercial practice". They basically will provide what the Government asks for. [...] Too often what the Govt has been asking for is COTS. How many failures of COTS-based acquisitions have there been over the past decade? Too many!

[..] "Best commercial practice" is what real commercial companies are doing. They have in-house software expertise, they use a robust software development environment, and they base their software development on a sound software architecture. It is no secret/no surprise that Rational and their competitors have a growing market in the commercial world and a shrinking market in the government world. I'm not suggesting that the

Government can or should develop software in-house. I am suggesting that the Government needs enough in-house software expertise to know what it is buying. It is still true that you get what you pay for", and that "apples and oranges" are not the same!

Watts Humphrey recently published a new book entitled "Winning Software - An Executive Strategy." Not surprisingly, the book is directed primarily at executives of commercial companies. [...] Frankly, its message is pretty simple: "Software projects rarely fail for technical reasons; invariably, the problem is poor management." He poses two interesting questions and buttresses them with numerous examples: "Why do competent software professionals agree to dates when they have no idea how to meet them? Why do executives accept schedule commitments when the engineers offer no evidence that they can meet these commitments?" He asserts that Management's undisciplined approach to commitments contributes to every one of the five most common causes of project failure: unrealistic schedules, inappropriate staffing, changing requirements, poor quality work, believing in magic.

I don't have time this morning to amplify these critical management issues - buy and read the book, and make sure that your bosses read it.

What is Formal Methods Programming? Basically, it is all of the above rolled together: sound management, established engineering processes, robust software development environment, model based architecture, and a reliable programming language. I was thrilled to see, in the March edition of CrossTalk an article by Peter Amey of Praxis Critical Systems, in which he stated: "There is now compelling evidence that development methods that focus on bug prevention rather than bug detection can both raise quality and save time and money." He then goes on to say that a key ingredient is the use of unambiguous programming languages that allow rigorous analysis very early in the development process. As many of you know, I was an early and vocal advocate of Ada, primarily because, unlike other languages, it enforces basic software engineering practice. In that article Peter describes the use of a subset of Ada known as SPARK. He says, "The exact semantics of SPARK require software writers to think carefully and express themselves clearly; any lack of precision is ruthlessly exposed by its support tool, the SPARK Examiner." He indicates that there were significant savings from using SPARK in a critical avionics program, including most notably that formal FAA test costs were reduced by 80%. Unfortunately, this is an isolated DOD example - the same rigor and discipline of formal methods programming should

apply for all major software intensive system developments.

Finally, let me say a word about Predictability. [...] Question is - how can we make [...] the PEOs and PMs who are charged with delivering the capabilities needed [...] smart enough to know that you can't just buy software as a product off the showroom floor. There must be understanding of the software engineering paradigm that produces software. In fact, more than this, to be assured of getting software that works on a predictable schedule and at predictable cost requires that someone in government be smart enough to enunciate the basic processes that will be employed by the contractor to produce. This is important because, otherwise, competitors will bid an unrealistically low price and unrealistically fast schedule, and be awarded the contract. To perform at the low cost means no robust software development environment, no time and effort devoted to creating a valid software architecture, and probably means cheap people. If the government wants to "get software right", sufficient process guidance must be given to assure that the contractors all bid essentially the same level of capability. I believe that Government should be explicit about the need for architecture, a robust software development environment, perhaps even the requirement for a language like SPARK, but, as a minimum, it needs to specify that the performing organization must be at least CMM Level 3. [show slides comparing before and after - In short, at Level 1 virtually all projects have cost and schedule overruns, whereas at Level 3 virtually all projects are on target. As regards Defect Rates and \$ per Source Line of Code, there is substantial improvement on the order of 20-50%.]

It really is true that you get what you pay for. If you want it cheap, you'll get it cheap - but it may not work in the manner envisioned, if it will work at all. [...]

URL: <http://www.adaic.org/news/mose-int.html>

Subject: *AdaIC Article - Mosemann Interview*

Ada vs. 'Bumper sticker management': Lloyd K. Mosemann II on Excellence in Software Development

By Ann S. Brandon

"Bumper sticker management" inflates the cost of software and lowers its quality, according to Lloyd K. Mosemann II, a senior vice president of SAIC and retired deputy assistant secretary of the Air Force. In a recent telephone interview with the AdaIC News, Mosemann defined "bumper sticker management" as following rather than leading the consumer. [...]

"Most senior people have no idea what Ada is vs. why Ada should be chosen over

anything else." He spoke from a personal viewpoint and not as an SAIC representative.

Mosemann led software management for long enough to stumble again and again across the many obstacles to change. After 25 years as deputy assistant secretary to the Air Force, he retired as head of communications, computers and support systems. "In that capacity I oversaw a leadership role with Ada," he said. Even as an early proponent of Ada, he admitted that he "had a tough time selling it to senior management once C++ came on the scene. People heard that [C++] was new and better. They didn't consider its difficulty with multiple tasking and multiple inheritance and how that affected software's lifecycle. Those are some of the things that it made attractive but also difficult to maintain."

Software managers still depend on Ada for developing robust and portable software. Today, Mosemann cites Praxis SPARKAda as an example of using Ada for its reliability. Otherwise, most companies go for "what's new and what's popular." Software managers often ignore Ada because of a sense that "not enough people are experienced programmers. Also, it's not sexy so people didn't want to use it." He "supported Ada for the same reasons" that he is a "big supporter of process improvement." [...]

[...] In the keynote speech at the most recent STC, held in April 2002 in Salt Lake City, Mosemann called for software managers to insist on "formal methods programming," which he defined as "sound management, established engineering processes, robust software development environment, model based architecture, and a reliable programming language."

In the AdaIC News interview, he referred to the foundation for successful software development as a "three-legged stool: a mature process; robust tools like Rational's; and a language that's reliable, like Ada.

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

- January 06-09 Software Technology Track of the 36th **Hawaii International Conference on System Sciences (HICSS-36)** Big Island of Hawaii, USA. Includes mini-tracks on: Experimental Software Engineering; Domain-Specific Languages; Distributed Object and Component-based Software Systems (Design Patterns for Distributed Systems, Middleware, Programming Languages and Environments for Distributed Object and Component Systems, ...); etc.
- January 08-11 **Workshop on Techniques for Trusted Components**, Prato (near Florence), Italy.
- © January 13-15 8th IEEE Computer Society's **International Workshop on Object-oriented Real-Time Dependable Systems (WORDS'2003)** Guadalajara, Mexico.
- January 15-17 30th Annual ACM SIGPLAN-SIGACT **Symposium on Principles of Programming Languages (POPL'2003)** New Orleans, Louisiana, USA. Topics include: design, definition, analysis, and implementation of programming languages, programming systems, and programming abstractions.
- January 18 10th International **Workshop on Foundations of Object-Oriented Languages (FOOL 10)** New Orleans, Louisiana, USA. Following POPL'03. Topics include: the general area of foundations of object-oriented languages; language semantics, type systems, program analysis and verification, concurrent and distributed languages, etc.
- January 18 ACM SIGPLAN **Workshop on Types in Language Design and Implementation (TLDI'2003)** New Orleans, Louisiana, USA. Co-located with POPL'03 Topics include: Typed intermediate languages and type-directed compilation; Type-based language support for safety and security; Type safety and security of system programming languages; Type-based program analysis, transformation, and optimization; Types for security protocols, concurrency, and distributed computing; Type-based specifications of data structures and program invariants; etc.
- February 03-05 9th **International Conference on Languages and Models with Objects (LMO'2003)** Vannes, France. Topics include (in French): Programmation par objet et programmation par composant; Programmation par objet et modélisation par objets; etc.
- February 04-07 **Australasian Computer Science Conference (ACSC'2003)** Adelaide, South Australia. Topics include: Compilers, Concurrency, Distributed Systems, Embedded Systems, Fault Tolerance, Formal Methods, Object-Oriented Systems, Programming Languages, Real-time Systems, Reliability, Software Engineering, Trusted Systems, etc.
- © February 05-07 11th Euromicro **Conference on Parallel Distributed and Network based Processing (PDP'2003)** Genoa, Italy. Topics include: Distributed Systems; Parallel Computer Systems; Models and Tools for Parallel Programming Environments; Advanced Applications (numerical applications with multi-level parallelism, real time distributed applications, distributed business applications, ...); Languages, Compilers and Runtime Support Systems (task and data parallel languages, object-oriented languages, scheduling and load balancing, task and object migration, ...), etc. Special sessions on: Advanced Tools for Parallel and Distributed Programming; Parallel Realtime Systems; Memory Hierarchies; etc.
- February 10-12 2nd **International Conference on Commercial Off-The-Shelf (COTS)-Based Software Systems (ICCBSS'2003)** Ottawa, Canada. Theme: "Multiple Paths, Multiple Solutions".
- February 17 20th anniversary of the first Ada standard (ANSI/MIL-STD-1815A), dated 1983 Feb 17.

- February 24-27 15th Annual **Software Engineering Process Group Conference (SEPG'2003)** Boston, Massachusetts, USA. Theme: "Assuring Stability in a Global Enterprise" Deadline for early registration: January 22, 2003.
- March 04-05 3rd **Workshop on Aspect-Oriented Software Development (AOSD'2003)** Essen, Germany.
- March 05-06 2nd **International Workshop on Unanticipated Software Evolution (USE'2003)** Warsaw, Poland. Topics include: Formal methods, language concepts and implementation techniques for USE; USE support in programming languages, component models and related infrastructures; etc.
- March 09-12 2003 ACM **Symposium on Applied Computing (SAC'03)** Melbourne, Florida, USA. Includes tracks on Embedded Systems: Applications, Solutions, and Techniques; Software Engineering: Applications, Practices and Tools; Programming Languages and Object Technologies; Parallel and Distributed Systems and Networking; etc.
- March 17-21 Joint ESA - 3rd **International SPICE Conference on Process Assessment and Improvement (SPICE'2003)** ESTEC, Noordwijk, The Netherlands.
- March 17-21 2nd **International Conference on Aspect-Oriented Software Development (AOSD'2003)** Boston, USA. Topics include: Applications; Software development methods; Reverse engineering and refactoring; Tools; Programming languages and implementation; Distributed systems; Composition, integration and evolution; etc.
- March 20-22 16th **Conference on Software Engineering Education and Training (CSEET'2003)** Madrid, Spain. Theme: "Software Engineering in Industry and University for the 21st Century"
- March 26-28 7th **European Conference on Software Maintenance and Reengineering (CSMR'2003)** Benevento, Italy. Topics include: experience reports, enabling technologies, etc.
- April 02-04 4th **International Conference on Software Testing (ICSTEST'2003)** Cologne, Germany.
- April 05-13 **European Joint Conferences on Theory and Practice of Software (ETAPS'2003)** Warsaw, Poland. Event includes: conferences from 7 to 11 April 2003, affiliated workshops on 5-6 and 12-13 April, 2003. Includes a.o. the following events:
- April 06 ETAPS2003 - 3rd **Workshop on Language Descriptions, Tools and Applications (LDTA'2003)**. Topics include: Program analysis, transformation, generation; Automatic generation of language processing tools; etc.
- April 06 ETAPS2003 - **Workshop on Software Composition (SC'2003)**
- April 07-11 ETAPS2003 - 12th **International Conference on Compiler Construction (CC'2003)**. Topics include: compilation and interpretation techniques; run-time issues; language constructs and their implementation; modularization constructs and techniques for separate compilation; tools for compiler construction or language support, including debuggers, profilers, refactoring tools, etc.
- April 07-11 ETAPS2003 - 12th **European Symposium on Programming (ESOP'2003)**. Topics include: design of programming languages and calculi; techniques, methods and tools for their implementation; exploitation of programming styles within different programming paradigms; automatic and manual methods for reasoning about programs; etc.
- April 07-11 ETAPS2003 - **Fundamental Approaches to Software Engineering (FASE'2003)**. Topics include: experience reports on best practices with component models and specifications, development tools, modelling environments, and software development kits; etc.
- April 07-11 ETAPS2003 - 9th **International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2003)**
- April 07-11 10th IEEE **Symposium and Workshops on the Engineering of Computer Based Systems (ECBS'2003)** Huntsville, Alabama, USA. Topics include: Component-Based Design and Reuse; Middleware for Embedded Systems; Applied Formal Methods; Education and Training;

Embedded Systems; Evolution, Reengineering and Legacy Systems; Reliability, Dependability, Safety and Security; Verification and Validation Standards; etc.

- April 08-10 7th **International Conference on Evaluation and Assessment in Software Engineering (EASE'2003)** Staffordshire, UK. Deadline for submissions: February 3, 2003 (experience papers), April 2, 2003 (posters).
- April 14-16 10th Annual **European Concurrent Engineering Conference (ECEC'2003)** Plymouth, UK.
- ☉ April 22-26 **International Parallel and Distributed Processing Symposium (IPDPS'2003)** Nice, France. Topics include: 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 22-26 IPDPS2003 - 8th **International Workshop on Formal Methods for Parallel Programming: Theory and Applications (FMPPTA'2003)**
- April 22-26 IPDPS2003 - 4th **Workshop on Parallel and Distributed Scientific and Engineering Computing with Applications (PDSECA'2003)**
- ☉ May 03-10 25th **International Conference on Software Engineering (ICSE'2003)** Portland, Oregon, USA. Deadline for submissions: January 20, 2003 (demos, posters, doctoral symposium).
- ☉ May 14-16 6th IEEE **International Symposium on Object-oriented Real-time distributed Computing (ISORC'2003)** Hakodate, Hokkaido, Japan.
- May 19-22 23rd **International Conference on Distributed Computing Systems (ICDCS'2003)** Providence, Rhode Island, USA. Topics include: Middleware; Fault Tolerant and Dependable Systems; Real-time and Embedded Systems; Software Engineering and Formal Methods; etc.
- May 26-29 4th **International Conference on eXtreme Programming and Agile Processes in Software Engineering (XP'2003)** Genova, Italy. Deadline for submissions: March 1, 2003 (PhD symposium). Deadline for early registration: April 1, 2003
- ☉ May 27-30 9th IEEE **Real-Time and Embedded Technology and Applications Symposium (RTAS'2003)** Toronto, Canada. Topics include: Real-time Linux and applications; DRE middleware, e.g. Real-time CORBA; Secure real-time systems; Real-time software component models; QoS-aware application design and patterns; Embedded control applications; etc. Deadline for paper submissions: January 15, 2003.
- May 28-30 7th **Brazilian Symposium on Programming Languages (SBLP'2003)** Ouro Preto, MG, Brazil. Topics include: Programming language design and implementation; Formal semantics of programming languages; Programming languages for mobile, WWW, and network computing; Teaching programming languages; etc. Deadline for submissions: February 14, 2003.
- June 09-14 28th Annual **USENIX Technical Conference (USENIX'2003)** San Antonio, Texas. Topics include: applications, architecture, implementation, and performance of modern computing systems; Reliability and QoS; Interoperability of heterogeneous systems; special FREENIX track on freely redistributed technology; etc.
- ◆ June 16-20 8th **International Conference on Reliable Software Technologies - Ada-Europe'2003**, Toulouse, France. Sponsored by Ada-Europe, in cooperation with ACM SIGAda (approval pending). Topics include: Management of Software Development and Maintenance; Software Quality; Software Development Methods and Techniques; Software Architectures; Tools; Kinds of Systems; Applications; Ada Language and Tools; Ada Experience Reports; Education and Training; Case Studies and Experiments; and a special session on Avionics and Space, including the use of Ada in this realm.
- June 16-20 ACM/IFIP **International Middleware Conference (Middleware'2003)** Rio de Janeiro, Brazil. Theme: "Information Systems for a Connected Society". Topics include: Distributed real-time and embedded (DRE) middleware platforms; Reliable and fault-tolerant middleware platforms; Formal Methods applied to middleware; Novel paradigms, APIs, and languages for distributed systems;

etc. Deadline for submissions: February 1, 2003 (tutorial proposals), February 20, 2003 (work in progress papers, workshop papers, posters).

- June 18-20 **International Conference on Application of Concurrency to System Design (ACSD'2003)** Guimaraes, Portugal. Topics include: Real-time and hybrid systems; Case studies of concurrent systems design and verification; Presentation of software tools supporting the above topics; etc.
- June 22-25 **International Conference on Dependable Systems and Networks (DSN'2003)** San Francisco, CA, USA. Topics include: Architectures for Dependable Computer Systems; Fault Tolerance in Transaction Processing; Fault Tolerance in Distributed & Real-Time Systems; Safety-Critical Systems; Software Testing, Validation, and Verification; Software Reliability; etc. Deadline for submissions: January 13, 2003 (workshops, tutorials), April 21, 2003 (student forum, fast abstracts).
- ☉ June 23-26 **2003 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'2003)** Las Vegas, Nevada, USA. Topics include: Parallel/Distributed applications; Reliability & fault-tolerance; Real-time & embedded systems; Object Oriented Technology & related issues; Software tools & environments for parallel & distributed platforms; Education: parallel & distributed processing in computer science curriculum; Recent history (last decade) of parallel/distributed processing & what to expect in the next decade: New Horizons; etc. Deadline for submissions: February 17, 2003 (draft papers).
- July 13-16 **22nd Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC'2003)** Boston, Massachusetts, USA. Topics include: design, specification, implementation, application and theory of distributed systems. Deadline for submissions: January 31, 2003.
- ☉ July 21-25 **17th European Conference on Object-Oriented Programming (ECOOP'2003)** Darmstadt, Germany. Deadline for submissions: March 1, 2003 (doctoral symposium), April 1, 2003 (demonstrations, posters).
- ☉ August 13-15 **16th International Conference on Parallel and Distributed Computing Systems (PDCS'2003)** Reno, Nevada, USA. Topics include: all areas of Parallel and Distributed Computing Systems, their modeling and simulation, design, use and performance, etc. Deadline for submissions: February 24, 2003.
- ☉ August 25-27 **6th Joint Modular Languages Conference (JMLC'2003)** Klagenfurt, Austria. Topics include: concepts of well structured modular software; teaching good design and programming style; construction of large and distributed software systems; etc. Includes on 25-26 August, some common tutorials, keynote talks and a common panel with EuroPar'2003. Deadline for submissions: February 9, 2003. Deadline for early registration: July 15, 2003.
- ☉ August 26-29 **9th International Conference on Parallel and Distributed Computing (Euro-Par'2003)** Klagenfurt, Austria. Topics include: all aspects of parallel and distributed computing. Includes on 25-26 August, some common tutorials, keynote talks and a common panel with JMLC'2003. Deadline for submissions: February 9, 2003.
- September 03-05 **29th EUROMICRO Conference (EUROMICRO'2003)** Antalya, Turkey. Deadline for submissions: March 10, 2003 (papers)
- September 08-12 **11th IEEE International Requirements Engineering Conference (RE'03)** Monterey Bay, California, USA. Deadline for submissions: January 31, 2003 (papers), March 1, 2003 (industry track), March 29, 2003 (workshops, tutorials, doctoral symposium), April 26, 2003 (posters, research demos)
- September 08-14 **12th International Formal Methods Europe Symposium (FME'2003)** Pisa, Italy. Topics include: concerns and risks for potential adopters of formal methods; cost-benefit analysis; reports on practical use and case studies (reporting positive or negative experiences); tool support and software engineering; etc. Deadline for submissions: March 7, 2003 (papers, workshops, tutorials), May 9, 2003 (tool exhibition).
- ◆ September 15-19 **12th International Real-Time Ada Workshop (IRTAW-2003)** Viana do Castelo, Portugal. Deadline for position paper submissions: June 1, 2003.

- September 25-26 3rd **International Conference on Quality Software** (QSIC'2003) Beijing, China. Topics include: Debugging; Economics of software quality and testing; Formal methods; Quality evaluation of software products and components; Reliability; Software quality education; Static and dynamic analysis; Testability; Testing of object-oriented systems; Testing of concurrent and real-time systems; Testing strategies, tools, processes, and standards; Tool support for improving software quality; Validation and verification; Application areas such as component-based systems, distributed systems, embedded systems, enterprise applications, ...; etc. Deadline for submissions: March 10, 2003.
- September 30-10/01 IEEE **International Symposium on Empirical Software Engineering** (ISESE'2003) Frascati, Italy. Topics include: strengths and weaknesses of technology in use and new technologies, etc. Deadline for submissions: March 29, 2003.
- October 06-09 13th **International Conference on Software Quality** (ICSQ'2003) Dallas, Texas, USA. Deadline for submissions: January 31, 2003 (abstracts, tutorials), March 28, 2003 (panels).
- October 20-24 6th **International Conference on UML - Modeling Languages and Applications** (UML'2003) San Francisco, USA. Topics include: tool support for any aspect of modeling or model use; models in the development and maintenance process; domain-specific and concern-oriented modeling; etc. Deadline for submissions: April 8, 2003 (abstracts), April 15, 2003 (papers), April 21, 2003 (workshops), June 29, 2003 (tutorials).
- October 28-31 IEEE **Symposia on Human-Centric Computing Languages and Environments** (HCC'03) Auckland, New Zealand. Topics include: tools that enable humans, using textual languages, visual languages or any other appropriate technologies, to accomplish their tasks more effectively; etc. Deadline for submissions: February 1, 2003 (workshops), March 15, 2003 (papers).
- December 10 Birthday of Lady Ada Lovelace, born in 1815. Happy Programmers' Day!

2004

- March 29 – April 02 **European Joint Conferences on Theory and Practice of Software** (ETAPS'2003) Barcelona, Spain. Event includes: conferences from 29 March to 2 April, 2004, affiliated workshops on 27-28 March and 3-4 April, 2004.

8th International Conference on Reliable Software Technologies



Ada-Europe 2003



June 16-20, 2003

Toulouse, France

Call for participation

A conference not to be missed !

The full conference will comprise a three-day technical program from Tuesday to Thursday. With 45 high quality papers submitted, the Program Committee is setting up a technically rewarding and enjoyable program. Stay tuned ! Final program to be announced by the end of January.

An exhibition will be held in parallel with the conference, with vendors presenting the latest products and services related to Ada and reliable software technologies.

Parallel workshops and tutorials on Monday and Friday will offer a unique opportunity to learn about new advances and improve your technical knowledge.


A most enjoyable venue

Toulouse is located in the sunny southern part of France. It is a city where high technology (Ariane, Airbus, CNES...) meets history, with lots of landmarks dating back to the middle-ages and beyond. Take a couple of extra days, and enjoy the superb countryside and the world famous gastronomy!

Links :

- Conference information :
<http://www.ada-europe.org/conference2003.html>
- Toulouse Tourist Office :
<http://www.ot-toulouse.fr/English>



	<p style="text-align: center;">12th International Real-Time Ada Workshop IRTAW-12 15-19 September 2003 Pousada Monte de Sta. Luzia, Viana do Castelo Portugal</p>
---	---

First Announcement

Over the last 15 years, the series of the **International Workshop on Real-Time Ada Issues** have provided focus for identifying issues with Ada 83 and 95; for proposing solutions to those problems; and for evaluating proposed changes to the language standard.

Since the standardization of Ada 95, the **IRTAW** series have assisted in the review of the real-time related chapters of the *Guide for the Use of the Ada Programming Language for High Integrity Systems* (ISO/IEC TR 15942:2000) and have developed and promoted the Ravenscar Tasking Profile. With the advent of Java and the development of Real Time Specification for Java, the Workshop has begun to consider the integration of embedded Ada and Java systems, and their interoperability.

In keeping with this tradition, the goals of **IRTAW-12** will be to:

- Examine the shape and the status of the language amendments proposed or promoted by previous **IRTAW** editions, with respect to the ongoing Ada language revision process, as well as the demand for further enhancements
- Consider the lessons learned from industrial experience with using Ada 95 in general and the Ravenscar Profile in particular in actual real-time projects
- Examine and develop paradigms for using Ada 95 for real-time distributed systems, including issues of robustness as well as of hard, flexible and application-defined scheduling
- Consider the impact of the OOP paradigm on multi-threaded, possibly distributed, real-time systems
- Review the status and contents of the *Guide for the Use of the Ada Ravenscar Profile in High Integrity Systems* (to become an ISO/IEC TR) and examine the issues of certifying software systems developed using Ada 95
- Examine the status of and the progress with the Real-Time Specification for Java and consider user experience with the reference implementation and with issues of interoperability with Ada in embedded real-time systems.

Participation at **IRTAW-12** is by invitation following the submission of a position paper addressing one or more of the above topics. Position papers should not exceed six pages. All accepted papers will appear, in their final form, in the Workshop Proceedings, which will be published as a special issue of Ada Letters (ACM Press).

Submit position papers, in PDF, to the Program Chair by e-mail at: tullio.vardanega@math.unipd.it

Program Committee:

Ben Brosgol, Alan Burns, Brian Dobbing, Michael Gonzalez Harbour, Mike Kamrad, Stephen Michell, Luis Miguel Pinho (Local Chair), Juan Antonio de la Puente, George Romanski, Joyce Tokar, Tullio Vardanega (Program Chair), Andy Wellings.

Important Dates

Receipt of Position Paper: 1 June 2003
 Notification of Acceptance: 30 June 2003

Final Copy of Paper: 1 November 2003
 Workshop Date: 15-19 September 2003



<http://www.hurray.issep.ipp.pt/irtaw2003>



An Invitation to Ada 2005

Pascal Leroy

Chairman, Ada Rapporteur Group

Rational Software, 1 place Charles de Gaulle, 78067 St Quentin en Yvelines, France;

email: pleroy@rational.com

Abstract

Starting in 2000, the ISO technical group in charge of maintaining the Ada language has started looking into possible changes for the next revision of the standard, around 2005. Based on the input from the Ada community, it was felt that the revision was a great opportunity for further enhancing Ada by integrating new programming practices, e.g. in the OOP area; by providing new capabilities for embedded and high-reliability applications; and by remedying annoyances encountered during many years of usage of Ada 95. This led to the decision to make a substantive revision rather than a minor one. This article outlines the standardization process and schedule, and presents a number of key improvements that are currently under consideration for inclusion in Ada 2005.

1 Introduction

Every ISO standard is reviewed every five years to determine if it should be confirmed (kept as is), revised, or withdrawn. The international standard for Ada [1], known as ISO/IEC 8652 in ISO jargon, was published in 1995. A Technical Corrigendum, ISO/IEC 8652 Corr. 1, was published at the end of the first review period, in 2001. This document corrected a variety of minor errors or oversights in the original language definition, with the purpose of improving the safety and portability of programs. However, it didn't add significant new capabilities.

With 7 years' experience with Ada 95, the ISO working group in charge of maintaining the language (known as JTC1/SC22/WG9) has come to the conclusion that more extensive changes were needed at the end of the second review period, which ends in 2005. As a result, it was decided to develop an Amendment to integrate into the Ada language the result of more than 10 years of research and practice in programming languages. (In ISO terms, an Amendment is a much larger change than a Technical Corrigendum.)

WG9 has asked its technical committee, the Ada Rapporteur Group (ARG) to prepare proposals for additions of new capabilities to the language. Such capabilities could take the form of new core language features (including new syntax), new predefined units, new specialized needs annexes, or secondary standards, as appropriate. The changes may range from relatively minor to quite substantial.

2 Revision guidelines

As part of the instructions it gave to the ARG, WG9 has indicated that the main purpose of the Amendment is to address identified problems in Ada that are interfering with the usage or adoption of the language, especially in those application areas where it has traditionally had a strong presence: high-reliability, long-lived real-time and embedded application, and very large complex systems.

In particular, the ARG was requested to pay particular attention to two categories of improvements:

- Improvements that will maintain or improve Ada's advantages, especially in those user domains where safety and criticality are prime concerns;
- Improvements that will remedy shortcomings of Ada.

Improvements that fall into the first category include: new real-time features, such as the Ravenscar profile, and features that improve static error detection. Improvements that fall into the second category include a solution to the problem of mutually dependent types across packages, and support of Java-like interfaces.

In selecting features for inclusion in the Amendment, it is very important to avoid gratuitous changes that could impose unnecessary disruption on the existing Ada community. Therefore, the ARG was asked to consider the following factors when evaluating proposals:

- Implementability: can the proposed feature be implemented at reasonable cost?
- Need: does the proposed feature fulfill an actual user need?
- Language stability: would the proposed feature appear disturbing to current users?
- Competition and popularity: does the proposed feature help improve the perception of Ada, and make it more competitive with other languages?
- Interoperability: does the proposed feature ease problems of interfacing with other languages and systems?
- Language consistency: is the provision of the feature syntactically and semantically consistent with the language's current structure and design philosophy?

In order to produce a technically superior result, it was also deemed acceptable to compromise strict backward compatibility when the impact on users is judged to be

acceptable. It must be stressed, though, that the ARG has so far been extremely cautious on the topic of incompatibilities: incompatibilities that can be detected statically (e.g., because they cause compilation errors) might be acceptable if they are necessary to introduce a new feature that has numerous benefits; on the other hand, incompatibilities that would silently change the effect of a program are out of the question at this late stage.

In order to provide input to the ARG, the Ada community is invited to submit proposals for evolutions of the language to <mailto:ada-comment@ada-auth.org>. All proposals will be looked into, but they will be judged by the criteria listed above. In particular, proposals that demonstrably meet users needs and that are worked out in sufficient detail are more likely to be included in the Amendment. As mentioned in the schedule below, the hard deadline for receiving proposals from the general public is September 2003. Earlier submissions are very much appreciated, as the ARG will have more time to study, analyze, and revise them.

In organizing its work, the ARG creates study documents named Ada Issues (AI) that cover all the detailed implications of a change. The Ada Issues may be consulted on-line at <http://www.ada-auth.org/ais.html>. The reader should keep in mind though that these are working documents that are constantly evolving, and that there is no guarantee that any particular AI will be included in the final Amendment.

3 Revision schedule

At the time of this writing, WG9 targets the following schedule for the development of the Amendment:

- September 2003: receipt of the final proposals from groups other than WG9 or delegated bodies.
- December 2003: receipt of the final proposals from WG9 or delegated bodies.
- June 2004: approval of the scope of the Amendment, perhaps by approving individual AIs, perhaps by approving the entire Amendment document.
- Late 2004: informal circulation of the draft Amendment document, receipt of comments, and preparation of final text.
- Spring 2005: completion of proposed text of the Amendment.
- Mid 2005: WG9 email ballot.
- 3Q 2005: SC22 ballot (SC22 is the parent body of WG9, and is in charge of the standardization of all programming languages).
- Late 2005: JTC1 ballot, final approval.

The elaboration of the Amendment being a software-related project, and a very complex one to boot, it would not be too surprising if the above schedule slipped by a couple of months. However, the work will essentially be schedule-driven, and the ARG will effectively stop developing new

features at the beginning of 2004. The following two years will be spent writing and refining the Amendment document to ensure that its quality is on a par with that of the original Reference Manual.

4 Proposed new features

The rest of this article gives a technical overview of the most important AIs that are currently being considered for inclusion in the Amendment. Again, this is just a snapshot of work currently in progress, and there is no guarantee that all of these changes will be included in the final Amendment. And even if they are, they might change significantly before their final incarnation.

We first present the new features related to real-time, safety and criticality issue; then those that pertain to the object-oriented programming paradigm; and finally a number of features that remedy shortcomings of the current language. For each feature, the number of the relevant AI is given in the section header.

4.1 Ravenscar profile for high-integrity systems (AI 249 and AI 305)

The Ravenscar profile [2] is a subset of Ada (also known as a tasking profile) which was initially defined by the 8th International Real-Time Ada Workshop (IRTAW), and later refined by the 9th and 10th IRTAW. This profile is intended to be used in high-integrity systems, and makes it possible to use a run-time kernel of reduced size and complexity, which can be implemented reliably. It also ensures that the scheduling properties of programs may be analyzed formally.

The Ravenscar profile has been implemented by a number of vendors and has become a *de facto* standard. The ARG plans to include it in the Amendment. However, in looking at the details of the restrictions imposed by Ravenscar, it was quickly realized that many of them would be generally useful to Ada users, quite independently of real-time or high-integrity issues. So the ARG decided to proceed in two phases: first, it defined a number of restrictions and one new pragma; then it specified the Ravenscar profile in terms of these restrictions and pragma. This approach has three advantages:

- Current users of the Ravenscar profile are virtually unaffected.
- Users who need to restrict the usage of some Ada constructs without abiding by the full set of restrictions imposed by Ravenscar can take advantage of the newly defined restrictions.
- Implementers may define additional profiles that cater to the needs of particular users communities.

AI 305 defines new restrictions corresponding to all the Ada constructs that are outlawed by the Ravenscar profile. To take only two examples, restrictions `No_Calendar` and `No_Delay_Relative` correspond to the fact that Ravenscar forbids usage of package `Ada.Calendar` and of relative delay statements. AI 305 also defines a new configuration pragma, `Detect_Blocking`, which indicates that the runtime

system is required to detect the execution of potentially blocking operations in protected operations, and to raise `Program_Error` when this situation arises.

AI 249 then defines a new configuration pragma, `Profile`, which specifies as its argument either the predefined identifier `Ravenscar` or some implementation-defined identifier (this is to allow compiler vendors to support additional profiles). The `Ravenscar` profile is defined to be equivalent to a list of 19 restrictions, to the selection of policies `Ceiling_Locking` and `FIFO_Within_Priorities`, and to the detection of blocking operations effected by pragma `Detect_Blocking`.

4.2 Execution-time clocks (AI 307)

The 11th IRTAW identified as one of the most pressing needs of the real-time community (second only to the standardization of the `Ravenscar` profile) the addition of a capability to measure execution time.

AI 307 proposes to introduce a new predefined package, `Ada.Real_Time.Execution_Time`, exporting a private type named `CPU_Time`, which represents the processor time consumed by a task. Arithmetic and conversion operations similar to those in `Ada.Real_Time` are provided for this type. In addition, it is possible to query the CPU time consumed by a task by calling the function `CPU_Clock`, passing the `Task_ID` for that task. This package also exports a protected type, `Timer`, which may be used to wait until some task has consumed a predetermined amount of CPU time.

`Ada.Real_Time.Execution_Time` makes it possible to implement CPU time budgeting, as shown by the following example, where a periodic task limits its own execution time to some predefined amount called WCET (worst-case execution time). If an execution time overrun is detected, the task aborts the remainder of its execution, until the next period:

```
with Ada.Task_Identification;
with Ada.Real_Time.Execution_Time;
...
use Ada.Real_Time;
use type Ada.Real_Time.Time;
...
task body Periodic_Stopped is
  The_Timer : Execution_Time.Timer;
  Next_Start : Real_Time.Time:=Real_Time.Clock;
  WCET : constant Duration:=1.0E-3;
  Period : constant Duration:=1.0E-2;
begin
  loop
    The_Timer.Arm
      (To_Time_Span (WCET),
       Ada.Task_Identification.Current_Task);
    select
      -- Execution-time overrun detection.
      The_Timer.Time_Exceeded;
      Handle_The_Error;
    then abort
      Do_Useful_Work;
    end select;
    The_Timer.Disarm;
    Next_Start := Next_Start +
      Real_Time.To_Time_Span (Period);
    delay until Next_Start;
  end loop;
end Periodic_Stopped;
```

Here the call to `Arm` arms the timer so that it expires when the current task has consumed WCET units of CPU time. The loop then starts doing the useful work it's supposed to do. If that work is not completed before the timer expires, the entry call `Time_Exceeded` is accepted, the call to `Do_Useful_Work` is aborted, and the procedure `Handle_The_Error` is called.

4.3 Abstract interfaces to provide multiple inheritance (AI 251)

At the time of the design of Ada 95, it was decided that multiple inheritance was a programming paradigm which imposes too heavy a distributed overhead to introduce in Ada, where performance concerns are prevalent.

Since then, an interesting form of multiple inheritance has become commonplace, pioneered notably by Java and COM. An “interface” defines what methods a class must implement, without supplying the implementation of these methods. A class may “implement” any number of interfaces, in which case it must provide implementations for all the methods inherited from these interfaces. Interfaces have the attractive characteristic that they are relatively inexpensive at run-time. In particular, unlike full-fledged multiple inheritance, they do not impose a distributed overhead on programs which do not use them.

The ARG is studying the possibility of introducing interfaces in Ada. This is a sizeable language change, which affects many areas of the language, so this proposal is somewhat less mature than most of the others discussed in this article.

AI 251 proposes to add new syntax to declare interfaces:

```
type I1 is interface; -- A root interface.
type I2 is interface; -- Another root interface.
-- An interface obtained by composing I1 and I2:
type I3 is interface with I1 and I2;
```

An interface may have primitive subprograms, but no components. In many respects, it behaves like an abstract tagged type; in particular, constructs, which would create objects of an interface type, are illegal.

It is expected that most componentless abstract types in Ada 95 could be turned into interfaces with relatively little effort, and with the added flexibility of using them in complex inheritance lattices.

Note that, in order to preserve compatibility, the word “interface” is not a new keyword. It is a new kind of lexical element, dubbed “non-reserved keyword”, which serves as a keyword in the above syntax, but is a normal identifier everywhere else.

A tagged type may implement one or more interfaces by using a new syntax for type derivation:

```
type I4 is interface ...;
type T1 is tagged ...;
type T2 is new T1 and I3 and I4 with
  record
    ...
  end record;
```

In this instance, `T2` is normally derived from `T1` (and it inherits the primitive operations of `T1`) and implements

interfaces I3 and I4. It must therefore override all the primitive operations that it inherits from I3 and I4.

Interfaces become really handy in conjunction with class-wide types. The Class attribute is available for interfaces. Thus, a parameter of type I'Class can at execution be of any specific tagged type that implements the interface I. Similarly, it is possible to declare an access type designating I'Class, and to build data structures that gather objects of various types which happen to implement I.

Because many reusable components in Ada are written as generics, and for consistency with the rest of the language, new kinds of generic formal types are added. They are useful to pass interface types to a generic or to indicate that a formal derived type implements some number of interfaces. Their syntax is similar to that of normal interface and derived type declarations.

In implementation terms, the only operations that may incur a significant performance penalty are membership tests and conversions from a class-wide interface. Say that X is a parameter of type I'Class (I being an interface) and T is a specific tagged type. Evaluating T(X) (a conversion) requires a check at execution time that X is actually in the derivation tree rooted at T. This check already exists for normal tagged types, but it is more complex in the presence of interfaces, because in this case the derivation structure is an acyclic graph, not a tree. Also, as part of the conversion the dispatch table must be changed so as to ensure that dispatching executes the operations of T.

The following example shows a concrete case of usage of interfaces:

```

type Stack is interface;
procedure Append (S : in out Stack;
                  E : Elem) is abstract;
function Length (S : Stack) return Natural is abstract;
procedure Remove_Last (S : in out Stack;
                       E : out Element) is abstract;

type Queue is interface;
function Length (Q : Queue) return Natural is abstract;
procedure Append (Q : in out Queue;
                  E : Elem) is abstract;
procedure Remove_First (Q : in out Queue;
                        E : out Element) is abstract;

type Deque is interface with Queue and Stack;
<...>
package ... is
...
  type My_Deque is new Deque with private;
private
  type My_Deque is new Blob and Deque with
  record
    ... -- Implementation details here.
  end record;
end ...;
procedure Print (S : Stack'Class) is
  E : Elem;
begin
  if Length (S) /= 0 then
    Remove_Last (S, E);
    Print (E);
    Print (S);
    Append (S, E);
  end if;
end Print;

```

Here Stack and Queue are two interfaces that declare the operations (or the contract) characteristic of stacks and queues. Deque is an interface obtained by mixing these two interfaces. It has the characteristic that elements can be removed at either end of the queue. My_Deque is a concrete implementation of Deque, which declares record components as needed, and overrides the subprograms Append, Length, Remove_First and Remove_Last.

Finally, the procedure Print recursively prints a stack by removing its first element, printing it, printing the rest of the stack, and re-appending the first element. It can be used with any type that implements the interface Stack, including Deque and My_Deque (but not with type Queue, which does not have a Remove_Last primitive operation).

4.4 Handling mutually dependent types across packages (AI 217)

The impossibility of declaring mutually dependent types across package boundaries has been identified very early after the standardization of Ada 95. This problem, which existed all along since Ada 83, suddenly became much more prominent because the introduction of child units and of tagged types made it more natural to try and build interdependent sets of related types declared in distinct library packages.

The ARG has spent considerable effort on this issue, which is obviously a high-priority one for the Amendment. After considering many approaches, it has settled on the notion of “type stubs”.

A type stub declares an incomplete type that will be completed in another unit, and gives the name of the unit that will contain the completion:

```

type T1 is separate in P;
type T2 is tagged separate in P;

```

The fundamental rule (the one that makes it possible to break circular dependencies) is that when the above type stubs are compiled, the package P doesn't have to exist in the environment. Later, when P is compiled, the compiler will check that it declares an untagged type named T1 and a tagged type named T2.

The possible usages of the type declared by a type stub are very restricted, and similar to those of normal incomplete types: in essence, for untagged types, it is possible to declare an access type designating T1, and not much else.

For tagged types, though, two additional capabilities are provided. First, it is possible to reference the class-wide type T2'Class. Second, it is possible to declare subprograms with parameters of type T2 or T2'Class. The reason why incomplete tagged types may be used as parameters is that they are always passed by-reference, so the compiler doesn't need to know their physical representation to generate parameter passing code: it just needs to pass an address.

Note that independently of type stubs, the notion of tagged incomplete types is new: in Ada 95, incomplete types don't tell whether their completion will be tagged or not, and this is a somewhat problematic situation.

Once type stubs are available, it is possible to declare mutually dependent types in distinct library units. The following code fragment shows a simple example:

```

package Employees is
  type Department is tagged separate in Departments;
  type Dept_Ptr is access all Department'Class;
  type Employee is tagged private;
  procedure Assign_Employee (E : in out Employee;
                             D : in out Department);
  function Current_Department (D : in Employee)
    return Dept_Ptr;

private
  ...
end Employees;
package Departments is
  type Employee is tagged separate in Employees;
  type Emp_Ptr is access all Employee'Class;
  type Department is tagged private;
  procedure Choose_Manager
    (D : in out Department;
     Manager : in out Employee);

private
  ...
end Departments;

```

Here each of the packages declares a type stub for the type declared in the other package, as well as an access type designating values of this type stub. Therefore, the full type declaration of Department (not shown) may contain an Emp_Ptr, and the full type declaration of Employee may contain a Dept_Ptr.

The packages Employees and Departments may be compiled in any order, as they don't depend semantically on each other (although chances are that each body will depend semantically on the other package).

While the above example looks simple and elegant, it leads to a multiplication of access type declarations when the number of types that belong to the same cycle increases, and that may become inconvenient. In such a situation, it is advisable to gather each type stub and its attendant access type in a distinct library unit:

```

package Employees_Interface is
  type Employee is tagged separate in Employees;
  type Emp_Ptr is access all Employee'Class;
end Employees_Interface;
package Departments_Interface is
  type Department is tagged separate in Departments;
  type Dept_Ptr is access all Department'Class;
end Departments_Interface;
with Departments_Interface, Employees_Interface;
package Employees is
  type Employee is tagged private;
  procedure Assign_Employee
    (E : in out Employee;
     D : in out Departments_Interface.Department);
  ... – Same as above
end Employees;
with Departments_Interface, Employees_Interface;
package Departments is
  type Department is tagged private;
  procedure Choose_Manager
    (D : in out Department;
     Manager : in out Employees_Interface.Employee);
  ... – Same as above
end Departments;

```

Here, all the types that are in a cycle involving, say, Employees, can just reference the package Employees_Interface, and don't need to declare their own type stub or access type.

4.5 Accidental overloading when overriding (AI 218)

It is possible in Ada 95 (and in other programming languages, e.g., Java, C++) for a typographic error to change overriding into overloading. When this happens, the dynamic behavior of a program won't be what the author intended, but the bug may be very hard to detect. Consider for instance:

```

with Ada.Finalization;
package Root is
  type Root_Type is new Ada.Finalization.Controlled
    with null record;
  -- Other primitive operations here.
  procedure Finalize (Object : in out Root_Type);
end Root;
with Root;
package Leaf is
  type Derived_Type is new Root.Root_Type
    with null record;
  -- Other primitive operations here.
  procedure Finalise (Object : in out Derived_Type);
end Leaf;

```

Here presumably the author of package Leaf intended to redefine the procedure Finalize to provide adequate finalization of objects of type Derived_Type. Unfortunately, she used the British spelling, so the declaration of Finalise (note the 's') does not override the inherited Finalize (with a 'z'). When objects of type Derived_Type are finalized, the code that is executed is that of Root.Finalize, not that of type Leaf.Finalise.

This is obviously a safety concern, and some programming languages (e.g., Eiffel, C#) provide mechanisms to ensure that a declaration is indeed an override. The ARG intends to provide a similar mechanism, although the details are still unclear. One approach is to define new pragmas; another is to invent new syntax. Both approaches have drawbacks, and at the time of this writing the interactions with the rest of the language have not been fully worked out, so it would not be useful to spell out the details here.

4.6 Access to private units in the private part (AI 262)

The private part of a package includes part of the implementation of the package. For instance, the components of a private type may include handles and other low-level data structures.

Ada 95 provides private packages to organize the implementation of a subsystem. Unfortunately, these packages cannot be referenced in the private part of a public package—the context clause for the private package is illegal. This makes it difficult to use private packages to organize the implementation details of a subsystem.

AI 262 defines a new kind of with clause, the “private with” clause, the syntax of which is:

```

private with A, B.C, D;

```

This clause gives visibility on the entities declared in units A, B.C, and D, but only in the private part of the unit where the clause appears. Thus, a public unit may reference a private unit in a private with.

The following is an example derived from code in the CLAW library (which provides a high-level interface over the Windows UI). The low-level interface for an image list package looks like:

```
private package Claw.Low_Level_Image_Lists is
  type HImage_List is new DWord;
  type IL_Flags is new UInt;
  ILR_DEFAULTCOLOR : constant IL_Flags :=
    16#0000#;
  ILR_MONOCHROME : constant IL_Flags :=
    16#0001#;
  ...
end Claw.Low_Level_Image_Lists;
```

The high-level interface for the image list package looks like:

```
private with Claw.Low_Level_Image_Lists;
package Claw.Image_List is
  type Image_List_Type is tagged private;
  procedure Load_Image
    (Image_List : in out Image_List_Type;
     Image : in String;
     Monochrome : in Boolean := False);
  ...
private
  type Image_List_Type is tagged
  record
    Handle : Claw.Low_Level_Image_Lists.HImage_List;
    Flags : Claw.Low_Level_Image_Lists.IL_Flags;
    ...
  end record;
end Claw.Image_List;
```

Here the private part of the high-level package needs to reference declarations from the low-level package. Because the latter is a private child, this is only possible with a private with clause.

4.7 Downward closures for access to subprogram types (AI 254)

One very important improvement in Ada 95 was the introduction of access-to-subprogram types, which make it possible to parameterize an operation by a subprogram. However, this feature is somewhat limited by the accessibility rules: because Ada (unlike, say, C), has nested subprograms, the language rules must prevent the creation of dangling references, i.e., access-to-subprogram values that outlive the subprogram they designate.

Consider for example the case of a package doing numerical integration:

```
package Integrate is
  type Integrand is access function (X : Float)
    return Float;
  function Do_It (Fn : Integrand; Lo, Hi : Float)
    return Float;
end;
```

This package works fine if the function to be integrated is declared at library level. For instance, to integrate the predefined square root function, one would write:

```
Result := Integrate.Do_It
  (Ada.Numerics.
   Elementary_Functions.Sqrt'Access,
   0.0, 1.0);
```

However, the package Integrate cannot be used with functions that are not declared at library level. One good practical example of this situation is double integration, where the function to be integrated is itself the result of an integration:

```
function G (X : Float) return Float is
  function F (Y : Float) return Float is
  begin
    ... -- Returns some function of X and Y.
  end F;
begin
  return Integrate.Do_It (F'Access, 0.0, 1.0); -- Illegal.
end G;
```

```
Result : Float := Integrate.Do_It (G'Access, 0.0, 1.0);
```

The accessibility rules are unnecessarily pessimistic here, because the integration algorithm does not need to store values of the access-to-subprogram type Integrand, so there is no risk to create dangling references.

AI 254 proposes to introduce anonymous access-to-subprogram types. Such types can only be used as subprogram parameter types. Because they cannot be used to declare components of data structures, they cannot be stored and therefore cannot be used to create dangling references. With this feature, the package Integrate would be rewritten as:

```
package Integrate is
  function Do_It
    (Fn : access function (X : Float) return Float;
     Lo, Hi : Float)
    return Float;
end;
```

Now, the double integration example above would be legal, as there are no accessibility issues with anonymous access types. Note that the syntax may look a bit heavyweight, but it actually follows the one that was used in Pascal.

4.8 Aggregates for limited types (AI 287)

Limited types allow programmers to express the idea that copying values of a type does not make sense. This is a very useful capability; after all, the whole point of a compile-time type system is to allow programmers to formally express which operations do and do not make sense for each type.

Unfortunately, Ada places certain limitations on limited types that have nothing to do with the prevention of copying. The primary example is aggregates: the programmer is forced to choose between the benefits of aggregates (full coverage checking) and the benefits of limited types. These two features ought to be orthogonal, allowing programmers to get the benefits of both.

AI 287 proposes to allow aggregates to be of a limited type, and to allow such aggregates as the explicit initial value of objects (created by object declarations or by allocators). This change doesn't compromise the safety of limited types; in particular, assignment is still illegal, and the initialization expression for a limited object cannot be the name of another object; also, there is no such thing as an aggregate for a task or protected type, or for a limited private type.

Because a limited type may have components for which it is not possible to write a value (e.g., components of a task or protected type), AI 287 also allows a box “ \diamond ” in place of an expression in an aggregate. The box is an explicit request to use default initialisation for that component.

The following is an example of an abstraction where copying makes no sense, so the type `Object` should be limited. Actually, this type has a component of a protected type, so it *has to be* limited.

```

package Dictionary is
  type Object is limited private;
  type Ptr is access Object;

  function New_Dictionary return Ptr;
  ...
private
  protected type Semaphore is ...;
  type Tree_Node;
  type Access_Tree_Node is access Tree_Node;

  type Object is limited
    record
      Sem: Semaphore;
      Size : Natural;
      Root : Access_Tree_Node;
    end record;
end P;
package body Dictionary is
  ...
  function New_Dictionary return Ptr is
  begin
    return new T(Sem => <>,
                Size => 0,
                Root => null);
  end New_Dictionary;
  ...
end Dictionary;

```

With the introduction of aggregates for limited types, the allocator in the body of `New_Dictionary` is legal. Note that in Ada 95, one would have to first allocate the object, and then fill the components `Size` and `Root`. This is error-prone, as during maintenance components might be added and not initialized. With limited aggregates, the coverage rules ensure that if components are added, the compiler will flag the aggregates where new component associations must be provided.

4.9 Pragma assert, pre-conditions and post-conditions (AIs 286 and 288)

Several Ada compilers support an `Assert` pragma, in largely the same form. As part of the Amendment work, the ARG intends to standardize this pragma, an associated package, and an associated configuration pragma for controlling the effect of the pragma on the generated code.

AI 286 defines pragma `Assert` as taking a Boolean expression and optionally a message:

```

pragma Assert (Angle in 0.0 .. Pi / 2.0 or
              Angle in Pi .. 3 * Pi / 2.0,
              Message => "Angle out of range");

```

This pragma may appear anywhere, including in a declarative part. At execution, the Boolean expression is evaluated, and if it returns `False` the exception `Assertion_Error` is raised with the indicated message. This exception is declared in a new predefined package, `Ada.Assertions`:

```

package Ada.Assertions is
  pragma Pure (Ada.Assertions);
  Assertion_Error : exception;
end Ada.Assertions;

```

In practice, it is also useful to be able to enable or disable assertion checking in an entire program or in a collection of

units. In order to help with this usage model, AI 286 also defines a configuration pragma, `Assertion_Policy`. Thus, the pragma:

```

pragma Assertion_Policy (Ignore);

```

disables all assertion checking in the entire environment or in specific units, while the pragma:

```

pragma Assertion_Policy (Check);

```

enables normal assertion checking.

On a related topic, the ARG has studied the possibility of improving the support of the “programming by contract” model, in a manner similar to what Eiffel provides. This would be done by expressing pre- and post-conditions for subprograms, and invariants for types and packages. AI 288 defines a number of pragmas to that effect. However, this proposal is not yet mature, and its details are very much in flux.

4.10 Pragma Unsuppress (AI 224)

It is common in practice for some parts of an Ada program to depend on the presence of the canonical run-time checks defined by the language, while other parts need to suppress these checks for efficiency reasons. For example, consider a saturation arithmetic package. The multiply operation might look like:

```

function "*" (Left, Right : Saturation_Type)
  return Saturation_Type is
begin
  return Integer (Left) * Integer (Right);
exception
  when Constraint_Error =>
    if (Left > 0 and Right > 0) or
       (Left < 0 and Right < 0) then
      return Saturation_Type'Last;
    else
      return Saturation_Type'First;
    end if;
end "*";

```

This function will not work correctly without overflow checking. Ada 95 does not provide a way to indicate this to the compiler or to the programmer.

AI 224 introduces the configuration pragma `Unsuppress`, which has the same syntax as `Suppress`, except that it cannot be applied to a specific entity. For instance, to ensure correctness of the above function, one should add, in its declarative part, the pragma:

```

pragma Unsuppress (Overflow_Check);

```

The effect of this pragma is to revoke any suppression permission that may have been provided by a preceding or enclosing pragma `Suppress`. In other words, it ensures that the named check will be in effect regardless of what pragmas `Suppress` are added at outer levels.

4.11 Unchecked unions: variant records with no run-time discriminant (AI 216)

Ada does not include a mechanism for mapping C unions to Ada data structures. At the time of the design of Ada 95, it was thought that using `Unchecked_Conversion` to obtain the effect of unions was satisfactory. However, easy interfacing with C unions is important enough that several compilers have defined a method to support it. The ARG has decided to standardize this interfacing technique.

AI 216 defines a new representation pragma, `Unchecked_Union`, which may be applied to a discriminated record type with variants to indicate that the discriminant must not exist at run-time. For example, consider the following C type, which could represent an entry in the symbol table of a compiler; where a symbol could be either an object or a package:

```
struct sym {
  int id;
  char *name;
  union {
    struct {
      struct sym *obj_type;
      int obj_val_if_known;
    } obj;
    struct {
      struct sym *pkg_first_component;
      int pkg_num_components;
    } pkg;
  } u;
};
```

This data structure maps to the following unchecked union type in Ada:

```
type Sym;
type Sym_Ptr is access Sym;
type Sym_Kind_Enum is (Obj_Kind, Pkg_Kind);
```

```
type Sym(Kind : Sym_Kind_Enum :=
          Sym_Kind_Enum.First) is
```

```
record
  Id : Integer := Assign_Id (Kind);
  Name : C_Ptr;
case Kind is
  when Obj_Kind =>
    Obj_Type : Sym_Ptr;
    Obj_Val_If_Known : Integer := -1;
  when Pkg_Kind =>
    Pkg_First_Component : Sym_Ptr;
    Pkg_Num_Components : Integer := 0;
end case;
end record;
pragma Unchecked_Union (Sym);
```

Because of the presence of the pragma, the discriminant `Kind` is not present at run-time. This means that the application has to be able to determine from the context if a symbol is a package or an object. Those operations which would normally need to access the discriminant at run-time (like membership test, stream attributes, conversion to non-union types, etc.) raise `Program_Error`. Note that representation clauses may be given for an unchecked union type, but it is obviously illegal to give a component clause for a discriminant.

4.12 Directory operations (AI 248)

Most modern operating systems contain some sort of tree-structured file system. Many applications need to manage these file systems (by creating and removing directories, searching for files, and the like). Many Ada 95 compilers provide access to these operations, but their implementation-defined packages differ in many ways, making portable programs impossible.

The POSIX libraries provide operations for doing this, but these libraries usually are available only on POSIX systems, leaving out many popular operating systems including MS-DOS, most flavours of Windows, and even Linux.

Therefore, the ARG has decided to standardize a minimum set of capabilities to access directories and files. The purpose is not to provide interfaces to all the features of the underlying file system, but rather to define a common set of interfaces that makes it possible to write programs that can easily be ported on different operating systems. This is similar in spirit to the definition of `Ada.Command_Line` in Ada 95.

AI 248 defines a new predefined package, `Ada.Directories`, with operations to:

- Query and set the current directory;
- Create and delete a directory;
- Delete a file or a directory or an entire directory tree;
- Copy a file and rename a file or a directory,
- Decompose file and directory paths into enclosing directory, simple name, and extension;
- Check the existence, and query the size and modification time of a file;
- Iterate over the files and directories contained in a given directory.

It is intended that implementations will add operating-system-specific children of `Ada.Directories` to give access to functionalities that are not covered by this unit (e.g., file protections, encryption, sparse files, etc.).

Conclusion

This paper has given an overview of the standardization process for the 2005 revision of Ada, and it has presented those proposals that are reasonably stable and mature.

The ARG has also studied (and is still studying) a large number of other topics including: generalizing the use of anonymous access types; improving formal packages; allowing access-to-constant parameters; improving visibility rules for dispatching operations; and supporting notification of task termination. Some of these ideas seem promising, other might be disruptive, and so it's too early to tell which ones will make it into the Amendment and which ones will not.

The ARG will be pleased to receive comments and suggestions on the work it is doing on the Amendment. Our collective goal is to make Ada more powerful, safer, more appealing to its users, without imposing unreasonable implementation or training costs. Whatever the details of the changes that ultimately make it into the Amendment, we anticipate that 2005 will be a good vintage.

References

- [1] S. T. Taft, R. A. Duff, R. L. Brukardt, E. Ploedereder, Consolidated Ada Reference Manual, LNCS 2219, Springer-Verlag.
- [2] A. Burns, "The Ravenscar Profile", *ACM Ada Letters*, Vol. XIX, No. 4, pp. 49-52, December 1999.

Evolutionary Computation in Ada 95: A Genetic Algorithm Approach

Luís M Santos

University of Algarve, Campus de Gambelas, Faro, Portugal; email: lmsantos@ualg.pt

Abstract

Ada has started to find its place in fields traditionally dominated by C and Fortran, for instance, high performance computing. In many of these fields there are several computational problems for which the evolutionary computation theory provides elegant solutions. Among the evolutionary computation paradigm, the genetic algorithms have seen a wide dissemination and application to different domains. The increasing interest in genetic algorithms originated the proliferation of libraries and other software solutions for their implementation.

In this paper, an object-oriented approach to the modelling of genetic algorithms and their implementation in Ada95 are presented. The Ada95 constructs used to implement the software engineering analysis is also discussed. Further insights are given relative to state-of-the-art parallel genetic algorithms, together with their implementation by concurrent features. Experiments performed will illustrate the feasibility of Ada95 as a computing platform for computation-intensive tasks.

Keywords: Ada95, Evolutionary computation, Genetic Algorithms.

1 Introduction

Evolutionary computation encompasses methods of simulating evolution on a computer. Different paradigms exist to perform this task, most of them were derived from the fields of evolution strategies, evolutionary programming, artificial life and genetic algorithms (GA). The literature has seen the publication of several computational libraries devoted to the implementation of these paradigms [3]. Researchers — most of whom not necessarily software engineers — usually develop these libraries in order to solve concrete problems¹. Their implementations are typically coded with any known tool, e.g. Matlab, C and Fortran.

Genetic Algorithms are the algorithmic models that will be discussed in this paper. This paper does not intend to give a deep introduction to GAs. Only their most relevant aspects

will be presented such that the reader may understand their implementation in Ada. For a more technical presentation, please refer to [10].

Genetic algorithms are often viewed as function optimizers, although the range of problems to which they have been applied is quite broad (e.g. simulations of military scenarios [9, 15]). In a broader usage of the term, these algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure. Recombination operators are applied to these structures so as to preserve and discover critical information by generating new sample points in a search space.

Usually, there are only two main components of most genetic algorithms that are problem dependent: the problem encoding and the evaluation function. The rest of the framework is problem independent. It can be designed in the most efficient way, and implemented in any programming language, taking into account the best software engineering principles, e.g. reusability, genericity, performance, etc.

Since the early days of evolutionary computing, various analyses have tried to apply object-oriented concepts to model their behaviour. Many attempts have been made, some of them quite successful. Two such examples are the public software packages TOLKIEN [17] and GALIB [18].

This work does not intend to be a discussion on how to develop a binding to an already existent solution, but a rather new approach using software engineering principles and the mechanisms included in Ada95 [1, 2].

The general objective behind the implementation of the library was to make it as easily usable as possible. On the other hand, it will be desirable that the use of the library require as little coding as possible by the final user.

This paper concentrates on mechanisms such as genericity, abstract types and controlled types. Throughout the paper some comparisons to C++ and Fortran90 are made, because of the wide acceptance of these languages. Based on the analysis and the experiments performed, it can be shown that Ada95 can be an implementation language as convenient as C++ and Fortran for this type of problems.

In Section 2 I present the analysis and implementation of the GA in Ada95. Section 3 discusses a parallel implementation of the island model by concurrent constructs. Section 4 presents practical applications.

¹ Evolutionary computation has been applied to a wide range of fields, including stock market prediction, military simulations, optimizations among others.

Section 5 discusses further lines of research for e.g. implementation of similar algorithms in embedded devices.

2 Genetic Algorithm model

As stated by Booch [4], in order to efficiently implement a problem in an OO language, one first needs to conceptually define classes whose interactions are well defined. Meyer [5] suggested that useful libraries of reusable Ada components depend on the development of powerful abstractions and the ease of composition to use the abstractions. "Great library design...results from a far-sighted choice of abstractions."

Conceptually, an implementation of a genetic algorithm begins with a population of random chromosomes. One then evaluates these structures and allocates reproductive opportunities in such a way that those chromosomes, which represent a better solution to the target problem, are given more chances to "reproduce" than those chromosomes, which are poorer solutions. The population is then left to evolve for several generations.

The Ada95 features used to model genetic algorithms include

- Generics - To define genes carrying different types of information (e.g. binary or quaternary). Together with inheritance, allows the instantiation of different types of chromosomes, using the same code (reusability).
- Child libraries – To define the extension of the library to implement different types of GAs, apart from Genitor. This is the only type of GA implemented in the first version of the library. Other GAs that can be implemented are e.g.. Simple, Incremental, Steady-State GA.
- Limited controlled types - To define the instantiation and assignment of chromosomes, whenever a deep copy of these structures needs to be made.
- Strong typing - Every application benefits from the fact that a large number of programming inconsistencies are detected and prevented at compile time.
- Enumerated data types – To define a construct to implement genes having discrete states.
- Array and record slicing – To define the crossover operations between chromosomes.
- Parallel constructs - Concurrent GAs can be implemented by the tasking mechanism and protected types can cater for data synchronization among them.

The advantage of Ada comes from the fact that some of the above features are not found in other languages. For instance, C++ lacks strong typing and built-in parallelism, and the notions of enumerated data types or inheritance are unknown to Fortran90 [16].

A genetic algorithm can be modelled after three major classes: *Gene*, *Chromosome* and *Population*. The Gene is the basic structure, which represents the atomic elements. A Chromosome object models a solution. This is formed by a string of genes together with other solution specific attributes (e.g. fitness²). The Population object aggregates all the chromosomes and performs evolutionary methods on them.

```

generic
  type GeneType is private;
package genebase is
  subtype probability is float range 0.0..1.0;
  type Tgenebase is tagged private;
  procedure SetGene(Object : in out Tgenebase;
    value : in GeneType);
  function GetGene(Object : in Tgenebase)
    return GeneType;
private
  type Tgenebase is tagged record
    GeneInfo : GeneType;
    Order : natural := 0;
    MutationProb : probability := 0.02;
  end record;
end genebase;

```

Figure 1 Specification of the generic package Gene.

The root Gene class was made generic to allow the derivation of different types of genes. The most common type is the gene caring binary information. This and other types of genes can be defined by inheriting from this class. Common attributes are part of the generic class, including the mutation probability and the order of the gene inside a chromosome. See Figure 1. Inheritance is then used to define specific types of genes. The methods needed to operate in these particular genes are defined in the derived class. These include e.g. the mutator and display methods.

The root chromosome is an abstract base class and cannot be instantiated. It defines a number of attributes and function prototypes specific to the chromosome and its derived classes. The genetic operators for chromosomes are sub-programs that take generic chromosomes as their arguments. This makes it possible to define new behaviours for existing chromosome classes without deriving a new class. A derived chromosome class contains several instantiations of genes (*GenomeLength*). This raised the necessity to make the Chromosome class of limited controlled type, in order to control the allocation, instantiation and assignment operations of the container. In the latter case, the need is obvious since the assignment requires a deep copy of the dynamic gene structure.

² Fitness is a chromosome's measure of adequacy to the solution of the problem.

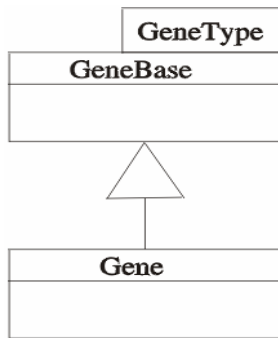


Figure 1 UML specification of the root parameterised class GeneBase, and a derived class, Gene. The derived classes instantiate the generic parent, thus specifying a concrete type for the Gene's elements. Further methods to manipulate these elements are also provided by the derived class.

The creation of these abstractions is a natural model for the problem of GA. Encapsulation and data hiding in this way makes the Chromosome class independent of the Gene implementation. This allows easy extension of the package to other types of Genes, since their definition (by generic type instantiation plus implementation of methods) are the only changes that need to be made.

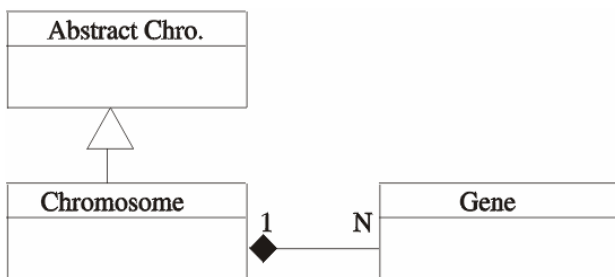


Figure 2 The Chromosome class derives from an abstract chromosome class and aggregates several genes in a container. The methods and specific attributes from the classes are not shown. The Population class is implemented in a similar fashion, by aggregating several chromosomes.

The design of the Population class is very similar to the Chromosome class. It has a container for a set of chromosomes (PopulationSize) together with population specific attributes (e.g. crossover probability) and the methods to operate on them. The population also contains a function used for selecting individuals from the population, according to different criteria. This is used for selection during evolution and, as we will see later on, migration between islands in the parallel model.

The use of a Genetic Algorithm starts by the instantiation of a population object, defining some basic parameters, and evoking the evolve method. This method then executes a Genitor [12] type of GA, by evoking a succession of sub-programs (Evaluate, Selection, Crossover, Mutate) for a specified number of generations. This method was chosen for its linear implementation (similar to the simple GA) and high learning rate.

```

with ada.finalization; use ada.finalization;
with chromosome; use chromosome;
package population is
    type Tindividuals is array(natural range <>) of
        TChromosome;
    type PtrTIndividuals is access Tindividuals;
    subtype probability is float range 0.0..1.0;
    type Tpopulation is new limited_controlled with private;
    procedure Initialize(Object : in out Tpopulation);
    procedure InitializePopulation(Object : in out
        Tpopulation);
    procedure Evolve(Object : in out Tpopulation);
private
    type Tpopulation is new limited_controlled
        with record
            PopulationIndividuals:PtrTIndividuals;
            MutationProb: probability;
            CrossOverProb : probability := 0.5;
            PopulationSize:natural := 10;
            GenomeLenght:natural := 10;
            CrossOverPoint : natural := 5;
            NGenerations : natural := 1;
        end record;
end population;
    
```

Figure 4 Definition of the limited controlled package used to implement the population class.

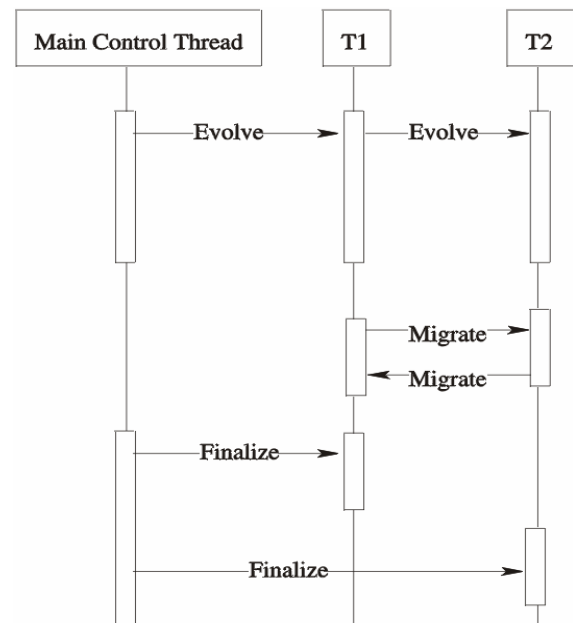


Figure 5 Simplified view of parallelism in the island model. After initialization from the main control thread, tasks T1 and T2 evolve independently. After one generation, migration occurs between them, where the best individual in one task migrates (it is effectively copied) to the next task. This process is repeated for several generations.

3 Parallel Genetic Algorithms

Since their appearance, computational evolution paradigms have been target of parallelization interest. By nature, GAs are very amenable to parallelization, and in order to avoid the premature convergence phenomenon [14], some models of parallelism have been proposed. These range from coarse-grained models to massive parallel algorithms in which each processor is responsible for an element of the population. An example of the former case is the island model [11]. In this model, several sub-populations evolve in time in separate islands. Each of these sub-populations is created randomly without any correlations with the other sub-populations. After some time, genetic information is transmitted—through a process called migration—to neighbour islands. Depending on the migration strategy used, these sub-populations might interact in a loose or very strongly coupled way. In the work developed for this paper, migration occurs at regular intervals to model periodic migration seasons. The best individual of each island is sent to its neighbour islands, in a round-robin fashion.

The Ada tasking mechanisms provide an excellent implementation means for this advanced parallel genetic algorithm. A direct implementation of the model can be obtained by mapping each island to an Ada task. The number of synchronization points (rendezvous) was kept to a minimum to improve performance. Rendezvous occur at initialization of the evolution, and after a periodic number of generations, to synchronize data and model migration.

Test	Time (s)
2 GAs in parallel	2.2
2 GAs in series	4.1
1 GA	2.11

Table 1 Computation times observed on a dual Pentium III @ 750 MHz. The genetic population was composed of 20 individuals, and the times measured after five hundred generations. The time to evaluate 2 GAs in parallel was almost the same for a single GA, this accounting for the small Ada overhead in task management.

This type of coarse-grained parallelism is very efficiently implemented in the Ada control parallelism—not so efficient in the data parallelism of e.g. Fortran90, which is better suited for massive parallel processing—since a small number of inter-task communication occurs [8]. Tasks will execute in parallel, potentially calling other subprograms, but sharing a rather small set of common data. Other paradigms of coarse-grained parallel models [13] can be implemented, after the necessary topological adjustment (mapping of GAs to tasks). These are left for further development for this work.

Data is exchanged between different tasks through a shared data structure. For that purpose, Ada provides protected types. Methods to manipulate these types can be defined in

order to access data in mutual exclusion grained. Figure 5 shows an UML [6] sequence diagram with the interaction between two tasks.

It was reported that the Ada tasking mechanism introduces minimal overhead (about 1%) in task switching. Thus, using a delay to achieve collaborative multitasking approximated an almost perfect load balancing of the tasks. In a system with N_p processors and N_t GAs, each one corresponding to an individual task, the delay τ can be defined as:

$$\tau = \left(\frac{N_t}{N_p} - 1 \right) T_t$$

where T_t is the time to compute a single generation.

4 Experiments

Some experiments were performed in order to be representative of real-case scenarios, and test different aspects of the library. Two of these examples will be presented in this Section. The library was first tested by defining a population of 20 individuals containing 20 genes each. The genes can have two possible values, A and B.

The cost function³ was designed to find alternating sequences of A and B. Tests demonstrated a mean convergence of the entire population around 100 generations.

Another test was aimed at finding a path through a 3D space with obstacles. The objective was to find the best path for a particle to move from a point A to a point B on that discrete space. The cost function to be minimized was designed in a way that it would increase when the particle approaches an obstacle, or moving away from the destination. The algorithm was able to find the path to the destination while avoiding the obstacle in a number of generations approximately equal to the discrete distance between points A and B.

5 Discussion

In this paper we have seen the feasibility of using Ada and its object-oriented mechanisms to implement scientific applications, namely genetic algorithms. The last revision of Ada, where OO features became fully supported, is a fairly recent language, when compared to others more popular within the scientific community like C and Fortran. The use of these languages has become a *de facto* standard for scientific applications. New techniques in general, like OO, and Ada in particular will have to struggle their way to become more popular. The availability of libraries such as these one could be an important contribution towards that.

We have also seen how the different constructs provided by Ada can be used as an advantage in the implementation of GA. The Ada parallel constructs allowed a very linear

³ The cost function is the function that models the problem to which one wants to apply GA. It is the function that needs to be minimized (or maximized).

implementation of parallel GAs, essentially mapping a GA in a task. The model can even be extended at run time with the Ada dynamic task creation capabilities. This will enable more flexibility and increase the capability to adapt the model to specific problems.

More than just another library written in Ada, this paper brings the insight and motivation to use Ada in domains that are traditionally dominated by C (together with its OO extensions, C++) and Fortran. The library can be downloaded from http://w3.ualg.pt/~lmsantos/ada_ga.tgz.

A short-term task will be to port the library to Personal Digital Assistants and other embedded devices. This could be achieved by using the compilation of Ada source code into Java byte code—JGNAT provides an excellent tool for that—in order to use it in these devices [7]. The computational power of these devices has increased considerably in the last years, making it viable to run evolutionary algorithms on them. The existence of such a library available on these devices would make it easier to develop AI applications, and possibly offer salesmen the solution of the salesman problem.

Acknowledgments

This work was funded by the FCT (Fundação para a Ciência e a Tecnologia) contract SFRH/BD/1309/2000.

References

- [3] S. Tucker Taft, Robert A. Duff, (1997), *Ada 95 Reference Manual: Language and Standard Libraries*, International Standard ISO/IEC 8652:1995(E). Lecture Notes in Computer Science, vol. 1246. Springer-Verlag, ISBN 3-540-63144-5.
- [4] John Barnes (1997), *Ada 95 Rationale: The Language, The Standard Libraries*. Lecture Notes in Computer Science, vol. 1247. Springer-Verlag, ISBN 3-540-63143-7.
- [5] Bearley, J.H.D. (1996), *The Hitchhiker's Guide to Evolutionary Computation. v.3.4*, Technical report, available at the ENCORE sites.
- [6] Booch, G. (1994), *Object-Oriented Analysis and Design*, second edition, Benjamin Kummings (ed).
- [7] B. Meyer (1994), *Reusable Software*, Prentice Hall.
- [8] Booch, G., Rumbaugh, J., et al. (1999), *The Unified Modelling Language*, User Guide, Davis, L. (ed), Reading, Massachusetts: Addison Wesley.
- [9] B.M. Brosgol (2000), *A comparison of Ada and JavaTM as a Foundation Teaching Language*, Ada Core Technologies, available at <http://www.gnat.com>.
- [10] Burns, A. and Wellings, A. (1995), *Concurrency in Ada*, Cambridge University Press.
- [11] C.M. Fonseca and P.J. Fleming (1995), *An Overview of Evolutionary Algorithms in Multiobjective Optimization*, Evolutionary Computation, vol 3 no 1, pp 1-16.
- [12] Davis, L. (1991), *Handbook of Genetic Algorithms*, David, L. (ed.) Van Nostrand Reinhold, New York.
- [13] D. Whitley (1993), *Toward Models of Island and Cellular Parallel Genetic Algorithms*, Inter. Conf. Genetic Algorithms (submitted).
- [14] D. Whitley (1989), *The GENITOR algorithm and selective pressure: Why rank-based allocation of reproductive trials is best*, Proceedings of the Third International Conference on Genetic Algorithm, In Schaer, J. D. (ed), Mateo, CA: Morgan Kaufmann, pp 116-121.
- [15] E. Cantú-Paz (2000), *Efficient and Accurate Parallel Genetic Algorithms*, Genetic Algorithms and Evolutionary Computation, Kluwer, vol 1.
- [16] J. Schaffer and L. Eshelman and D. Offutt (1991), *Spurious correlations and premature convergence in genetic algorithms*, In “Spurious correlations and premature convergence in genetic algorithms”, J.D. Schaffer, L.J. Eshelman, and D. Offutt (eds), vol 1, pp 102-112.
- [17] J. Schlabach, C. Hayes and D. Goldberg (1997), *SHAKA-GA: A Genetic Algorithm for Generating and Analyzing Battlefield Courses of Action*, University of Illinois UC (white paper), C. Engle Jr. (ed), Anaheim, California.
- [18] K. Mangold (1995), *Ada95 - An Approach to Overcome the Software Crisis?*, Ada in Europe, M. Toussaint (ed), Springer-Berlin, pp 4-10.
- [19] Tang, A. (1994), *Constructing GA applications using TOLKIEN*, Technical report, Dept. Computer Science, Chinese University of Hong Kong.
- [20] Wall, M. (1995), *Overview of Matthew's genetic algorithm library*, <http://lancet.mit.edu/ga>

COTS RTOS Suitability for High Integrity Systems

Brian Dobbing¹, Terry Totten²

¹ Praxis Critical Systems Ltd, 20 Manvers Street, Bath BA1 1PX email: brian.dobbing@praxis-cs.co.uk

² QinetiQ Limited, Cody Technology Park Farnborough GU14 0LX email: totten@QinetiQ.com

Abstract

This paper reports on a study that has been carried out by Praxis Critical Systems Ltd for QinetiQ into the suitability of commercial-off-the-shelf (COTS) real-time operating systems (RTOS) for use in high integrity and safety critical avionics software architectures, in particular those that are being proposed for Integrated Modular Avionics (IMA).

Keywords: RTOS, safety, avionics.

1 Introduction

As the use of software in critical systems grows ever more widespread and complex, so the cost of developing new software to rigorous safety standards becomes increasingly high. Historically, software for new safety critical and high integrity systems has been “bespoke” - hand-crafted according to the specific requirements of the system. But as software complexity and interaction have risen dramatically, the pressure is on to maximize re-use.

Unfortunately it is usually the case that much of the bespoke software that is in service is unsuitable for re-use because it has never been designed with that goal in mind. The advent of top-quality programming languages such as Ada provide a sound framework for the development of reusable software at the level of porting it unchanged between various kinds of processor, but if the code is not written generically enough in the first place, then it is rarely cost-effective to try to re-use the existing structure for similar, but subtly different requirements and semantics.

On the other hand, COTS software is designed from the outset for re-use. Vendors of COTS products invest significant effort into ensuring that there is sufficient configurability and parameterization to satisfy disparate customer requirements. Unfortunately COTS software is traditionally rarely developed to the rigorous standards and processes that are required for use in high integrity systems, and is usually classed as being Software of Unknown Pedigree (SOUP).

As the need for software re-use in high integrity systems has intensified, and interest in the use of COTS software for this purpose has grown, so the COTS product vendors have reacted to the market opportunities and have recently developed a few products to industry safety standards, including some to the highest integrity levels required by civil and military avionics systems. However there is generally a deficiency in communicating to the COTS

product vendor’s software engineering team what such end-users actually require, and so even today’s products that make grand claims about certifiability may in some cases be somewhat inadequate for use in say a safety critical SIL4 military avionics application.

Of the many general purpose COTS software packages of potential application in advanced avionics systems the Real-Time Operating System (RTOS) is of particular interest for two reasons.

Firstly, the operating system has become the accepted means of controlling resources in all forms of computer system. While the real-time variant has taken longer to become established, it is now in routine use in a wide range of embedded control environments. As its use has increased the RTOS has borrowed features from its mainframe and workstation counterparts. Some of these features appear to offer substantial benefits for IMA systems. For example, memory partitioning (which often requires hardware support from a memory management unit) can provide the isolation between avionics functions that was provided by physical separation in older “federated” avionics systems. Such partitioning is likely to be an essential component of a safety argument for tightly integrated systems such as IMA.

The second reason for our interest in the RTOS is that, rather than benefiting from the functional partitioning mentioned above, the RTOS must actually provide it. Consequently, the RTOS must be a high integrity component. In fact, its integrity must be at least as high as that of the highest integrity application function under its control. The importance of the RTOS is increased by the fact that, with current designs, an RTOS kernel will be present in every general purpose processing node of a distributed avionics system. A fault that occurs in the execution of the RTOS, unless mitigated by some mechanism external to it, has the potential to cause failure of the complete IMA system rather than merely a single avionics function.

QinetiQ has been carrying out research into several RTOS related areas for a number of years. Of particular interest has been the definition of open architectures for avionics that would facilitate the incorporation of COTS software components without locking the aircraft operator into a particular COTS vendor. More recently attention has turned to the safety and certification issues raised by the use of a COTS RTOS within such architectures. Thus a study into the suitability of COTS RTOS products for use within a military standard software architecture was defined by

QinetiQ, and was undertaken by Praxis Critical Systems Limited and QinetiQ. This paper summarizes the results of this study. The full details of the study may be found in the Final Reports of the two phases of the project [1, 2].

2 Approach

The study has addressed this topic from two opposite directions:

- 1 From the avionics application perspective, it is necessary to evaluate whether a safety case can be made for using a COTS RTOS product within subsystems that are at the highest integrity levels.
- 2 From the COTS RTOS perspective, it is necessary to determine the safety requirements on the RTOS component in IMA systems such that the vendor can develop the right product for the job.

A sample high integrity military avionics application was selected as the basis for making a safety case for the RTOS component of the underlying software environment, which was assumed to conform to the Allied Standards Avionics Architecture Council (ASAAC) Phase II Software Architecture. This architecture is outlined in Appendix A of the Final Report for phase 1 of the study [1]. The selected application is a Stores Management System for a military aircraft that manages the release of the weapons. This application exhibits the need to support both continuous and on-demand functionality. A few sample subsystems were selected and were analyzed to identify the fault trees that define the various hazards at the RTOS interface level, their causal effects, and their consequences. These hazards include failures within the RTOS component itself, as well as failures at the application level and at the software architecture level whose hazardous effect may be mitigated by certain features and safeguards within the RTOS component.

The result of putting together the safety case was a set of RTOS requirements for safety, functionality, performance, configurability, maintainability, support for high-level languages such as Ada, and availability of evidence of reliability and certifiability. The production of these requirements then led to the question of whether any current COTS RTOS products would be able to meet all of them and if not, whether any particular critical requirement(s) were actually impractical to meet, and whether any that could practically be met were rarely, if ever supported.

In order to check the requirements in practice, a sample COTS RTOS product that has achieved a measure of certification against an accepted standard was assessed as a candidate for the RTOS component within an ASAAC-compliant software architecture for a SIL 4 system that is to be certified using guidelines based on the UK standard Def Stan 00-55 [4]. This assessment took the form of an initial round of questions and answers based on various materials defining the product that were supplied by the vendor, followed by a two-and-a-half day on-site audit of the internals of the product and its certification evidence. The

audit focused on obtaining sufficient information about the functionality and certifiability of the product so as to be able to assess it against the requirements that had been derived from the safety case.

3 Results

The results of the study were rather mixed. On the positive side, a safety case was successfully put together for the use of a generic COTS RTOS in a high integrity military application. The identified hazards were categorized into those that relate to temporal non-determinism, temporal inaccuracy, functional errors, resource exhaustion, communication errors, data corruption, and lack of error recovery mechanisms. An example of a hazard in each of these categories follows:

- 1 Unbounded priority inversion on contention when accessing shared resources.
- 2 Coarse granularity for timing operations leading to release jitter.
- 3 Invalid parameters in RTOS API calls.
- 4 Fragmentation of memory for dynamic allocation.
- 5 Unreliable message transmission.
- 6 Data corruption due to lack of adequate protection mechanisms.
- 7 Inability to perform selective restarting of failed processes / threads.

In order to mitigate the effect of the full set of identified hazards, almost 100 requirements for the use of a COTS RTOS product in current IMA software architectures were generated. These requirements varied in importance and were rated as being mandatory, recommended or nice-to-have. All requirements that had to be met to fulfill the safety case were classed as mandatory.

On the less positive side, the audit of the sample certified COTS RTOS product revealed a number of inadequacies in both functionality and availability of quality certification evidence. The conclusion was that this product in its present form and without external mitigation measures was not suitable for use in the full range of IMA applications because safety would be compromised, although it may be suitable for high availability / fault tolerant systems that do not require any safety guarantees, for example telecommunications.

As a follow-on from the study, another audit of a COTS RTOS product for which certification materials were being developed was carried out. This audit yielded more positive results, giving some confidence that it would be possible to use a suitable COTS RTOS in a high integrity IMA system, although the lack of a modular safety case weakened the argument for its acceptability within a system at Def Stan 00-55 SIL 4.

The other main issue resulting from the audits is that the trend amongst COTS RTOS products is for them to assume total control of the processor, with limited scope for

delegation of responsibility and configurability. This model conflicts somewhat with the structure of the ASAAC software architecture, in which the operating system is hardware independent and is mainly concerned with the scheduling and partitioning of the applications, whilst the separate module support layer is responsible for the hardware level support such as interrupts, timers, I/O devices, traps and MMU control. Nevertheless, the use of a carefully crafted Board Support Package as part of the COTS RTOS architecture should provide a degree of configurability, and allow for some clear separation of responsibilities.

4 Further Work

In order for several general purpose COTS RTOS products to be usable in high integrity subsystems, there needs to be some convergence between the stringent requirements of the software architecture, and the quality of certification evidence and functionality of COTS RTOS products. Further work could examine the scope for such convergence in two ways, by:

- 1 examining whether the long-term evolution of COTS RTOS products can be influenced by identifying the inadequacies that are commonly found in today's products with respect to their functionality, certifiability and suitability for use in safety critical subsystems;
- 2 examining whether the long-term evolution of software architectures such as that defined by ASAAC can be influenced towards a higher level of compatibility and integration with general purpose COTS RTOS products, for example by identifying architecture defence mechanisms to mitigate hazards that may result from commonly-occurring RTOS deficiencies.

In order to obtain a broader view of the current suitability of COTS RTOS products for high integrity systems so as to provide data for goals a) and b) above, it is necessary to perform further audits based on the requirements that have been defined in the study. Although the results of one of the audits of the sample COTS RTOS products were disappointing with respect to the degree to which the RTOS currently meets the requirements, it is clear that there are several other vendors of similar general purpose RTOS products that make claims of certifiability. Some of these products may provide a higher degree of compatibility with the requirements, and the data from these audits would also serve as identifying common themes and patterns across the broad COTS RTOS product range. Thus the short-term future work should focus on performing further audits in order to gather this information, particularly since COTS RTOSs are being actively considered for potential use in forthcoming military and civil avionics programmes.

When selecting a suitable candidate COTS RTOS product for a follow-up audit, a number of considerations should be taken into account:

- 1 Support for applications written in distinct source languages to include at least Ada95 and C.

- 2 Certification evidence that relates to the higher integrity levels within stringent standards such as RTCA/DO-178B [3] levels A and B and Def Stan 00-55 [4] SIL 4.
- 3 In-service history within safety-critical application environments, preferably including the avionics sector, to demonstrate a degree of functional compliance.

When considering support for Ada95, it is also important to ensure that the required certification evidence for the Ada runtime system is present.

5 Conclusions

It is clear that the demands of software re-use will result in the extensive use of COTS products in high integrity systems. As this market need expands, so will the COTS vendor response in the supply of products with certification claims. It is important to be able to assess these claims objectively, so that unsuitable products are not used particularly within safety critical systems.

It is also important to be able to provide guidelines to COTS RTOS vendors as to how their products could evolve to maximize their suitability of use in high integrity systems.

Finally it is important that the evolution of software architectures such as ARINC 653 APEX [5] and ASAAC Phase II Architecture should take into consideration common limitations of COTS RTOS products, such that alternative strategies can be adopted to compensate for such deficiencies and to provide mitigation of any resulting hazards.

The paper has reported the findings of a study that has produced a safety case for the real-time operating system component of a standard software architecture for high integrity military avionics applications, and has generated almost 100 requirements that can be used to assess the suitability of COTS products for this kind of use.

References

- [21] DERA COTS Safety, Reliability and Certification Study, Phase 1 Final Report, Praxis Critical Systems document S.P1078.40.3, Definitive Issue 1.0, 05/06/2001.
- [22] COTS Safety, Reliability and Certification Study, Phase 2 Final Report, Praxis Critical Systems document S.P1078.41.4, Definitive Issue 1.0, October 2001.
- [23] RTCA DO-178B, Software Considerations in Airborne Systems and Equipment Certification, 1992.
- [24] Def Stan 00-55, Requirements for Safety Related Software in Defence Equipment.
- [25] Minimum Operational Performance Standards for Avionics Computer Resource, Version 4.0, RTCA SC-182 / EUROCAE WG-48, August 23rd 1999.

Ada-Europe 2002 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 GmbH

Contact: *Peter Dencker*

Durlacher Allee 95, D-76137 Karlsruhe, Germany

Tel: +49-721-98653-22

Email: dencker@aonix.de

Fax: +49-721-98653-98

URL: www.aonix.com

DDC International A/S (DDC-I A/S)

Contact: *Erik Jensen*

Gammel Lundtoftevej 1B, DK-2800 Lyngby, Denmark

Tel: +45-45-87-11-44

Email: eje@ddci.dk

Fax: +45-45-87-22-17

URL: www.ddci.com

Green Hills Software GmbH

Contact: *Hans-Joachim Fischer*

Unterreit 6, D-76135 Karlsruhe, Germany

Tel: +49-721-9862580

Email: sales-ger@ghs.com

Fax: +49-721-9862581

URL: www.ghs.com

RainCode Corp

Contact: *Deborah Torrekens*

1 Rue de l'Autonomie, B-1070 Bruxelles, Belgique

Tel: +32-2-522-06-63

Email: Deborah@raincode.com

Fax: +32-2-522-09-30

URL: www.raincode.com

Rational Software Austria GmbH

Contact: *Martina Rumpeltes*

Lassallestraße 7b, A-1020 Wien, Austria

Tel: +43-1-245-31-160

Email: info@rational.com

Fax: +43-1-245-31-200

URL: www.rational.com

TNI Europe Limited

Contact: *Pam Flood*

58A Mill Street, Congleton, Cheshire CW12 1AG, UK

Tel: +44-1260-29-14-49

Email: info@tni-europe.com

Fax: +44-1260-29-14-49

URL: www.tni-europe.com