ADA USER JOURNAL

Volume 24 Number 4 December 2003

Volume 25 Number 1 March 2004

Contents

Editorial Policy for *Ada User Journal*

Publication

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

Aims

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

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

The Journal publishes the following types of material:

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

Further details on our approach to these are given below.

Original Papers

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

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

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

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

Unless the authors state explicitly otherwise, submission of an article is taken to imply that it represents original, unpublished work, not under consideration for publication elsewhere.

News and Product Announcements

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

Reprinted Articles

While original material is our first priority, we are willing to reprint (with the permission of the copyright holder) material previously submitted elsewhere if it is appropriate to give it a wider audience. This includes papers published in North America that are not easily available in Europe.
We have a reciprocal approach in granting permission for other publications to reprint papers originally published in *Ada User Journal.*

Commentaries

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

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

Announcements and Reports

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

Reviews

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

Submission Guidelines

All material for publication should be sent to the Editor, preferably in electronic format. The Editor will only accept typed manuscripts by prior arrangement.
Prospective authors are encouraged to contact the Editor by email to determine the best format for submission. Contact details can be found near the front of each edition.
Example papers conforming to formatting requirements as well as some word processor templates are available from the editor. There is no limitation on the length of papers, though a paper longer than 10,000 words would be regarded as exceptional.

Editorial

The shipment of this issue has undoubtedly endured your continued patience and, surely not a consolation to you, it actually has strained but not won our perserverance. You certainly appreciate that the production of the Ada User Journal is entirely based on volunteer work, and, at that, of people much doted in overloaded work already without it. The pipeline work that takes the Journal to your mailtray is much exposed to glitches that may delay it considerably. We have been looking and will continue to look very thoroughly at ways of shortening the turnaround time and of lessening the risk factors. Some measures have already been taken, others will be in the near future.

In front of you stands a most notable effect of the first range of corrective measures that we recently took, after seeking and obtaining the approval of the Ada-Europe Board: we have combined the much belated 24-4 issue of the journal, with 25-1, which we managed to produce in a stunning record time, into a 92-page journal that hopefully delivers enough bundled value to our readership in a time frame that will also help us get increasingly closer to the scheduled issue dates. What we achieved for the 25-1 part of this bundled issue was due to a number of factors, one of which is particularly worth of mention: after several years of dedicated -- and increasingly too heavy -- service as editor of the News section of the journal, Dirk Craeynest has finally found a young, energetic and motivated replacement in the person of Santiago Urueña Pascual, a PhD student at the Technical University of Madrid in Spain, a country that continues to offer phenomenal support to the Ada community. The production of the 25-1 News section was a joint effort of Dirk and Santiago, which proved exceptionally smooth and effective. Dirk will survey the work of Santiago in the next quarter, with a view to seeing Santiago take over the full load of experience that Dirk built over his years of service. We should all welcome Santiago into his new job of News editor.

Let me come now to the actual contents of this bundled issue, which, out of backward- and forward- looking articles, spans almost of full year of very significant Ada-related activity.

In the 24-4 part of the issue, Brian Dobbing, John Barnes and Miguel Pinho give us a witty insider's account of the 12th International Real-Time Ada Workshop, which took place in gorgeous Viana do Castelo, Portugal, in September 2003. Next to this, Miguel Masmano reports his first-time experience as a contributing attendee, and an officially prized one, at that, to the Ada-Europe 2003 conference in Toulouse.

The 25-1 part of the issue includes the agenda for the 2004 General Assembly of Ada-Europe and the motions currently proposed for it. Article-wise, it features three short but very important documents (all cleared by their respective authority) that will help our readership gain a more thorough understanding of the directions taken by the Ada Rapporteur Group in their preparatory work for the 2005 revision of the Ada language standard. And finally, with a grateful thank to the program chairs of the Ada-Europe 2004 conference, we are also able to offer an advance insight into the keynote talk that Tucker Taft will deliver as the opening event.

In the confident hope of having delivered value that you will appreciate despite the long lead time, we look forward to more timely appointments in the future with the Ada User Journal.

Tullio Vardanega
Padova
December 2003 – March 2004
Email: tullio.vardanega@math.unipd.it

News – 24-4

Dirk Craeynest (ed)

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

Contents

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]

Jun 5 - ACT presentation at DASIA 2003

URL:
http://www.eurospace.org/presentations_dasia2003/dasia2003_presentation_54.pdf

[Extract of program for Thursday 5 June 2003 -- dc]

Benefits and Misunderstandings of Free Software in the European Space Industry (C. Comar, F. Gasperoni - ACT Europe, France, R. Dewar – Ada Core Technologies, USA)

Sep 9 - Baltimore/DC SIGAda Meeting on Highlights of Ada-Europe 2003

From: Currie Colket <colket@mitre.org>
Date: Thu, 28 Aug 2003 11:04:12 -0400
Organization: The MITRE Corporation
Subject: Baltimore SIGAda Chapter Scheduled for Tuesday, 9 September 2003

Our Next Meeting is scheduled for Tuesday, 9 September 2003, [...].

Currie Colket and Bill Bail will be speaking on "Highlights of the 8th International Conference on Reliable Software Technologies (Ada-Europe 2003), Held at Toulouse, France from 16-20 June 2003".

The presentation will start at 7:30 P.M. [...] at the Johns Hopkins University/Applied Physics Laboratory in Laurel, Maryland. [...]

Slides From Karl Nyberg's Presentation Available:

At the DC SIGAda meeting on 8 May 2003, Karl Nyberg of the Grebyn Corporation gave an excellent presentation titled: Optical Character Recognition (OCR) of Cryptographic Source Code. [See also "May 8" topic in AUJ 24.3 (Sep 2003), p.132. -- dc] Slides from his presentation are available online as a PowerPoint Presentation at http://www.acm.org/sigada/locals/dc/20030 5_OCR_of_Cryptographic_Source_Code.ppt (ppt, 118KB).

Jeff Castellow, Chair, DC SIGAda

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

From: Clyde Roby <roby@ida.org>
Date: Tue, 2 Sep 2003 07:39:33 -0400
Subject: SIGAda and Conferences reminder
To: SIGAda-announce@acm.org

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

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

[...] Clyde Roby, SIGAda Secretary

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

From: Hubert Keller <keller@iai.fzk.de>
Date: Wed, 29 Oct 2003 10:19:52 +0100
Subject: Nachruf GI Teiltagung Sicherheit
To: Fachruppenleitung Ada <ada-lg@mail.gi-fb-sicherheit.de>, ada@mail.gi-fb-sicherheit.de

[Extracts translated from German. See also same topic in AUJ 24.3 (Sep 2003), p.133. -- dc]

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

The new Fachbereich "Security - Protection and Reliability" held its "Initiationstagung" under the same name, as part of this year's annual GI Conference "Informatik 2003" which was held from 29.9 to 2.10.2003 at the Johann Wolfgang Goethe university in Frankfurt/Main. With over 300 participants it was a large success and exceeded substantially the number of participants of the different predecessor conferences.

The lectures covered the entire range of Security and Safety, among others critical infrastructures, fault-tolerant systems, safety-critical software, data security, trustworthy computing, [...]

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

Ada mailing list, Ada@mail.gi-fb-sicherheit.de, http://mail.gi-fb-sicherheit.de/mailman/listinfo/ada

Oct 2 - Ada UK / Embedded Systems Club Autumn Conference

From: Hazel <Hazel@Adaxia.com>
Date: Mon, 11 Aug 2003 12:52:36 +0100
Subject: Autumn Conference - Ada UK & Embedded Systems Club
To: Ada UK & ESC Contacts <Hazel@Adaxia.com>

We are pleased to announce that the Autumn Ada UK/Embedded Systems Club conference will be held in Swindon, UK on October 2nd 2003.

We are currently putting the finishing touches to the programme for this event, and details are being posted to the Ada UK and Embedded Systems Club websites as they become available. [...]

From: Hazel <Hazel@Adaxia.com>
Date: Thu, 25 Sep 2003 22:27:56 +0100
Subject: Detailed programme for Autumn Conference now available
To: Conference Contacts <Hazel@Adaxia.com>

[...] A detailed conference programme, showing the scheduling for this two stream event, is now available at the websites: www.AdaUK.org.uk & www.EmbeddedSystemsClub.com

Two technical keynotes start and finish the day: "Java for High Integrity Systems" and "Developing Reliable Software for Embedded Applications"

As our opening keynote, [the latter] presentation sets the scene for the two parallel streams, offering a mix of 60 minutes technical presentations and the popular "short and sharp" 20 minute vendor presentation slots.

"Why use a real time OS on DSP?", Richard Blackburn, OSE.

"Web-enabling Ada applications in AWS", Franco Gasperoni, ACT Europe.

"Developments in OO tools for the aero, transport and defence industries", Bill Warwick, TNI Europe Ltd.

"ARTiSAN's Ada synchronizer", Francis Thom, Artisan Software Tools.

"Requirements driven development for embedded systems", Andy Gurd, Telelogic.

"Next generation testing tools for embedded applications", Mark Pitchford, PolySpace Technologies.

"Exception freedom - turning the dials up (and something new...)", Rod Chapman, Praxis Critical Systems.

"Network security for IPv4 and IPv6 - an overview", Paul Tingey, Wind River.

"Model-driven development with Ada", Andy Lapping, I-Logix.

"Testing embedded software to achieve quality standards using automated tools", Jim Kelly, LDRA Ltd.

"Automated global data checking for Ada", Ian Gilchrist, IPL.

"The use of off-shore software services - gain the benefit of eight years experience in 20 minutes", Steve Baker, Silver Software.

There is also an exhibition area, not to mention lunch and refreshments.

Hazel Lawton, Conference Manager, Adaxia Ltd, www.AdaUK.org.uk, www.EmbeddedSystemsClub.com

Dec 7-11 - ACM SIGAda 2003 Conference

From: ricky.sward@ix.netcom.com (Ricky E. Sward)
Date: 23 Oct 2003 19:14:33 -0700
Subject: SIGAda 2003 Conference Announcement
Newsgroups: comp.lang.ada

Conference Announcement - SIGAda 2003, 7-11 December 2003, San Diego, California, USA. Sponsored by ACM SIGAda

The SIGAda 2003 conference offers a top-quality technical program focused on important strengths of the Ada programming language. Three days of technical papers, keynotes, and invited presentations will report on how Ada is achieving success in the challenging realm of software engineering. We are fortunate to have four leaders in the software engineering community; Ben Brosgol, Joyce Tokar, David A. Wheeler, and Steve Grimaldi (tentative) will provide keynote addresses to set the tone for our conference. We are also fortunate to have Jim Moore, Tullio Vardanega, and Pascal Leroy provide a special discussion of the work of the ISO/IEC JTC1/SC22 WG9 on Wednesday afternoon. The WG9 Forum will give you an opportunity to find out how Ada will evolve to meet our future requirements and at the same time give you an opportunity to provide input to its future.

Beyond the formal conference of selected papers and presentations, SIGAda 2003 offers workshops and tutorials with the same duality of on-theme and complementary topics. SIGAda's tutorials and workshops provide full- or half-days for those working the same issues to share with each other and leverage everyone's accomplishments; workshop products are "delivered" to the community. The broad offerings of career-enhancing tutorials include basic Ada 95 introductions for software engineers new to Ada, intermediate and advanced Ada topics for practitioners striving to expand their Ada expertise, and several language-independent technology topics. Join us in understanding how these topics mutually support the disciplined development and evolution of serious, high quality software systems.

For more information on the conference schedule, [...] visit the SIGAda 2003 web site at [...] http://www.acm.org/sigada/conf/sigada2003/

Please feel free to contact me with any further questions.

Ricky E. Sward, SIGAda 2003 Program Chair

Jun 14-18 - Ada-Europe 2004 Conference

From: dirk@heli.cs.kuleuven.ac.be (Dirk Craeynest)
Date: 11 Oct 2003 12:54:17 +0200
Organization: Ada-Europe, c/o Dept. of Computer Science, K.U.Leuven
Subject: 2nd CfPapers, Reliable Software Technologies, Ada-Europe 2004
Newsgroups: comp.lang.ada,fr.comp.lang.ada

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

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

Organized, on behalf of Ada-Europe, by the University of the Balearic Islands, in cooperation with ACM SIGAda (approval pending) and Ada-Spain

Ada-Europe organizes annual international conferences since the early 80's. This is the 9th event in the Reliable Software Technologies series, previous ones being held at Montreux, Switzerland ('96), London, UK ('97), Uppsala, Sweden ('98), Santander, Spain ('99), Potsdam, Germany ('00), Leuven, Belgium ('01), Vienna, Austria ('02), Toulouse, France ('03).

The 9th International Conference on Reliable Software Technologies (Ada-Europe 2004) will take place in 2004 in Palma de Mallorca, Spain.

The full conference will comprise a three-day technical program and exhibition from Tuesday to Thursday, and parallel workshops and tutorials on Monday and Friday. [...]

The 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 exhibition on June 15, 16 and 17. Vendors of software products and services should contact the Exhibition Chair, Peter Dencker (dencker@aonix.de) at their earliest convenience for further information and to ensure their inclusion. [...]

Dirk.Craeynest@cs.kuleuven.ac.be, Ada-Europe'2004 Publicity Chair

Ada and Education

New Video on Ada Trains Laboratory

From: John McCormick <mccormick@cs.uni.edu>
Date: Thu, 7 Aug 2003 11:52:30 -0500
Subject: New Ada Trains video
To: team-ada@acm.org

This video is an update of the one that I showed in my SIGAda '99 Keynote address. [See also "Software Engineering: On the Right Track" in AUJ 22.4 (Dec 2001), p.196, and "Switching from C to Ada in a Real-Time Class" in AUJ 21.1 (Apr 2000), p.8. -- dc] It shows my new Real-Time Embedded Systems Laboratory here at the University of Northern Iowa. I worked with our video production people on campus and they did an excellent job. It includes my C versus Ada project completion data.

I have it in REAL streaming video form on the lab web site: http://www.cs.uni.edu/~mccormic/RealTime/

I hope you enjoy watching it as much as I did making it.

John W. McCormick, Computer Science Department, University of Northern Iowa, Cedar Falls, IA 50614-0507

Learning Material for Beginners

From: Jean-Yves Lenhof <jylenhof@pasdespam.fr>
Date: Tue, 07 Oct 2003 23:16:47 +0200
Subject: Re: Cherche cours Ada

Newsgroups: fr.comp.lang.ada

[Translated from French: -- dc]

> I'm looking on the net for good Ada courses for beginners, [...]

The Lovelace tutorial might be a good start...

http://www.dwheeler.com/lovelace/lovelace.htm

[See also: "Resources to Learn Ada" in AUJ 23.4 (Dec 2002), pp.190-191, and the "Ada and Eduction" news section in most AUJ issues. -- dc]

From: Stephane Richard <stephane.richard@verizon.net>
Date: Tue, 07 Oct 2003 22:07:29 GMT
Subject: Re: Cherche cours Ada
Newsgroups: fr.comp.lang.ada

If you want, go to my site (http://www.adaworld.com) and click on the "Learning Center" button. Next on either "Tutorials" or "Free books". I have 4 tutorials (among others Lovelace) and 7 PDF files in the "general" section of "Free books". Enjoy

Stéphane Richard, "Ada World" Webmaster

[See also "New AdaWorld Web Site" further in this AUJ issue. -- dc]

From: Allez le TFC <on_est_en_L1t@tfc.info>
Date: Tue, 14 Oct 2003 20:48:36 +0200
Subject: Re: Cherche cours Ada
Newsgroups: fr.comp.lang.ada

Go to the site (http://intra1-cycle.insa-tlse.fr/didac_2.htm). This is for the first year of the INSA of Toulouse. Ada is taught there from the first year on, the tutorial should help you get started.

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: 27 Jul 2003 09:59:27 -0700
Subject: Public Real-Time Ada 95 Courses 23-26 September 2003 in Carlsbad CA
Newsgroups: comp.lang.ada

Absolute Software will be holding a public course "Developing Real-Time Systems in Ada 95" on 23-26 September 2003 in Carlsbad, CA. You can find a full description and registration form on our web-site, www.abssw.com. Click the Public Courses button in the left margin.

(We also offer courses on object-oriented methods and other object-oriented languages.) [...]

From: John Robinson <John@JohnRobinsonAndAssociates.com>
Date: Thu, 28 Aug 2003 14:30:57 +0100
Organization: John Robinson And Associates Ltd
Subject: ANN: Scheduled Training Courses Autumn 2003
To: <news@JohnRobinsonAndAssociates.com>

John Robinson And Associates Ltd - Scheduled Training Courses Autumn 2003 [...]

Ada Programming Course:

15th - 19th September 2003, Cheltenham, UK;

17th - 21st November 2003, Cheltenham, UK.

[...]

http://www.JohnRobinsonAndAssociates.com

From: rod.chapman@praxis-cs.co.uk (Rod Chapman)
Date: 16 Oct 2003 04:21:26 -0700
Subject: ANN: SPARK Training courses for 2004
Newsgroups: comp.lang.ada

Details of public SPARK Ada training courses for 2004 can now be found on www.sparkada.com

There's also a new 1-day course all about RavenSPARK intended for experienced SPARK users, or for folks also attending the basic "Software Engineering with SPARK" course in the same week.

Rod Chapman, SPARK Team, Praxis Critical Systems

Ada-related Resources

Another Ada Resources Page

From: aek@vib.usr.pu.ru (Alexander Kopilovitch)
Date: 24 Aug 2003 16:27:54 -0700
Subject: Less-known Ada-related WWW page
Newsgroups: comp.lang.ada

Today I incidentally found this page: http://www.cbel.com/ada_programming_language/

It has near 400 Ada links, some of them slightly unusual. Some of the links are incorrect; I did not found AdaPower among those links; but nevertheless this site, perhaps, deserves some attention (and correction -;) .

Ada in SEI's Software Technology Roadmap

From: Adrian Hoe <mailbox@adrianhoe.com>
Date: Fri, 29 Aug 2003 10:59:09 +0800
Subject: Re: Learn more on Ada
Newsgroups: comp.lang.ada

http://www.sei.cmu.edu/str/descriptions/ada83_body.html

http://www.sei.cmu.edu/activities/str/descriptions/ada95_body.html

Good information!

[From the home-page: "The Software Technology Roadmap (STR) is a directed guide containing the latest information on more than 69 software technologies. It is of interest to anyone acquiring, building, or maintaining software intensive systems." -- dc]

New AdaWorld Web Site

From: Stephane Richard <stephane.richard@verizon.net>
Date: Sat, 30 Aug 2003 18:23:50 GMT
Subject: Welcome to Ada World!
Newsgroups: comp.lang.ada

It's here. If you'd like to pay it a visit.

http://www.adaworld.com

Everything is there except: Projects, Code snippets, Tips and tricks, Forum.

So by all means, start sending, if you have anything in there. For the first 3, for the Forum, I'm working on it right now and hopefully will have it ready very soon.

Oh and to the one that said so, I hope it DOESN'T look like it was designed in the early nineties... It's been fun, and a lot more fun up ahead.

Feedback form will change too, right now I just wanted something that did the job .

Stéphane Richard, Senior Software and Technology Supervisor

From: Stephane Richard <stephane.richard@verizon.net>
Date: Wed, 24 Sep 2003 01:49:23 GMT
Subject: Mon nouveau Site Ada.
Newsgroups: fr.comp.lang.ada

[Translated from French. -- dc]

I have created my own site on the Ada language, and I would like to share it with you. The site is in English but if requests are strong enough, I would be pleased to offer a bilingual version. The first goal of the site is to promote Ada development projects and the many books and development tools available.

I request information on various Ada-related subjects, in particular, about active projects today to show how and in which context Ada is used presently. As you will see when visiting my site, I believe it is on the right track.

So welcome at Ada World (http://www.adaworld.com). Feel free to send me a message or links or projects to be added to make my site as complete as possible.

Stéphane Richard, "Ada World" Webmaster

On Ada Standardization and Ada 200Y

From: Randy Brukardt <randy@rrsoftware.com>
Date: Thu, 4 Sep 2003 22:55:02 -0500
Subject: Re: Bases for the Design of a Standard Container Library for Ada
Newsgroups: comp.lang.ada

> Time is an issue here because the Ada 2005 process includes really near deadlines now. Particularly, the first deadline for the standard container library proposal is the end of this month (September 2003). [...]

To clarify this:

The end of September deadline is for the submission of issues/proposals from "non-invited" groups. (That is, the general public.) The basic idea is to stop looking at new ideas at that point and start deciding on exactly what will be in the Amendment.

There is a secondary deadline of the end of December for "invited" groups -- which means all proposals need to be submitted by then, or there is little chance that they would be included in Ada 200Y. That even includes Tucker.

The basic reason for the deadlines is the need to cut off input so that we can really finish a document in the intended timetable. If we got input forever, we'd never really have a chance to finish. [...]

From: Randy Brukardt <randy@rrsoftware.com>
Date: Fri, 5 Sep 2003 14:40:10 -0500
Subject: Re: Bases for the Design of a Standard Container Library for Ada
Newsgroups: comp.lang.ada

Submissions ought to be sent to the Ada-Comment mailing list (like all questions/comments on the standard). The address is ada-comment@ada-auth.org. The list is open to the public; if you make a submission, you ought to join the list; see http://www.adaic.org/standards/articles/comment.html for details.

[See also "Comments on the Ada Standard" in AUJ 22.3 (Sep 2001), p.136. -- dc]

From: Vincent Tourvieille <tourvieille@shaw.ca>
Date: Mon, 08 Sep 2003 19:24:04 -0700
Subject: Re: Nouvelle norme Ada future ?
To: Marc Colinet <m.colinet@9online.fr>, ada-france@ada-france.org

[Translated from French. -- dc]

> Much information refers to a new Ada standard which could appear or in any case would be under discussion. What is really happening?

For the discussion, you could visit: http://ada-auth.org/arg-minutes.html

From: Clyde Roby <roby@ida.org>
Date: Wed, 8 Oct 2003 10:03:26 -0400
Subject: Re: APIWG Inquiries
To: SIGAda-APIWG@acm.org

[Extracted from a response to questions and remarks on the Ada standard and the ISO standardization process: -- dc]

URLs of websites that you might want to visit include:

http://www.sigada.org/ -- SIGAda's home page offers links to copies of the latest standard and rationale.

http://www.sigada.org/wg/apiwg/ -- APIWG is less than a year old; the home page currently does not offer a whole lot of information but what is there might be useful to you.

http://www.ada-europe.org/ -- similar to SIGAda in scope, Ada-Europe offers information about the use of Ada in Europe's Ada communities.

http://www.dkuug.dk/JTC1/SC22/WG9 -- the WG9 web page also offers links to what the group is doing and what is currently being discussed at its meetings.

http://www.adaic.org/ARA/ -- the Ada Resource Association has members who are Ada vendors or are Ada compiler vendors; they work together to "extend" the Ada language with useful additions in a much faster way than the standardization process does.

http://www.ada-auth.org/ -- the Ada Conformity Assessment Authority replaced the USDoD Validation Office to make sure that Ada compilers conform to the Ada language standard. [...]

Clyde Roby, ACM SIGAda Secretary, Acting Chair APIWG

From: Tucker Taft <stt@sofcheck.com>
Date: Wed, 8 Oct 2003 11:46:19 -0400
Organization: SofCheck, Inc.
Subject: Re: APIWG Inquiries
To: SIGAda-APIWG@acm.org

[...] Standardization is an admittedly slow process, because it is fundamentally built on consensus. On the other hand, there is nothing precluding a group of users establishing their own reusable libraries, and making them widely available. De-facto standards are often just as good, or even better, than de-jure standards. Unfortunately, a number of people have concluded that if something isn't in the Ada reference manual, it isn't "really" safe to use in "portable" code. But this is like trying to amend a constitution every time someone wants to pass a new piece of legislation. Changes to the reference manual are necessarily and appropriately slow in coming. Changes to a web site of shared code (such as could exist on http://www.adaworld.com) can be daily occurrences.

In any case, if you want insight into the ongoing work on updating the Ada standard, look at http://www.ada-auth.org -- you will find the minutes of the once-every-four-months Ada Rapporteur Group meetings, as well as all of the "Ada Interpretations" (AIs) which represent the ongoing work items of the ARG. [...]

From: Erhard Ploedereder <ploedere@informatik.uni-stuttgart.de>
Date: Wed, 8 Oct 2003 17:57:12 +0200
Subject: Re: APIWG Inquiries
To: SIGAda-APIWG@acm.org

I can add a couple of information bits....

The actual text of the Ada standard can be found on www.ada-auth.org in a number of formats. (Similarly, all the working documents for the 2005 revision are there)

There are two ways to get APIs standardized by ISO:

a) as part of the language RM -- in this case you are stuck with the five-year cycle of the RM for a number of administrative reasons.

b) as secondary standards -- they can be submitted at any time via WG9 as a document separate from the RM (but they then enter a similar cyclic life)

There is no reasonable way to get a more frequent revision cycle on ISO standards. It's possible in theory, but not in practice. In a way, ISO is right about this: an API isn't really an international standard, if it isn't stable over at least a five year cycle.

My personal opinion: more than ISO-standardized APIs, we need A SINGLE PLACE on the net, where a set of reasonably mature APIs is organized and maintained.

From: James W. Moore <moorej@mitre.org>
Reply-To: James.W.Moore@ieee.org
Date: Wed, 8 Oct 2003 12:47:13 -0400
Organization: The MITRE Corporation
Subject: Re: APIWG Inquiries
To: SIGAda-APIWG@acm.org

[...] I will elaborate a bit on the mechanism of Ada standardization.

Like all international standards, the Ada language standard is maintained by a Working Group (ISO/IEC JTC1/SC22/WG9) that is populated by representatives of "national bodies", i.e. countries. Currently we have participation from Canada, France, Germany, Italy, Japan, Russia, Switzerland, UK, and USA.

The process of international standardization is not inherently slow. The "speed of light" for an international standard is about 12 months. We have actually processed an Ada-related standard in 13 months. As with any other process that depends on consensus, the period of time to reach agreement is the determining factor. In the case of Ada, we avail ourselves of all mechanisms to accelerate work. Our track record indicates that we are highly effective in standardizing items quickly ONCE AGREEMENT IS REACHED.

In order to reach agreement quickly, we use informal committees, called Rapporteur Groups, which study issues and draft material for standardization. For example, we are currently processing a "Guide to the Use of the Ravenscar Profile". Our Annex H Rapporteur Group, working in collaboration with the University of York, reached agreement on its content in about 15 months. Approving the result as an ISO/IEC Technical Report will require another 12 or so months. (Actually, we could have gone faster but we are leaving open a window for possible last moment improvements.)

It is not correct to say that our process for revising the language takes ten years. Actually, we operate on a five-year cycle for that. We completed a Corrigendum to the standard (to deal with small problems) in 2000 and will complete an Amendment (to deal with larger problems) in 2005. It is our choice to move in a deliberate fashion for the underlying language standard because WG9 collectively believes that stability of language definition is very important in the Ada community.

Now, it is possible that you might disagree with WG9 on the respective value of stability versus change. If so, it would be important for you to participate in the process by which your particular nation takes positions on these issues, so that your opinion can be represented in the work of WG9. Most nations have relatively open processes by which you might participate. In the case of the US for example, a Technical Advisory Group (TAG) administered by the IEEE develops US positions and selects US delegations to WG9 meetings; the US TAG is open to membership by any US-domiciled organization. Other nations have similar processes.

I'm sure that any of the national organizations would be eager to have additional participation. I would be happy to provide contact information for any of them.

Regards, Jim Moore, Convener, ISO/IEC JTC1/SC22/WG9

James W. Moore, The MITRE Corporation, 7515 Colshire Drive, H505, McLean, VA 22102-7508, James.W.Moore@ieee.org

Comp.lang.ada mailing list

From: sk <sknipe@ktc.com>
Date: Thu, 25 Sep 2003 18:25:21 -0500
Subject: Re: Impossible to communicate via email
Newsgroups: comp.lang.ada

> Looks like the comp.lang.ada mailinglist isn't anymore:

Here is the new c.l.a info from the (working) subscription list distribution/digest ...

- comp.lang.ada mailing list

- comp.lang.ada@ada-france.org

- http://www.ada-france.org/mailman/listinfo/comp.lang.ada

[See also "More Ways to Access Comp.lang.ada Newsgroup" in AUJ 22.2 (Jun 2001), p.70. -- dc]

Ada Page on HitWeb Site

From: Brian Fraval <brian@fraval.org>
Date: Thu, 16 Oct 2003 01:58:21 +0200
Subject: Categorie Ada dans l'annuaire hitweb
Newsgroups: fr.comp.lang.ada

[Translated from French. -- dc]

I have just added a category "Programmation/Ada" in my personal directory. I included the sites quoted in the preceding messages... but I may have forgotten quite a few... I let the authors register their sites... (click on "add link")

http://hitweb.org/index234.html

[From the description: "HitWeb is a web site providing people with a powerful dictionary gathering a very thoroughly ranked set of internet addresses. Unlike web sites search engines, hitweb is based on a human intervention." -- dc]

Brian Fraval, http://brian.fraval.org/

AdaIC News Items

From: AdaIC Technical Webmaster <webmaster@adaic.com>
Date: Mon, 27 Oct 2003 20:33:56 -0600
Subject: [AdaIC] Two new Ada tools
To: <announce@adaic.com>

Two new Ada development tools were recently announced.

First, McKae Technologies' introduced DTraq. DTraq is a data logging and playback debugging tool providing near real-time data logging and analysis to aid debugging and validation. Captured, or 'tapped' data from a program can be viewed live while the program is running or, since it is being logged to a file, played back or printed out later for off-line review and analysis. DTraq handles all data conversion automatically by scanning the application's source code, identifying tapped data items, and extracting the information it needs to properly convert and display the logged items-simple scalar items as well as arrays and records. When the layout of data items change, rescanning automatically picks up the changes.

[See also "McKae Technologies - First Public Release of DTraq Data Logging and Playback Debugging Tool" further in this AUJ issue. -- dc]

Second, Excel Software's WinA&D tool has been upgraded to support Ada 95. WinA&D is a comprehensive software engineering tool for structured analysis and design, OOA/OOD with UML, multitask design, data modeling, requirements management, code generation for popular programming languages and a built-in scriptable reporting engine. When Ada is selected in WinA&D 3.5, tailored UML class models allow the designer to easily represent Ada packages and various types of relationships like aggregation (parent/child and parent/nested packages), generalization (generic instantiation) and dependency (withed packages). The UML modeling experience has been tailored to fit the conventions, standards and practices used by Ada developers. Ada code generated from the model is an exact implementation of the design including full package specifications and body files, subprogram files with arguments, data types and stubbed implementations.

[See also "Excel Software - WinA&D Design Tool Adds Ada Modeling and Code Generation" further in this AUJ issue. -- dc]

You can access both company's web sites (and the web sites of many other Ada development tools vendors) from the AdaIC's tools page: http://www.adaic.org/links/tools.html.

Randy Brukardt, Technical Webmaster, AdaIC.org/AdaIC.com

Ada-related Tools

Charles - Container Library

From: Matthew Heaney <matthewjheaney@earthlink.net>
Date: Wed, 30 Jul 2003 13:16:31 GMT
Subject: Re: Non-philosophical definition of Eiffel?
Newsgroups: comp.lang.ada

> What about this statement: "In general, in the design of Charles I have been willing to trade type-safety for flexibility and efficiency."

What that refers to specifically is the case of "dangling iterators."

If I do this:

```
procedure Op (C : in out
              Container_Type;
              E : in out
              Element_Type) is
  I : Iterator_Type;
begin
  Insert (C, E, I);
  Delete (C, I);
  E := Element (I); -- dangling
                    -- reference
end;
```

The model in Charles is that an iterator is implemented as a pointer to an internal node of storage, and it therefore confers no safety benefits beyond what a plain access type gives you.

To get a completely safe iterator -- one that prevents a dangling reference from ever occurring -- it is necessary to either reduce flexibility or reduce efficiency. But as a library designer, I am in no posi-

tion to decide how best to make that trade-off -- only the application developer can know that. Therefore, my philosophy has been to provide the most flexible and efficient primitives possible, which can then be combined as desired by the library user.

We are in agreement that all this business in Java where you have to perform a downcast when you extract an element is not type safe. But this is hardly the case in the STL and in Charles, which provide containers that really are type safe, unlike Java.

As for a container-of-D being a subclass of container-of-B, this model doesn't apply to STL or Charles, because those libraries eschew inheritance in favor of alternate (and simpler) mechanisms.

By using an iterator and a generic algorithm, the container itself disappears. So instead of a container-of-D, you have a sequence-of-D, which is a sequence-of-B, which is the equivalent of a container-of-B. No inheritance is necessary, thank you very much.

I'll have another release of Charles ready in the next few days.

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

From: Matthew Heaney <matthewjheaney@earthlink.net>
Date: Thu, 31 Jul 2003 13:10:03 GMT
Subject: Re: XML DOM Binding for Ada 95 - matter of style
Newsgroups: comp.lang.ada

> I thought one of the problems with [the Ada95] Booch [Components] was that it was using tagged types too much. Does Charles also use this approach?

One problem [with the Ada95 Booch Components] is that you have to make two instantiations: one for the root package, in which the root type is declared, and another, for the child package in which the actual container (derived) type is declared. This is a royal pain.

Charles deliberately avoided this technique, because no inheritance is necessary. Static mechanisms are used instead, which are simpler and more general.

So don't use a tagged type hierarchy when there's a simpler mechanism available. [...]

I'm not against tagged type ("object-oriented") programming, it's just that many programmers seem to treat this as a hammer looking for a nail, and automatically use a tagged type to solve a problem that is better solved other ways.

Look for a new release of Charles early next week. [...]

From: Matthew Heaney <matthewjheaney@earthlink.net>
Date: Sat, 02 Aug 2003 03:40:54 GMT
Subject: Re: XML DOM Binding for Ada 95 - matter of style
Newsgroups: comp.lang.ada

[...] The purpose of tagged records is to implement dynamic binding.

If the problem calls for dynamic binding, then maybe a tagged type is appropriate (but maybe not). If the problem does not require dynamic binding of operations, then you probably don't need tagged types.

> To implement [an XML DOM binding] with discriminated records (a feature I always found to have too many "gotchas!" built into it anyway) seems awkward and clumsy. If you don't think using tagged records to implement something like this design is appropriate, what are you saving them for?

I have never had a problem with discriminated records, so I don't know what "gotchas" you're referring to.

Tagged types are one tool among many. Use them when they make sense.

It probably doesn't make sense to use a tagged type when there is no dynamic binding.

Even when there is dynamic binding, using case statements or subprogram pointers is often simpler. It depends on the problem.

Note that the Charles library implements the full view of the container type as tagged, but only to add controlledness to the type, so that memory management is automatic. The partial view of the type is not tagged, and there is no dynamic binding. [...]

From: Matthew Heaney <matthewjheaney@earthlink.net>
Date: Fri, 05 Sep 2003 05:17:05 GMT
Subject: Re: Bases for the Design of a Standard Container Library for Ada
Newsgroups: comp.lang.ada

[About submitting proposals for an Ada container library to the ARG (see also "On Ada Standardization and Ada 200Y" earlier in this AUJ issue). -- dc]

The ASCLWG proposal will definitely be submitted this month -- it should be finished in a couple of weeks.

[See also "Ada Standard Containers Library Working Group" in AUJ 24.2 (Jun 2003), pp.75-76. -- dc]

The proposal is based on the Charles algorithms and container library, which is available from my home page.

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

[And from a later message: -- dc]

> You think you will be successful with your proposal?

Well, I don't know -- it's up to the ARG. I hope so. I think Charles is at the right level of abstraction, and satisfies the goals that a library should be safe, easy-to-use, flexible, and efficient.

One thing Charles has going for it is that it's modeled on the C++ STL, which has emerged as the de-facto standard by which other libraries are measured. The STL is a very, very good container library, and there is absolutely no reason why the STL can't be written in Ada95.

Realize of course that even though Charles is modeled on the STL, it is not a literal translation of the C++ version. Charles is first and foremost an Ada library.

From: Matthew Heaney <matthewjheaney@earthlink.net>
Date: Fri, 26 Sep 2003 00:25:41 GMT
Subject: Re: who is in charge of Ada 0Y?
Newsgroups: comp.lang.ada

> As far as I've read, there is no organization planning changes for Ada 0Y. In such case, Ada95 is as petrified as Latin, and so it's doomed. Have I missed something? is there such organization?

I don't know whether Ada95 is as petrified as Latin, but the language is actively being maintained under the stewardship of the ARG (Ada Rapporteur Group).

In fact I just submitted a proposal for a standard container library for Ada 0X to the ARG this very afternoon!

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

PragmARC - PragmAda Reusable Components

From: Jeffrey Carter <jrcarter@acm.org>
Date: Thu, 7 Aug 2003 18:49:36 -0700
Subject: Release of PragmAda Reusable Components
To: team-ada@acm.org

PragmAda Software Engineering announces a new release of the PragmAda Reusable Components. This release corrects errors and improves some components.

The PragmARCs are available from

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

The mirror at www.adapower.com will be updated soon, we hope.

[See also same topic in AUJ 23.4 (Dec 2002), p.192. -- dc]

Jeffrey R. Carter, PragmAda Software Engineering

Design of a Standard Container Library for Ada

From: maa@liacc.up.pt (Mario Amado Alves)
Date: 2 Sep 2003 11:59:11 -0700
Subject: Bases for the Design of a Standard Container Library for Ada

Newsgroups: comp.lang.ada

I'm submitting this document to the Ada community as a request for comments:

"Bases for the Design of a Standard Container Library for Ada ... an attempt to put together a complete, consistent, and correct set of bases for the design of a standard container library for Ada."

Please find the full 130 paragraph long document in

http://www.liacc.up.pt/~maa/bases_1.txt (public access) or in http://groups.yahoo.com/group/asclwg/files/Bases (might require Yahoo! login)

The document includes a Bibliography, and instructions on how to convey specific comments. Of course general comments are welcome as well as immediate replies right here.

Grace - Ada Library

From: Stephen Leake <Stephe.Leake@nasa.gov>
Date: 27 Oct 2003 11:56:57 -0500
Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)
Subject: Grace 0.51 released
Newsgroups: comp.lang.ada

I've made a new release of Grace, _the_ community Ada library . Config_Files has been improved. See http://savannah.nongnu.org/projects/grace/.

[See also "Grace.Config_Files" in AUJ 24.3 (Sep 2003), p.145, and "Platform Independent Access to Configuration Information" in AUJ 24.1 (Mar 2003), p.16. -- dc]

Ada Container Library Projects

From: Nick Roberts <nick.roberts@acm.org>
Date: Tue, 28 Oct 2003 00:17:45
Subject: Re: Grace 0.51 released
Newsgroups: comp.lang.ada

> If we do get a community library going, let's at least use the name Grace (for Grace Hopper), and not CAL or whatever.

Or alternatively, let's call it 'Charles', or maybe 'ASCL', or 'Booch', or 'PragmARC', or 'GAPSE', or 'AdaSL', or 'SAL', or 'Tenet', or ...

In fact, my guess is that it's going to be called 'Ada'. I believe the ARG are interested in introducing a basic set of containers into the next revision, [...] If you're interested, please look at AI-302: http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-10302.TXT

This proposal is huge, but it is currently undergoing revision to significantly reduce it. I think the proposal, submitted by Matthew Heaney and based on Charles, is basically very good. [...]

Links to some of the projects I mentioned are:

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

http://www.adapower.net/booch/

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

http://adasl.sourceforge.net/

http://homepage.ntlworld.com/ramatthews/ [for GAPSE]

http://ascl.sourceforge.net/

http://tenet.berlios.de/

[More information on these projects is included in virtually every AUJ issue. -- dc]

GNAT for Gentoo GNU/Linux

From: David Holm <david@realityrift.com>
Date: Wed, 27 Aug 2003 15:29:34 GMT
Subject: GNAT 5.0 for the curious
Newsgroups: comp.lang.ada

Gentoo GNU/Linux (http://www.gentoo.org/) now offers CVS snapshots of GNAT 5.0 if anyone is interested in testing it. They are masked since GNAT 3.15p is still the stable release, but just remove the gnat entry in /usr/portage/profiles/package.mask and you will be able to install it. I only have a few tests here (that fail with gnat in gcc 3) and they worked with GNAT 5.0. Anyway, if you are using gnat from gcc 3 you will definately want to ditch that and use the snapshot as it contains the uncrippled gcc patch along with much newer Ada-code. I will try to update the snapshot on a regular basis until 5.0 is released.

At the moment it is not possible to have multiple versions of GNAT installed in Gentoo. I'm working on this since then you would be able to have both 3.15p and 5.0-snapshots. I'm currently prioritizing porting it to other platforms though (ppc to begin with as I bought a Pegasos yesterday).

David Holm, Gentoo dev-team

[And from a later message: -- dc]

> Just curious: where did you get it? I wasn't aware that ACT released GNAT 5.0 to the public. [...]

Um, from libre.act-europe.fr/gnat. The CVSROOT is on the bottom half of the page.

From: Lionel Draghi <lionel.draghi@free.fr>
Date: Thu, 11 Sep 2003 00:13:15 +0200
Subject: Re: Une question sur gnat
To: ada-france@ada-france.org

[Extracts translated from French: -- dc]

> I just started a programming course using the Ada language. My problem is that I would like to use GNAT on my main computer. However, this computer, an Apple ibook, runs Linux, in fact Gentoo Linux 1.4. [...] I cannot use the binaries available one the official GNAT site and I cannot use RPM because my distro does not support them. My question is simple: how to compile GNAT in this case?

See http://dev.gentoo.org/~dholm/ada.html.

And there are quite a lot of Ada packages available, see http://csociety-ftp.ecn.purdue.edu/pub/gentoo-portage/dev-ada/. You should be able to put together a nice environment.

GNAT for Mac OS X

From: Alan B. Reynolds <alan1@actel.com>
Date: Wed, 27 Aug 2003 18:28:10 GMT
Subject: Re: Free Ada95 compiler for MacOS X
Newsgroups: comp.lang.ada

> Does anybody know of a simple package to get an Ada95 compiler on Mac OS X? I know versions are available for Unix so...

Try http://www.macada.org.

[See also same topic in AUJ 24.3 (Sep 2003), p.140. -- dc]

From: Adrian Hoe <mailbox@adrianhoe.com>
Date: Fri, 29 Aug 2003 10:52:59 +0800
Subject: Re: Free Ada95 compiler for MacOS X
Newsgroups: comp.lang.ada

And http://adrianhoe.com for an installation HowTo.

From: Jim Hopper <hopperj@macconnect.com>
Date: Thu, 23 Oct 2003 09:27:40 -0400
Subject: Re: www.gnuada.org maintainership
To: GNAT Discussion List <gnatlist@lyris.seas.gwu.edu>

For the Mac OS X stuff you can just provide a link to our OS X page at <http://www.macada.org> which is provided by David Botton.

The idea of having areas each managed by an individual is the way we have been doing it for the Mac site. For instance Drew and Al Reynolds are primary responsibility for the compiler port, Stephen Bespalko does GDB, Phillip Gonia is doing tutorials, I am doin bindings to Apple's Carbon Gui and example code. That's not to say that we don't all help out on other things: Andrew contributes to the bindings, and example code for instance and I work with Phillip on the tutorials, etc. This has worked out very well for us.

GNAT for Mac with Linux

From: Arnaud Rolly <arnaud.rolly@eikonex.net>
Date: Fri, 12 Sep 2003 14:39:20 +0200
Organization: Eikonex
Subject: Re: Une question sur gnat
cc: ada-france@ada-france.org

[Extracts translated from French: -- dc]

> It's the first time I hear talking about a Mac running Linux. To the best of my knowledge, all modern Macs either have a MacOS X system, which has a specific Unix kernel completely different from Linux, or MacOS 9. [...]

It's already quite a while possible to use Mac under Linux. The best suited distribution is Yellow Dog Linux (http://www.yellowdoglinux.com) of TerraSoft. It is interesting to note that TerraSoft has agreements with Apple for the support of the material. I deployed Linux on iMac (G3) and G4, it runs well: -)

Arnaud Rolly, Eikonex, Open Source Engineering, http://www.eikonex.net

From: Laurent Guerby <guerby@acm.org>
Date: Fri, 12 Sep 2003 21:04:05 +0200
Subject: Re: Une question sur gnat
To: Mathieu Leduc-Hamel <mleduc@savoirfairelinux.net>
cc: ada-france@ada-france.org

Apparently Ben Stanley managed to get GNAT running on ppc-yellowdog-linux [incl. tasking -- dc]: http://gcc.gnu.org/ml/gcc/2003-09/msg00287.html

Synchronization of FSF GCC with ACT GNAT

From: Mark Lorenzen <mark.lorenzen@ofir.dk>
Date: Tue, 28 Oct 2003 00:41:07 +0100
Subject: Good news regarding GNAT in FSF GCC tree
Newsgroups: comp.lang.ada

It seems that ACT will allocate resources to get the official FSF GCC tree up to date with the ACT tree. This is good news as there is supposed to be *a lot* of fixes in the ACT tree, that have not yet made it into the FSF tree. http://gcc.gnu.org/ml/gcc/2003-10/msg00833.html and msg00846.html

From: Preben Randhol <randhol@pvv.org>
Date: Wed, 29 Oct 2003 09:33:14 UTC
Subject: Re: Good news regarding GNAT in FSF GCC tree
Newsgroups: comp.lang.ada

Very good news! It would mean a lot if GNAT 3.x gets stable and can be used as it will be readily available on most major Linux distributions.

Fuzzy Sets for Ada

From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Mon, 25 Aug 2003 11:13:06 +0200
Subject: ANN: Fuzzy sets for Ada version 3.3
Newsgroups: comp.ai.fuzzy,comp.lang.ada

Fuzzy sets for Ada is free and distributed in source code.

http://www.dmitry-kazakov.de/ada/fuzzy.htm

[See also same topic in AUJ 23.3 (Sep 2002), p.132. -- dc]

Version 3.3 provides implementations of:

1. Confidence factors with the operations not, and, or, xor, +, *;

2. Classical fuzzy sets with the set-theoretic operations and the operations of the possibility theory;

3. Intuitionistic fuzzy sets with the operations on them;

4. Fuzzy logic based on the intuitionistic fuzzy sets and the possibility theory;

5. Fuzzy numbers both integer and floating-point ones with conventional arithmetical operations;

6. Linguistic variables and sets of linguistic variables with operations on them;

7. String-oriented I/O is supported.

AdaOpenGL 0.12 - Thin Ada OpenGL Binding

From: David Holm <david@realityrift.com>
Date: Tue, 12 Aug 2003 01:27:23 GMT
Subject: AdaOpenGL 0.12 released
Newsgroups: comp.lang.ada,comp.graphics.api.opengl

I just released AdaOpenGL 0.12. It includes bindings to WGL and GLFW and the GNAE support has been fixed. It has been synced with the latest version of gl.h, glx.h and glext.h. You no longer need a patched AdaBindX to use it with GLX.

As usual you can find it here: http://adaopengl.sf.net/

[See also "AdaOpenGL 0.10 Released" in AUJ 24.3 (Sep 2003), p.142. -- dc]

AdaGraph - High-Resolution Color Graphics

From: Jerry van Dijk <jdijk@acm.org>
Date: 01 Sep 2003 17:15:56 +0200
Subject: New, fixed, version of AdaGraph
Newsgroups: comp.lang.ada

For some time now AdaGraph was plagued by a Win2K/XP bug. This is now fixed.

You can download the new AdaGraph installer (version 0.5e) which now also includes some html documentation, at http://www.jvdsys.demon.nl

Jerry van Dijk, Leiden, Holland

GUI Programming Under OS2

From: Gregory Bourassa <bourassa@magma.ca>
Date: Thu, 31 Jul 2003 19:48:57 -0400 (EDT)
Subject: Re: Ada and OS2
Newsgroups: comp.lang.ada

> Has anybody experiences with GUI programming in Ada under OS2? Which GUI toolkit is available for using it under OS2?

On http://hobbes.nmsu.edu search in /pub/os2/dev/ada for adagraph.zip and os2apipm.zip. These are Ada bindings to the OS/2 APIs. Also in /pub/os2/dev/emx/contrib/gnat there is gnat-os2apipm.zip.

From: Georg Bauhaus <sb463ba@d2-hrz.uni-duisburg.de>
Date: Fri, 1 Aug 2003 14:04:15 UTC
Subject: Re: Ada and OS2
Newsgroups: comp.lang.ada

Visual Ada Developer is available for OS/2 too. [See further. -- dc]

[And from a later message: -- dc]

[...] maybe you could start with Tash from http://www.adatcl.com and Tcl/Tk from http://hobbes.nmsu.edu?

VAD 6.2 - Visual Ada Developer

From: Stas <dulman@attglobal.net>
Date: Wed, 1 Oct 2003 18:09:42 +0200
Subject: Announce: Visual Ada Developer (VAD) version 6.2. Leonid Dulman
Newsgroups: comp.lang.ada

[Extracts of announcement only. See also "VAD 6.1 - Visual Ada Developer" in AUJ 24.2 (Jun 2003), p.76. -- dc]

VAD 6.2 Common description.

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

VAD 6.2 has five realization: for tcl/tk 8.0.x, 8.2.3, 8.3.5 and 8.4.4 (last version). You need to install and to check tcl/tk first. [...]

VAD 6.2 is available in http://www.websamba.com/tcltk/vad.htm. You may dowload sources [...] and binaries [...] (Windows 9x/NT) [...] (Linux).

Any questions, any ideas, any problems, any help:

Leonid Dulman (dulman@attglobal.net)

XML4Ada95 - Binding to Xerces XML Package

From: Denny Vrandecic <nodix@tiscali.de>
Date: Wed, 30 Jul 2003 13:32:23 +0200
Subject: XML DOM Binding for Ada 95 - matter of style
Newsgroups: comp.lang.ada

I am writing an XML DOM Binding for Ada 95 (as part of my master thesis) - not really the whole parser, just a binding to a parser that is more complete (i.e. implements more modules of the DOM Spec) than the otherwise great XmlAda by ACT. The binding will be given to the community for free under a BSD-like license.

I'd love to write the binding in such a way, that later the user may easily switch to a native DOM Implementation (like a further developed XmlAda), which I'm sure will finally come. [...]

Denny Vrandecic, Student of computer science, University Stuttgart

From: Denny Vrandecic <nodix@tiscali.de>
Date: Wed, 15 Oct 2003 22:35:26 +0200
Subject: XML4Ada95 0.9 - released!
Newsgroups: comp.lang.ada

XML4Ada95 is an Ada 95 binding to the popular Xerces XML package written by the Apache Group and this is the first public release of the binding. The binding is released under the BSD license, Xerces is released under the Apache license, thus using this is absolutely free.

The binding offers a complete access to the DOM Level 1 and DOM Level 2 Core, and partial access to DOM Level 3 Core and DOM Level.

You can obtain more information on the following website: http://www.nodix.de/xml4ada95

You can obtain more information about Xerces at: http://xml.apache.org/xerces-c

I welcome any feedback, and will try to answer questions about the usage.

Imgsvr - Personal Picture Web Server

From: Patrice Freydiere <frett27@free.fr>
Date: Thu, 18 Sep 2003 12:26:24 +0200
Subject: imgsvr first release annoncement
Newsgroups: comp.lang.ada

I'm proud to announce the first release of Imgsvr 0.1, a personal picture web server based on the famous AWS/Template Parser components, implemented in the Ada language.

ImgSvr v0.1 supports FileBased JpegFiles thanks to JpegLib, Thumbnails generation and storage in a DBM Database, Custom browsing pages, and page layout inheritance.

Currently available on Linux platforms, but can be very easily recompiled on MS WXX.

http://imgsvr.tuxfamily.org

Binding to Lua 5.0 API

From: chris <chris.danx@ntlworld.com>
Date: Mon, 22 Sep 2003 00:36:50 +0100
Subject: [Ann] Ada 95 binding to Lua 5.0
Newsgroups: comp.lang.ada

I am pleased to announce the release of a portable binding to the Lua 5.0 programming api. Lua allows software developers to provide tailored scripting and extension facilities in their applications with a simple yet powerful programming API at virtually no cost (Lua is licensed under the terms of the MIT license). Now Ada 95 developers on a wide range of platforms can utilise Lua in their applications under the same terms as the Lua software (yes, that's right! It is licensed under the MIT license too).

A few days ago a highly experimental and featureless version of the binding was released, but with todays release (version 0.1) the binding includes most of the functionality of the C programmers API.

Notable exceptions are the debugging API and the operations on "user data" as defined by Lua. These are penciled in for 0.3/0.5 (hopefully soon).

This release includes:

- bindings to substantial portion of the api;

- two example programs adapted from the Lua book (a simple command line interpreter, an example of using Lua for configuring applications);

- a first attempt at a user manual.

Future releases will include: binding to the debugging api, "user data" facilities, binding to libaux facilities, example utilising Lua & AdaOpenGL, better documentation.

The binding can be found at www.cyberdanx.co.uk/ada95/lua.html.

Don't hesitate to email danx@cyberdanx.co.uk with queries, bug reports, patches, advice and suggestions!

[See also "Bindings for GUILE Interpreter" in AUJ 24.3 (Sep 2003), pp.142-143. -- dc]

Interfacing Ada with C++

From: Denny Vrandecic <nodix@tiscali.de>
Date: Wed, 15 Oct 2003 22:49:42 +0200
Subject: Interfacing Ada with C++
Newsgroups: comp.lang.ada

Just, if you're interested: while I was writing XML4Ada95 I had to solve some problems about interfacing Ada 95 with C++ without having it crash on exceptions and on I/O-streams. I have written down my experience and solutions with this in my Diplomarbeit (sorry, this is in German, but you may find the source code useful nevertheless).

Peek at: http://www.nodix.de/xml4ada95/interface.htm. Or ask me.

APQ 2.0 Thick Database Binding

From: Warren W. Gay VE3WWG <ve3wwg@cogeco.ca>
Date: Sun, 7 Sep 2003 23:45:40 -0400
Subject: APQ 2.0 is now Released (PostgreSQL/MySQL thick Binding)
Newsgroups: comp.lang.ada

[See also "APQ - PostgreSQL Ada95 Binding" in AUJ 23.4 (Dec 2002), p.195. -- dc]

At long last, the APQ 2.0 Thick Database Binding has now been released. See: http://home.cogeco.ca/~ve3wwg/software.html#APQ

What is APQ?

It is a thick binding to the database client libraries for PostgreSQL and MySQL (new). No ODBC driver or configuration is required. You can build APQ with any combination of the databases you want to support.

Also new is the concept of "Generic Database Programming". Using tagged objects and polymorphism, it is now possible to code your application in a database neutral way. See Chapter 8 "Generic Database Programming" for a description of this and a program example.

Why use APQ?

- Very simple to use (only 2 objects to use + 1 for blobs).

- Only Ada95 data types (no C language interfaces).

- Blob support (PostgreSQL only at this time).

- Full suite of generic functions/procedures for strongly typed programming.

- Date/time object support.

- No ODBC or other infrastructure to get going.

- Generic database programming is possible (database neutral code).

- Full support for NULL values (including strong typing).

- ACL or GPL2 dual license.

- Extensive programmer's reference manual (with examples).

- Soon to be available to win32 programmers.

Win32 builds are possible if you know what you're doing. I just haven't taken the time yet to document this procedure. A win32 binary release is planned, however.

APQ Forum at www.adaworld.com

From: Warren W. Gay VE3WWG <ve3wwg@cogeco.ca>
Date: Fri, 19 Sep 2003 13:06:29 -0400
Subject: Announce: APQ Forum Now Available at www.adaworld.com
Newsgroups: comp.lang.ada

I am making some progress in developing the win32 release for APQ-2.0, which is documented in updates to the APQ specific forum that was kindly provided at www.adaworld.com.

If you want to follow the status updates, or discuss APQ issues/ideas, then please check out the APQ site forum.

At the home page [see above -- dc], click on "Project's Forum" button at the left, to locate the APQ forum. Or just click on: http://www.adaworld.com/projectforum/

APQ 2.1 with Win32 Support

From: Warren W. Gay VE3WWG <ve3wwg@cogeco.ca>
Date: Wed, 24 Sep 2003 16:51:48 -0400
Subject: Announce: APQ-2.1 with Win32 Support Released
Newsgroups: comp.lang.ada

I am happy to announce, that APQ-2.1 with win32 support, is now available. It comes complete with a GUI self installer, thanks to the folks at Nullsoft Scriptable Install System (See http://nsis.sourceforge.net/site/index.php).

APQ is a client level Ada95 thick binding to Databases. At the present time, APQ provides native support for PostgreSQL and MySQL but other databases may be supported in the future.

The binary win32 APQ-2.1 release can be installed with a few mouse clicks by opening http://home.cogeco.ca/~ve3wwg/APQ-2.1.EXE

The installer is "GNAT aware", and will install APQ along side GNAT's Win32Ada binding. As a result, your APQ client programs will automatically link with it as required.

An uninstaller is included, for those who just want to give it a whirl.

A test program win32_test.adb is included in the above install. The test program only requires you to: 1. choose database support by uncommenting "with" statement; 2. edit in your account and password; 3. edit in your database name; 4. gnatmake win_test; 5. .\win_test

This program will create a table TEST_TBL, insert a few rows, and then perform a select on the table.

A full APQ-2.1 source release is also available from: http://home.cogeco.ca/~ve3wwg/apq-2.1.tar.gz

This includes a win32.pdf file with instructions for building the win32 release. The APQ pdf manual is also included.

APQ-2.1 Unix/Linux Enhancement: You no longer are required to provide library linker arguments. [...] The win32 port works the same way, thanks to GNAT pragmas.

Don't Forget the www.adaworld.com Forum for APQ. Thanks to Stéphane Richard there is a small forum at www.adaworld.com for comments, suggestions and questions related to APQ. From the main page, click on "Projects' Forum" (at left) to participate.

From: Warren W. Gay VE3WWG <ve3wwg@cogeco.ca>
Date: Sat, 11 Oct 2003 12:01:38 -0400
Subject: Announce: APQ-2.1 Win32 Binaries Install for GNAT 3.15p now Available
Newsgroups: comp.lang.ada

Due to the fact that the *.ali file format differs between GNAT 3.14p and 3.15p, I have made win32 binaries available for GNAT-3.15p users. The install includes a sample win32_test.adb program for your own testing pleasure, and a complete PDF manual for the APQ API.

Win32 users can compile APQ from sources, but it requires a number of tools, including the Microsoft C compiler. Additionally, there are a number of pitfalls that make the process less than elegant. For this reason, win32 users are encouraged to install the APQ binaries, using the GUI installer provided. The installer includes an uninstaller, for those that just want to test drive APQ. Linux/Unix users will find it easy to compile and install from sources.

Visit http://home.cogeco.ca/~ve3wwg and click on the APQ link that is prominently displayed there for more details and downloading.

Also, checkout the APQ in the www.adaworld.com "Projects' Forum" for further discussions about APQ and future developments.

What is APQ?

APQ is a thick Ada95 binding to the native drivers for PostgreSQL and MySQL database products (no ODBC required). Using an OO framework, you can easily write code that performs SQL operations on your favourite database platform. A complete PDF manual is provided. Almost all API operations include an example program fragment. [...] The next release will include support for Sybase Adapative Server Enterprise 12.5.

ODB - Object Persistency Framework

From: Michael Erdmann <Michael.Erdmann@snafu.de>
Date: Sat, 13 Sep 2003 14:04:19 +0200
Subject: Release of ODB 0.5
Newsgroups: comp.lang.ada

As part of the GNADE (http://gnade.sourceforge.net/ the object persistency framework ODB 0.5 is released.

The software allows you to store and retrieve Ada 95 objects. Please refer to the documentation on the home page.

This release is a prerelease with the following restrictions: the interface to the underlying storage media is not yet stable; the software has only been tested on Linux i686 architecture.

From: Michael Erdmann <michael.erdmann@snafu.de>
Date: Fri, 17 Oct 2003 21:59:18 +0200
Subject: Release of ODB 0.6.2
Newsgroups: comp.lang.ada

The Version 0.6.2 of ODB is available at: http://sourceforge.net/projects/gnade

This SW package provides means of implementing object persistency with Ada 95. Since the code of this project started as an spin of of some other work, this is still a development release and comments or requirements are welcome!

Minimal Run Time for GNAT

From: sk <sknipe@ktc.com>
Date: Fri, 29 Aug 2003 00:21:25 -0500
Subject: Re: porting/recompiling GNAT run time library
Newsgroups: comp.lang.ada

> Has someone made a minimal custom run time library for use with GNAT No_Run_Time pragma?

No runtime, but a standalone "pragma No_Run_Time" ELF [executable] booting from [the] GRUB [boot loader] with no GNAT dependencies.

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

Directly download one of the above and ignore the web page, it is not current.

[See also "Booting an Ada Main Program" in AUJ 24.2 (Jun 2003), p.79. -- dc]

From: jeff.huter@bigfoot.com (Jeff)
Date: 29 Aug 2003 06:14:47 -0700
Subject: Re: porting/recompiling GNAT run time library
Newsgroups: comp.lang.ada

Create a "gnat.adc" that contains:

pragma No_Run_Time;

pragma Restrictions(No_Exceptions);

This has worked for me in the past.

Cheddar - Real-Time Scheduling Simulator

From: Frank Singhoff <singhoff@beru.univ-brest.fr>
Date: 4 Sep 2003 13:28:42 GMT
Organization: Universite de Bretagne Occidentale
Subject: ANN: New release of Cheddar, a real time scheduling simulator
Newsgroups: fr.comp.lang.ada

The EA 2215 team is pleased to announce a new release of Cheddar, a free real time scheduling simulator.

[See also same topic in AUJ 23.4 (Dec 2002), pp.198-199. -- dc]

Cheddar is a free real time scheduling tool. Cheddar is designed for checking task temporal constraints and buffer sizes of a real time application/system. It can also help you for quick prototyping of real time schedulers. Finally, it can be used for educational purposes.

Cheddar is composed of two independent parts: an editor used to describe a real time application/system, and a framework. The editor allows you to describe systems composed of several processors which own tasks, shared resources and buffers. The framework includes feasibility tests and simulation tools. Feasibility tests can be applied to check that task response times are met and that buffer size are bounded. When feasibility tests can not be applied, the studied application can be analysed with scheduling and buffer simulations.

The current release is now 1.3p1. Cheddar is distributed under the GNU GPL license. It's free software, and you are welcome to redistribute it under certain conditions; see the GNU General Public License for details. Source code, binaries and documentations can be freely downloaded from http://beru.univ-brest.fr/~singhoff/cheddar

Cheddar is written in Ada with GtkAda. It runs on Solaris, Linux and win32 boxes and should run on every GNAT/GtkAda supported platforms (see ACT web site for details).

1) With Cheddar, you can:

Do scheduling simulations with classical real time schedulers: Rate Monotonic, Deadline Monotonic, Least Laxity First, Earliest Deadline First, POSIX.4 queueing policies: SCHED_OTHERS, SCHED_FIFO and SCHED_RR) with different type of tasks (aperiodic, periodic, task activated with a poisson process law, ...

Extract information from scheduling simulation: buffer utilization factor, task response times, task missed deadlines, number of preemption, ...

Apply feasibility tests on tasks or buffers (without scheduling simulation): compute task response times, apply processor utilization test, schedule for a given base period, compute bound on buffer size (when buffer are shared by periodic tasks).

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

Tools to express and do simulations/feasibility tests with task precedencies: schedule tasks according to task precedencies, compute Tindell end to end response time, apply Chetto and Blazewicz algorithms.

Do simulation when tasks are randomly activated.

2) The most important new features are:

- Cheddar project files are now saved in a XML format. Project can then be shared between several platforms and tools.

- Add response time feasibility tests for RM/DM/POSIX.4 in the non preemptive case. Add EDF/LLF response time feasibility tests in the preemptive and the non preemptive case.

- Add response time from scheduling simulation.

- When Cheddar crashes, the ongoing project is saved in a XML file called "cheddar_bug.xml" and then, project editing is not lost any more.

- Add parametric schedulers.

- Produce simulation results in string or XML format.

- Add some classic Queueing results in the framework to help buffer analysis.

- Add a C inferface to the Ada framework.

- Add some tools to do random scheduling simulation.

- Fix some bugs (see BUGS file).

3) Work in progress:

During the next year, we plan to improve the tool with the following features:

- Update the user's guide.

- Improvement of the buffer analysis features.

- Add feasibility tests and simulation features for multi-processors systems.

- Rebuild of shared resources support: adding new protocols and add parametric shared resource protocols.

Feel free to contact us for help or bugs report.

OS Options for Real-Time Ada

From: James Rogers <jimmaureenrogers@att.net>
Date: Fri, 24 Oct 2003 21:05:32 GMT
Subject: Re: Realtime and Ada - stupid newby question
Newsgroups: comp.lang.ada

> I am planning a realtime application for data acquisition on a pc104 platform and investigating my OS options. I'm normally a Delphi programmer, and rather than the pain of boning up on my very poor C/C++ skills, I would like to learn Ada for this project. Can someone please point me to some links which outline the pros and cons of the various OS's. I'm sure there are numerous I haven't heard of yet.

A few years ago I lead a robotic control development effort on a pc104 system. We looked at several good OS's. At the time the OS's with good Ada support were Multi from GreenHills, PharLap bundled with Aonix Object Ada, and VxWorks used with Object Ada or Gnat.

We chose the PharLap/Object Ada combination for a number of reasons including licensing costs. I believe any of these operating systems offer the features you are looking for.

> Also any background info I should read before starting in on Ada would be great.

You will want to read some of the online articles and books at www.adapower.com before starting. Ada looks a lot like Pascal, but has a lot of features that are different from Pascal. In particular you will want to study the Ada visibility rules.

Depending upon your design, you also may want to exercise some of the tasking capabilities of Ada. We certainly used them extensively in the robotic system. We also used generics, streams, and tagged types.

Jim Rogers

From: Martin Dowie <martin.dowie@btopenworld.com>
Date: Fri, 24 Oct 2003 21:15:42 UTC
Subject: Re: Realtime and Ada - stupid newby question
Newsgroups: comp.lang.ada

> [A few years ago] the OS's with good Ada support were Multi from GreenHills, PharLap bundled with Aonix Object Ada, and VxWorks used with Object Ada or Gnat.

Small correction - "Multi" is the Green Hills IDE - it supports numerous OS, including their own "Integrity" and Wind River's VxWorks.

From: Randy Brukardt <randy@rrsoftware.com>
Date: Fri, 24 Oct 2003 16:45:33 -0500
Subject: Re: Realtime and Ada - stupid newby question
Newsgroups: comp.lang.ada

> Is Pharlap an RTOS? My quick reading of their website seemed to indicate that it was a DOS extender that made DOS "more" realtime. Admittedly that may be real enough for my task.

It originally was a DOS Extender, but later on there were bare-machine versions

(that is, an RTOS). I have no idea what the current state is, though.

[From: Ed Falis <falis@verizon.net>]

As of the last time I looked, maybe 3 years ago, their Embedded ETS product was definitely an RTOS, and a nice lean one at that (disclaimer: I used to work with them when I was at Aonix). The DOS-Extender is an older product, that I believe was their original. But they have been building the embedded OS for quite some time now (~10 years).

[From: James Rogers <jimmaureenrogers@att.net>]

When I worked with it we used the ETS OS, which is clearly an RTOS. It has a very nice feature. The API is a subset of the Win32 API. This means you can do a lot of unit testing on your development PC before recompiling for the target. This helps speed development and lowers contention for the development target system.

In 1999 PharLap was working on adding some optional GUI extensions to ETS. I never used that version.

I liked ETS because we could tailor the complexity and size of the RTOS. We could use a bare-bones kernel requiring about 6K of memory, or we could add features such as a full file system, a TCP/IP stack, and several other interesting drivers. The file system could support disk-on-chip implementations so that you could treat a ramdisk as though it was a hard drive. We used a 40Mb ramdisk as though it was a Win32 file system on a disk drive, including the ability to define directories as well as files.

The amount of stack space, and the number of tasks supported by ETS was limited only by the amount of memory on your system. The OS did not provide hard limits to the number of tasks.

From: Jeff C <jcreem@yahoo.com>
Date: Sat, 25 Oct 2003 13:37:58 GMT
Subject: Re: Realtime and Ada - stupid newby question
Newsgroups: comp.lang.ada

> Of particular interest is availability of device drivers for the various OS's. We need a/d converters, serial cards, gps cards... the usual stuff.

If you need the largest selection of device drivers then I would consider Linux (not really an RTOS, if your requirements are soft RT you can probably get by with this), vxWorks (GNAT available, Rational Apex Ada available, Greenhills multi available), RT Linux (GNAT for this available http://rtportal.upv.es/apps/rtl-gnat/).

Also take a look at each of the offerings from http://www.lynuxworks.com/

From: Colin_Paul_Gloster@acm.org (Colin Paul Gloster)
Date: 28 Oct 2003 18:45:26 GMT
Organization: Dublin City University
Subject: Re: Realtime and Ada - stupid newby question
Newsgroups: comp.lang.ada

Read up on Ada 95's support for interfacing to C so that you will be able to use a wide range of device drivers.

Design Patterns and UML Mappings

From: Xaelis <Alexis.Muller@lifl.fr>
Date: Mon, 22 Sep 2003 13:17:08 +0200
Organization: Universite des Sciences et Technologies de Lille, France
Subject: Re: Design patterns (Modèles de conception)
Newsgroups: fr.comp.lang.ada

[Translated from French: -- dc]

> Can somebody tell me if there is literature (French or English) concerning Ada solutions for patterns such as Singleton, Factory and others (those of GoF in a general)?

I think that will make you happy:

http://www.acm.org/sigs/sigada/wg/patterns/patterns/GOF/source/ada/

http://www.acm.org/sigada/wg/patterns/

[See also "Design Patterns in Ada95" in AUJ 20.3 (Oct 1999), p.177, AUJ 20.2 (Jul 1999), pp.102-104 and AUJ 20.1 (Apr 1999), pp.12-14. -- dc]

> Is there literature about to the/a mapping UML -> Ada?

Perhaps ["Mapping the UML Notation to Ada 95" -- dc]: http://www.cs.hmc.edu/tech_docs/qref/rational/DevelopmentStudioUNIX.2000.02.10/docs/html/rose_ada95/1mapping.html

If you finds anything else, that interests me too.

[See also "UML to Ada Working Group" in AUJ 24.3 (Sep 2003), p.144, and "Ada-UML Profile" in AUJ 24.2 (Jun 2003), pp.79-80. -- dc]

Alexis Muller, Laboratoire d'Informatique Fondamentale de Lille (LIFL), Universite de Lille, France, http://www.lifl.fr/~mullera

GNU Ada-mode for Emacs 21

From: Jérôme Haguet <jeje66@despammed.com>
Date: Wed, 30 Jul 2003 09:44:37 +0200
Subject: Re: Need help w/ ada-mode.el under Emacs 21.
Newsgroups: comp.lang.ada

As far as I know, the real last version of ada-*.el is at: http://savannah.gnu.org/cgi-bin/viewcvs/emacs/emacs/lisp/progmodes/

For example, ada-mode 3.6 from http://libre.act-europe.fr/adamode/ main.html does not seem to support (perfectly) the new .ali format appeared in gnat 3.15.

[From: john@assen.demon.co.uk (John McCabe)]

I think you're right. It would be nice if the ACT free download and GNU versions at least appeared to be consistent

AdaCL - Ada Class Library

From: Martin Krischik <krischik@users.sourceforge.net>
Date: Tue, 02 Sep 2003 18:20:45 +0200
Subject: [Announcement] AdaCL 3.2.0 released.
Newsgroups: comp.lang.ada

[See also same topic in AUJ 24.3 (Sep 2003), pp.144-145. -- dc]

Notes: Maintenance Release, fixes bugs.

Changes: New Class BC.Support.Tagged_Reference to handle instances of abstract Classes.

Abstract: AdaCL is a library to write small, script like, programs in Ada. The main tasks currently implemented are filtering of text files (global search and replace), execution of external programs (inclusive redirection of standard input and standard output) and command line parsing.

In addition AdaCL features a true garbage collector (if you write scripts you don't want to think about memory management) and extensions to the Booch components for handling indefinite types.

The demo programm sarDo is slowly upgraded to a full featured search and replace tool and used by me on a daily basis.

http://www.ada.krischik.com

From: Martin Krischik <krischik@users.sourceforge.net>
Date: Tue, 23 Sep 2003 16:07:09 +0200
Subject: AdaCL 3.3.0 released.
Newsgroups: comp.lang.ada

Notes: This Release features new Indefinite Containers. The Indefinite Containers can store more different kind of data. Since they can store Element'Class as well Tagged Containers are not needed any more. Thanks to Matthew Heaney for pointing that out to me. Please see the new Homepage as well. [...]

Audio Input Packages

From: Chad R. Meiners <crmeiners@hotmail.com>
Date: Wed, 3 Sep 2003 12:04:42 -0400
Organization: Michigan State University
Subject: Re: Audio Input Package
Newsgroups: comp.lang.ada

> I'm looking for an Audio Input package of sorts - I need to open an audio file (probably just WAV format will do) and look at it. I don't need to directly access the sound card, nor play sound

> back through the speakers (although I guess that would be cool, but not necessary).

http://www.technology.niagarac.on.ca/courses/comp630/WavFileFormat.html

> Anyone have any ideas where I could find such a thing? If not, then something in C that could be linked [...] into Ada and used would do just as well. [...]

AdaSDL has a thin and thick binding that allows you to play wave files. In Windows you can easily call the appropriate API to play sounds.

[See also "AdaSDL Binding to Multiplatform Game Development Library" in AUJ 24.1 (Mar 2003), p.14. -- dc]

From: Matthew Heaney <matthewjheaney@earthlink.net>
Date: Thu, 04 Sep 2003 02:19:44 GMT
Subject: Re: Audio Input Package
Newsgroups: comp.lang.ada

I have a both WAV and AVI file parsers written in Ada95. They were used on the program that won the Ada-Belgium programming contest last year.

I can send you the sources. Send me your email address and I'll do it tomorrow.

(I plan on posting the entire sources on my website, but they used an early version of the Charles container library, which has been volatile of late because of AI-302.)

[See also "Ada-Belgium Announces Ada Programming Competition" in AUJ 22.4 (Dec 2001), p.195, and "Charles - Container Library" earlier in this issue. -- dc]

AdaVox 0.51 - Wave Sound File Player

From: Warren W. Gay VE3WWG <ve3wwg@cogeco.ca>
Date: Wed, 03 Sep 2003 13:10:02 -0400
Subject: Re: Audio Input Package
Newsgroups: comp.lang.ada

http://home.cogeco.ca/~ve3wwg/adavox-0.51.tar.gz

For a description see: http://home.cogeco.ca/~ve3wwg/software.html and scroll down to "AdaVox Sound Software" for the details.

[See also "AdaVox 0.5 - Wave Sound File Player" in AUJ 22.3 (Sep 2001), pp.138-140, and "C developer switches to Ada for Linux development" in AUJ 21.4 (Jan 2001), pp.246-248. -- dc]

AdaVox Features:

- Plays a wide range of *.wav files:
- Standard PCM wave files
- Microsoft ADPCM compression format
- IMA/DVI ADPCM compression format
- u-Law wave files
- a-Law wave files
- Plays Sun's *.au/*.snd file formats
- Standard PCM samples
- u-Law compression format
- a-Law compression format
- Plays multiple files in sequence
- Endian neutral (good for PowerPC)
- Ada95 sound packages can be used for applications
- Optional "realtime" priority for reliable operation
- Tested on FreeBSD
- Tested on Linux (RedHat 7.0, 2.2.16 kernel)
- Netscape friendly

Even though this software provides a "command", there are packages that can be used to do what you want to do. I had planned to do more with this, and maybe will someday, but I have more urgent projects to do first (like writing compressed formats). It will read compressed formats, but not generate compressed formats.

MIDI and Ada

From: Dr. Justice <aleistad@broadpark.no>
Date: Sun, 12 Oct 2003 05:24:15 +0200
Subject: Ada and MidiShare
Newsgroups: comp.lang.ada

I've decided that I want to try and use Ada for my next projects. I come from a background of mostly assembly, C and PL/SQL and wanting to move into a "new and better" language for general computing I can not think of anything better than Ada. I'm convinced I'll have a nice adventure - you can add two new users (a friend of mine as well) to the statistics

For Ada to be really useful for me, I need windowing GUI, MIDI I/O, and ideally audio I/O as well. My platform is Windows (can use Linux too) and GNAT 3.15p, and I'd like to be able to build for all of Windows, Linux and MacOS (with a priority on Windows).

GtkAda should do the GUI part nicely. Very nice kit it seems (I have yet to try it out in practice!).

After much searching and reading on the web I understand that MIDI and audio support is not readily available. It seems that Claw supports at least audio, but that is only for Windows. The audio support I'd want is very simple, just buffered streams to/from the audio inputs/outputs for realtime recording and playback. Performance is not critical initially. I will continue my search for possible C libraries, but I'm pretty blank in this area (sound/MIDI on Win/Mac/Linux).

[...] I still have more research to do before I'm sure how I will proceed with respect to the MIDI/audio bits. Anyway, I will get on with my Ada self-education, and play some with C bindings.

Having just these simple tools of MIDI and audio I/O could open up new possibilities for Ada, and attract new users, yes? [...]

If there are others who share an interest in creating MIDI/audio applications using Ada, and want to discuss or cooperate on IO libraries, or have any good pointers on these subjects, please let me know! As mentioned I have done /much/ searching, so the most obvious hits for Ada/sound/MIDI on AllTheWeb, Google and Dejanews are covered.

Are Leistad

From: aleistad <aleistad@broadpark.no>
Date: Fri, 17 Oct 2003 02:29:04 +0200
Subject: MIDI library for Ada
Newsgroups: comp.lang.ada

I'm glad to report that I now have an Ada program that plays a scale over MIDI (ref the thread: "Ada and MidiShare).

The immediate solution was right under my nose, in the form of Win32Ada's bindings to the Windows multimedia services (they are not mentioned in the Win32Ada.hlp file). All I needed to do was to instantiate some data types and call a couple of functions.

If I'm not mistaken, there is no real dependency on Win32Ada as such, it only provides thin bindings to the "libwinmm" library, and no Ada function are used. If the supplied libwinmm.a is the same as in general gcc distributions, all that needs to be done is to write a similar thin binding, covering only MIDI (and possibly audio). This is of course only the very basic byte-by-byte I/O.

The status so far is the MIDI is easily achievable, and I have the basics running fine on two different Win98 PC's .

To get reception going the use of a callback is required; that's next on my list of experiments (and my Ada education).

So - if there is interest, I'd propose that we co-develop a production quality standalone MIDI library (and possibly audio too). My own capabilities are limited, as I'm just learning Ada, but I will contribute in any way I possibly can. Please, let's discuss!

From: Patrice Freydiere <frett27@free.fr>
Date: Sun, 26 Oct 2003 09:19:31 +0100
Subject: Re: MIDI library for Ada
Newsgroups: comp.lang.ada

[...] I started implementing a full Ada MIDI library. Now, this library is able to read MIDI files, get all chunks and parse MIDI events. You can plug your custom

parsing procedure to do what you want. It is also able to write MIDI files.

The packages are not actually able to bind to a physical MIDI, or an OS Midi interface, but it must not be very difficult.

If you are interested, just email me I'll send you the sources.

Auto_Text_IO & SAL

From: Stephen Leake <Stephe.Leake@nasa.gov>
Date: 08 Sep 2003 10:09:26 -0400
Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)
Subject: new versions of SAL, Auto_Text_IO
Newsgroups: comp.lang.ada

I've released new versions of SAL (1.50) and Auto_Text_IO (3.00). These have been tested with GNAT 3.15p, GNAT 3.16a1, and GNAT 5.01a.

Available at http://www.toadmail.com/~ada_wizard/

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

SAL provides a container library, a robotics/satellite math library, and miscellaneous other stuff.

Auto_Text_IO generates Text_IO children for most Ada packages.

This is a significant enhancement of Auto_Text_IO. The run-time packages are now in SAL (replacing the old SAL.Generic_Array_Text_IO), and all of the child Text_IO packages in SAL are generated by Auto_Text_IO, so they now all provide Get.

Auto_Text_IO now supports generic packages, variant records, and allows users to override the core Put and Get to do special output formatting or input validation.

Also, the inside of Auto_Text_IO has been cleaned up significantly.

In SAL, I've also added new packages for Gaussian distributions, standard deviation, and made a few other minor changes.

Note that there is a bug in GNAT 3.15p that causes 'gnatmake all_sal' to fail when optimization is turned on. You can still use SAL with GNAT 3.15p with optimization; just build the pieces you need, by including the SAL project file in your project file. And hope GNAT 3.16p comes out soon.

Eliza in Ada

From: Jano <402450@cepsz.unizar.es>
Date: Mon, 15 Sep 2003 20:29:26 +0200
Subject: Eliza implementation in Ada
Newsgroups: comp.lang.ada

In another spin of my Ada whereabouts, I'm now interested in finding an Ada open source implementation of the classic Eliza chatterbot (or improved versions). I'm looking through bot code resources but I find surprisingly few things (or I'm searching really bad). I have located a few in Pascal, Basic and Prolog, but that's all.

The next best thing would be a C implementation which I suppose would be easy to bind, and finally an implementation in any language that an average programmer could understand (in that case, simplicity would be a plus, I'd not like to translate a program with a lot of [obscure] lines).

From: Rod Haper <rhaper@houston.rr.com>
Date: Mon, 15 Sep 2003 19:08:33 GMT
Organization: Road Runner - Texas
Subject: Re: Eliza implementation in Ada
Newsgroups: comp.lang.ada

The BUSH shell scripting language (GPL license) contains an Eliza script in the examples directory. Since BUSH is based on AdaScript (a subset of Ada 95) you should find it fairly ease to port to Ada 95. Here's a link to the BUSH home page: http://www.vaxxine.com/pegasoft/bush.html

[See also "PegaSoft - BUSH AdaScript Business Shell" in AUJ 24.2 (Jun 2003), p.85. -- dc]

Ada and Cryptography

From: Freejack <user@nospam.net>
Date: Wed, 01 Oct 2003 21:54:26 GMT
Newsgroups: comp.lang.ada
Subject: Glade and Cryptlib.

Has anyone attempted to integrate a crypto package such as cryptlib http://www.cs.auckland.ac.nz/~pgut001/cryptlib/ into the Glade Partition Control System?

This seems like it would be a useful project. I'm aware of the various Ada crypto efforts [see also same topic in AUJ 23.3 (Sep 2002), pp.139-140. -- dc], but I don't know of any that have attempted to take this sort of approach.

The reason I ask is because I've been playing with cryptlib for a while now and would like to take a shot at doing a distributed app which handles encryption and such transparently, using Glade. Cryptlib is pretty self-contained and portable, so it seems like a good place to start.

(I'm referring to the Gnat-glade DSA package, not the GUI builder.)

From: Pascal Obry <p.obry@wanadoo.fr>
Date: 02 Oct 2003 19:48:02 +0200
Subject: Re: Glade and Cryptlib.
Newsgroups: comp.lang.ada

See http://lglwww.epfl.ch/ada/filters/index.html. This is old but could be a good starting point.

Ada Source Code Reformatters

From: Laurent Pautet <pautet@inf.enst.fr>
Date: Thu, 23 Oct 2003 12:36:44 +0200
Subject: Re: Outil de reformatage des sources
cc: ada-france@ada-france.org

[Extracts translated from French: -- dc]

> I am interested in such a tool to reformat existing Ada code.

gnatpp (GNAT Pretty Printer)

From: Pascal <pascal.pignard@wanadoo.fr>
To: ada-france@ada-france.org
Subject: Re: Outil de reformatage des sources
Date: Mon, 27 Oct 2003 22:46:40 +0100

There's also the "reformat" tool.

See: http://wuarchive.wustl.edu/languages/ada/swtools/reformat, or for the Mac: http://www.adapower.net/macos/tools.html#Anchor_Reformat

Pascal Pignard, http://perso.wanadoo.fr/blady

Ada-related Products

ARTiSAN - Goodrich Engine Control Systems Standardizes on ARTiSAN's Real-time Studio

URL: http://www.dedicated-systems.com/VPR/layout/display/pr.asp?PRID=6368

Summary: Goodrich Engine Control Systems in Birmingham, UK, Selects ARTiSAN's Real-time Studio Professional for Mission Critical Software Development

Full Text:

Cheltenham, UK September 23, 2003. ARTiSAN Software Tools, a global leader for UML-based, real-time systems and software modeling tools, today announced that Goodrich Engine Control Systems' Birmingham UK facility has selected ARTiSAN's Real-time Studio Professional as its standard tool for software development on mission critical, UML-based projects. Goodrich is one of the world's leading aerospace engine control systems suppliers.

"We have been using Real-time Studio for roughly two years," commented Jim Daly, System Architect at Goodrich. "The tool has proven to be very stable, versatile, quick to learn, and is a popular choice with our engineers. We evaluated other tools but Real-time Studio differentiated itself with its extremely intuitive and customizable user interface. It is very flexible supporting four variants of Ada, it can be integrated with tools from other

vendors and also has a degree of built-in configuration management. ARTiSAN's consulting services quickly brought us up to speed with Real-time Studio and UML and helped us to achieve success on our projects much faster than would have been possible on our own."

"Real-time Studio's user interface has evolved to the point where a newcomer to the tool can quickly come up to speed, and an expert user can focus on actual work rather than workarounds," commented Jeremy Goulding, President and CEO of ARTiSAN Software. "Out of the box, whether users are working with C, C++, Java or Ada, the tool is very instinctive quite simply, it works the way you do. Its open architecture makes it even more intuitive, allowing users to customize the tool to suit specific needs. Real-time Studio is one of the last remaining design tools to truly support the Ada community not simply from a marketing perspective, but from a practical, engineering one, and in a UML design process too. Our customers working on mission and safety critical applications, such as Goodrich, expect and deserve nothing less."

About ARTiSAN Software Tools: [see announcement in AUJ 24.3 (Sep 2003), p.147, or visit www.artisansw.com. -- dc]

About Goodrich Engine Control Systems

Goodrich is the world's leading independent engine control supplier. Among the few truly global corporations in the aerospace equipment sector, no one has more jet-age experience, a better understanding of leading technologies and broader systems capabilities to optimize the overall control system. It has a wide range of applications for civil engines, large and small, helicopters and all forms of military aircraft including trainers, transporters, and fighters.

Goodrich is committed to building lighter, smarter systems that last longer, and has extensive experience in developing high reliability electronic engine controls with safety critical software. For more information, visit www.enginecontrols.goodrich.com

Contact: ARTiSAN Software Tools Inc., Richard Gastwirt, richard.gastwirt@artisansw.com

DDC-I - Windows Migration Package for TADS-68xxx Customers

URL: http://www.ddci.com/news_vol4num6.shtml
Subject: Embedded News from DDC-I - DDC-I Online News

August/September 2003 [...] DDC-I Online News [...] Vol. 4 Issue 6 [...]

DDC-I Offers Current TADS-68xxx Customers A Budget-Conscious Windows Migration Package

Phoenix, AZ. August 15, 2003. Streamlining the transition from VAX or Unix-hosted development systems for their existing TADS user base, DDC-I today announced the availability of their Windows (NT/2000/XP) migration package, now also available for TADS-68xxx customers. Fully customizable, it offers current TADS (68xxx, 1750A & i960) customers a direct, affordable migration path to the most popular PC-based network and enterprise computing platform.

"Allowing customers to define which tools and support they require, rather than handing them a rigid list of tiered options, is why we created a flexible TADS for Windows migration package," explains Harold "Bud" Blum, DDC-I Senior Software Engineer and Product Champion for the TADS product line.

Customers dictate their package parameters to create a least-cost migration path with DDC-I's expert guidance. To keep recurring costs level, software support from any current license agreement carries over, and the customer has complete freedom to select the quantity of seats to rehost and whether to upgrade their software versions during the migration. All necessary license transfers and keys to replace current TADS licenses are included.

Two days of onsite consulting are also included in the package at no additional charge to assist with rescripting, tool adaption, memory and segment set up, related Ethernet work, board support packages and a final project report with detailed recommendations.

"Our customers safety-critical software development tools have to keep pace with the latest development environments, and upgrading the TADS products to the Windows platform gives them the ability to handle such taxing upgrades with minimal disruption to the development environment they depend on," concludes Blum.

DDC-I - Multi-Language SCORE IDE Offers Windows Native Capability

URL: http://www.ddci.com/news_vol4num7.shtml
Subject: Embedded News from DDC-I - DDC-I Online News

October 2003 [...] DDC-I Online News [...] Vol. 4 Issue 7 [...]

DDC-I's Multi-Language SCORE IDE Offers Windows Native Capability

Phoenix, AZ -- October 1, 2003 -- DDC-I today announced the addition of Windows Native capability to the versatile SCORE (Safety Critical, Real-time Embedded) Integrated Development Environment.

The primary purpose of Windows native capability for SCORE is to support the same functions as our cross products, and it is especially useful for engineers who need to start software development and testing before their custom hardware is available, explains David Mosley, DDC-I Engineering Manager and Product Champion for SCORE.

The first multi-language IDE based on non-proprietary open system standards, SCORE is a Commercial-Off-The-Shelf product delivering ease-of-use at every project level while guaranteeing maximum software portability and reusability.

Using the same graphic interface as all of DDC-I's Windows-hosted cross-compilation products, SCORE's Windows Native capability offers proven quality while also saving significant time and money during the transition to new processor technologies. Leaving embedded system developers free to mix application development among different programming languages including C, Embedded C++ and Ada, the toolset includes a highly reliable compiler, a seamlessly integrated multi-language debugger and two small, exceptionally fast run-time systems (tasking & non-tasking).

The key components in SCORE are DDC-I's next generation compilers. Based on ANDF (Architecture Neutral Distribution Format) technology, compilers for each programming language generate a common intermediate representation which is converted to the final object code during a later language-independent phase. SCORE supports multiple languages, host environments and target platforms. Since ANDF is an Xopen standard, the SCORE system possesses a truly open architecture.

Developers today increasingly need to migrate software to new targets, and we are constantly expanding the SCORE IDE to eliminate barriers to efficient multi-language development and address the growing need to combine reusable software components, written in different languages, targeting different processors and often developed on different platforms, concludes Mosley.

DDC-I - New Pricing Option for SCORE IDE

URL: http://www.ddci.com/news_vol4num8.shtml
Subject: Embedded News from DDC-I - DDC-I Online News

November 2003 [...] DDC-I Online News [...] Vol. 4 Issue 8 [...]

DDC-I's Multi-Language SCORE IDE Offers Flexible Subscription Pricing Royalty Free!

Phoenix, AZ. October 21, 2003 - Veteran software tools supplier DDC-I today announced a subscription pricing option for the versatile SCORE (Safety Critical, Object-oriented, Real-time Embedded) Integrated Development Environment as a cost-sensitive alternative to traditional licensing models. Allowing the customer to subscribe to only the number of program seats actually required during each single year period, from one to whatever needed, DDC-I's subscription model is highly competitive, and even includes DDC-I's "Atlas Advantage" support package.

"With a clear understanding of increasingly challenging budget structures confronting many project programmers and the long-term importance of minimizing maintenance costs, subscription pricing for SCORE offers customers the simplest, most sensible means to align programming and project expenses," explains DDC-I President Ole Oest.

Providing industry-leading programming products and support services to a distinguished list of commercial and defence contractors including Boeing, Lockheed-Martin, and Sikorsky, DDC-I also supports diverse development beyond aviation and aerospace, from commercial satellite and telecommunication systems to next-generation voice and data network technology, bullet trains and medical equipment.

Subscription pricing reduces up-front costs, allowing the customer to increase or decrease the number of subscribed seats to match the dynamics of a program's staffing profile. In the maintenance phase of the program's life cycle the subscription could go as low as one seat.

With subscription pricing no cash is tied up in a capital purchase, and while the annual subscription fee is very competitive, the customer still retains the benefits of DDC-I's acclaimed support program. In addition, SCORE remains royalty free (no fee is associated with distribution of the customer's application), an additional feature aimed directly at the customer's bottom line.

"The creativity, engineering support, and customer service that help our customers succeed are what we continue to offer every developer in the real-time embedded systems market," Oest concludes. "Whether it's aerospace, satcom and telecom, networking infrastructure - or any safety-critical application where system downtime is simply not an alternative - we have the tools and pricing programs to help our clients excel."

Green Hills - Barco Selects INTEGRITY-178B RTOS for MOSArt

URL: http://www.dedicated-systems.com/VPR/layout/display/pr.asp?PRID=6296
Release Date: Monday, September 15, 2003

Barco Selects INTEGRITY-178B RTOS for its MOSArt Modular Open System Architecture

Green Hills Software's INTEGRITY-178B real-time operating system (RTOS) has been selected by Barco, a leading provider of high performance imaging technologies, for its new MOSArt avionics open system.

Barco's MOSArt is an ARINC 653 compliant Modular Open System Architecture for real time avionics applications. MOSArt's open, modular approach leads to a reduced weight and power consumption, improved MTBF as well as allowing every partition to have a different criticality level (up to DO178B level A).

The MOSArt architecture allows system integrators to easily develop and/or integrate their own software on Barco's powerful open hardware platform. By doing this, customers can preserve the intellectual property of their core technology (FMS, mission computer, etc).

"We chose INTEGRITY-178B from Green Hills as our first choice of RTOS for MOSArt because we strongly believe that this is the best product on the market for building ARINC 653 based modular open system architecture avionics subsystems," said Jean-Christophe Monfret, R&D Manager Software, BarcoView - Avionics.

INTEGRITY-178B is an ARINC-653 compliant, hard real-time RTOS optimised for safety-critical and mission-critical applications that require the utmost security and fast, predictable response. Utilising hardware memory protection and an advanced two-level partition scheduler, INTEGRITY-178B provides complete time, space, and resource partitioning between applications operating on the same hardware platform. INTEGRITY-178B also provides guaranteed resource availability in both the time and space domains. This combination facilitates "robust partitioning" (as defined in ARINC 653), enabling applications that have been assigned different DO-178B safety levels to run concurrently on the same processor.

INTEGRITY-178B includes an RTOS simulator (ISIM) that enables programmers to develop and test their code on a PC or workstation without the need for target hardware. INTEGRITY-178B also features a real-time event analyzer (EventAnalyzer(TM)) that enables viewing of system and user events in a graphical display.

Editorial contact information: Christopher Smith, Vice President of Marketing, Green Hills Software, Inc., chriss@ghs.com. Sales enquiries to: Green Hills Software Ltd, mktg-europe@ghs.com, www.ghs.com

Praxis Critical Systems - New SPARK & RavenSPARK Definitions Available

From: rod.chapman@praxis-cs.co.uk (Rod Chapman)
Date: 7 Oct 2003 10:46:04 -0700
Subject: ANN: New SPARK (including RavenSPARK) definition now available
Newsgroups: comp.lang.ada,comp.lang.eiffel,comp.realtime

Praxis Critical Systems are pleased to announce the new definition of the SPARK language (including RavenSPARK) is now available for public comment.

RavenSPARK adds the tasking facilities of the Ada95 Ravenscar Profile to the core SPARK language - probably the most significant expansion in the expressive power and size of SPARK ever made.

[See also "Praxis Critical Systems - SPARK Incorporates Ravenscar Features" in AUJ 24.1 (Mar 2003), p.22. -- dc]

In addition to the language definition, the following documents are also available: An idiom guide and rationale for RavenSPARK (read this first!); A worked design exercise using RavenSPARK; A Quick Reference Chart for RavenSPARK.

These documents are all available for download in PDF format from www.sparkada.com

We welcome comments regarding SPARK from all interested parties – please contact us at sparkinfo@praxis-cs.co.uk

Announcements regarding professional tool support for RavenSPARK, and upgrades for buyes of the "SPARK Book" will be made shortly.

Rod Chapman, SPARK Team, Praxis Critical Systems

Praxis Critical Systems - SPARK Toolset 7.0

From: rod.chapman@praxis-cs.co.uk (Rod Chapman)
Date: 17 Oct 2003 05:53:31 -0700
Subject: ANN: SPARK Toolset release 7.0
Newsgroups: comp.lang.ada

We're pleased to announce the immediate availability of Release 7.0 of the SPARK Toolset. Most importantly, release 7.0 supports the RavenSPARK tasking extensions to SPARK.

The toolset release note and a press release are available at www.sparkada.com as usual.

Supported customers, academic users and tool partners should have all received their upgrade packages by now. An upgrade package for buyers of John Barnes' SPARK Book will be available soon.

Rod Chapman, SPARK Team, Praxis Critical Systems

Praxis Critical Systems - SPARK Book Upgrade Packages Available

From: rod.chapman@praxis-cs.co.uk (Rod Chapman)
Date: 17 Oct 2003 11:01:37 -0700
Subject: ANN: SPARK Book Upgrade packages now available
Newsgroups: comp.lang.ada

I'm pleased to say that upgrade packages for the "SPARK Book" are now available from www.sparkada.com for both Windows and GNU/Linux.

These bring the "Free Demo" SPARK toolset up to release 7.0, and also include a complete new set of documentation describing the latest release of SPARK, the RavenSPARK language extensions, and all the new tools.

Two RavenSPARK example programs are included (a Stopwatch controller, and the ubiquitous Minepump controller...)

These packages are large (over 5 Megabytes each), so please be patient if our server seems temporarily overloaded.

SPARK Team, Praxis Critical Systems

RainCode - Evaluation Version of RainCode Checker

From: Deborah Torrekens <deborah@phidani.be>
Date: Mon, 11 Aug 2003 13:10:06 +0200
Subject: Announcement: coding rules checking
Newsgroups: comp.lang.ada

RainCode just published an evaluation version of the RainCode Checker, which checks about 70 Ada coding rules on a set of 20 Ada sources. These are standard rules, but we can implement company-specific rules in the tool as well.

To download it, please go to http://www.raincode.com/online, and sign on.

Once you've logged on, please select "RainCode product line" on the home page, then click on "Downloads" in the menu. Choose: "RainCode Engine for Ada demo version." There, you will have 2 files that you can download:

adarcdemo.zip: is a demo and evaluation version of the RainCode Engine for Ada. The demo dynamically shows you a few RainCode scripts applied on a set of GNAT sources. This version also allows you to write your own scripts and run them on the set of given sources. To find out how to do this, please open: readme.txt. Note that under Windows 2000, you need to have the Administrator rights on your machine, in order to install this demo/evaluation version.

checkerAda.zip: about 70 Ada coding rules can be checked on a given set of 20 Ada sources, in a graphical interface. To run it, you will need to have Java 1.3 or higher installed on your machine. Unzip the file, and double click on run.bat to start the program. Please open the readme.txt file for more information.

Don't hesitate to contact me for any further information you might need.

[See also "RainCode - Evaluation Version of RainCode for Ada Available" in AUJ 23.3 (Sep 2002), p.146. -- dc]

McKae Technologies - First Public Release of DTraq Data Logging and Playback Debugging Tool

From: Marc A. Criley <marc@mckae.com>
Date: Mon, 01 Sep 2003 21:57:53 GMT
Subject: Announce: DTraq First Public Release
Newsgroups: comp.lang.ada

McKae Technologies announces the first public release of DTraq, version 0.960.

DTraq is a data logging and playback debugging tool providing near realtime data logging and analysis to aid debugging and validation. Captured, or 'tapped' data from a program can be viewed live while the program is running or, since it is being logged to a file, played back or printed out later for off-line review and analysis.

DTraq differs from other logging and playback tools in that no data layout maps or byte interpretations or "data dumpers" need to be manually created. Nor is the application responsible for converting the raw binary data to text form before logging it. DTraq handles all conversion automatically by scanning the application's source code, identifying tapped data items, and extracting the information it needs to properly convert and display the logged items-simple scalar items as well as arrays and records. When the layout of data items change, rescanning automatically picks up the changes.

The DTraq binary distribution, comprehensive documentation, and source code is available for download at http://www.mckae.com/dtraq.html.

DTraq is currently available for Linux systems running GNAT 3.15p.

DTraq operation has been verified on both RedHat 8.0 and RedHat 9.0 Linux distributions (albeit the user needs to set an environment variable when using the latter distribution--a README.RedHat9 file in the distribution covers this). Because DTraq utilizes ASIS-for-GNAT, which is tightly bound to a corresponding compiler version (in this case 3.15p), DTraq has currently been validated only against GNAT 3.15p.

For more information or questions, please contact marc@mckae.com.

Marc A. Criley, Mckae Technologies

From: Guillaume Foliard <guifo@wanadoo.fr>
Date: Tue, 02 Sep 2003 22:54:22 +0200
Subject: Re: Announce: DTraq First Public Release
Newsgroups: comp.lang.ada

> I've never used nor seen such a tool. Can you provide more information about what it can be used for? How can it help debugging?

I'm using such a tool at work, which has been developed in-house. It allows you to record during execution data messages (effective parameters, data going through a network connection, etc...) and to analyse them afterwards.

The important part there is not to record the data but to interpret it without having to type million lines of Put procedures by hand. This is a two step process. First you need an Ada parser to analyse the data structure of what you want to record and to generate all the Ada code needed to access and display those structures. Then this code will be used in other applications for displaying, plotting, etc...

From my own experience with such a tool, and with the help of a good software design, more than 90 percent of the defects are resolved by just reading the inputs and the outputs and the involved code, without any debugger. This is indeed possible because of the natural safety of Ada. When bugs occur we can generally safely ignore all the low-level bugs C programmers are familiar with to focus on the functional stuff.

From: Jeffrey Creem <jeff@thecreems.com>
Date: Wed, 03 Sep 2003 00:00:48 GMT
Subject: Re: Announce: DTraq First Public Release
Newsgroups: comp.lang.ada

I have not looked at the details about what this tool provides but at a high level I can say that this could be really useful for "system level" debugging/analysis information.

What I mean is that in many systems, one ends up creating some sort of data logging interface that keeps track of either inputs to some top level module and/or intermediate calculations to better understand how the system is operating.

Sounds like this would be a quick way to get this information out more easily. It probably will not be as quick as some of

the hand/rolled solutions (which dump data in a raw binary format for later off-line expansion to ASCII) but still useful in some cases.

[From: Marc A. Criley <marc@mckae.com> -- dc]

Pretty close.

DTraq does log data in binary form, though, so you get the performance benefits of doing that. And it also does both near simultaneous conversion to text for live monitoring, and post-run conversion for playback, printing, and analysis. (The logging and text conversion is performed by a logging server, which can run on a totally different machine so as to minimally interfere with the system under test. Data is transferred from app to server via TCP/IP.)

[From: Jeffrey Creem <jeff@thecreems.com> -- dc]

Cool.. It makes it even more exciting than I had originally thought. (And I was already pretty excited about it!)

From: Marc A. Criley <marc@mckae.com>
Date: Wed, 03 Sep 2003 00:50:52 GMT
Subject: Re: Announce: DTraq First Public Release
Newsgroups: comp.lang.ada

> I've never used nor seen such a tool. Can you provide more information about what it can be used for? How can it help debugging?

Many projects of any significant size have some sort of "logger", a package or class or file of logging functions. As the program runs, information that the developer has decided would be useful to record for subsequent analysis or monitoring is logged. (Actually, specific logging requirements may also be placed on an application by its customer, especially for military systems.)

So such loggers basically record to one or more log files the information that is submitted to the logging service while the app runs, along with a timestamp and perhaps other useful information.

There are two general approaches regarding the form of the logged data: One is that the application preconverts the data to text and then logs that text, so the application has the responsibility of properly interpreting and formatting the data that is then logged.

The second approach is to more-or-less provide an address and a number of bytes to the logger, and the logger then writes those raw bytes to the log file. This address/size combination can sometimes be dressed up as somehow simply specifying the object to log, or utilizing a suitably instantiated generic (which is what DTraq does).

The problem with the first approach is that text representations usually take up more bytes than the raw data, which means added I/O or network traffic, plus the text conversion impacts application performance.

The problem with the second approach is that the raw bytes still have to be converted to text at some point. So there are a couple typical ways of doing that: defining layout templates that are overlayed on the data to break them up and that describe how to interpret each piece -- scalar, record component, array element, string, integer, float, character, etc. Another is to create a "data dumper" that programmatically converts a data stream into text, which is then invoked by the logging system's "viewer". And of course whenever a data item's format changes, the corresponding template or dumper needs to be updated, and when there's a need to log a new type of data, a new template/dumper has to be created.

For the basic logging part, DTraq is like most other loggers – you identify the data item you want to log, instantiate a generic procedure with the type of that data item, and then add invocations of that instantiation wherever you want to "tap" the value of that item.

Something like:

```
type Color_Type is
   (Red, Green, Blue);
Current_Color : Color_Type :=
   Red;
procedure Tap_Color is
   new DTraq.Tap (Color_Type,
                  121,
                  "Color_Type");
-- Tap's parameters are the type
-- to tap, a numeric ID you pick
-- to associate with it, and an
-- optional string that will be
-- later conveyed to the viewer.
```

Then wherever you want to record the current value insert:

```
Tap_Color(Current_Color);
```

At this point the current value of Current_Color goes out to the log file.

Now what you can do with DTraq is monitor your taps live while the program is running, and verify that they're taking on the values you expected as the program is put through its paces.

After you've done your run, you can then reload a log file into Dtraq and have it play it back for you, single or multiple step through the logged data, forward or backward, reviewing how the key values changed as the program executed. And of course you can print out all or portions of the logged data.

The key area where DTraq differs from other loggers is in the conversion from raw bytes to text. DTraq scans your Ada source code, and with the aid of the Ada Semantic Information Specification (ASIS) automatically locates the "Tap" invocations, identifies each type being tapped, and analyzes it (including any components it may have) to understand how to convert instances of the data type into corresponding text, whether that item is a number, a character, string, enumeration, record, or array.

This is all automatically done by running the "mkdtq" component of DTraq over your code base--you NEVER have to manually convert, or describe how to convert, raw data to text. When a data item changes or a new one is added, another invocation of mkdtq automatically picks up the changes.

Makes things _much_ less tedious

Hope this helps, there's detailed info in the DTraq user manual, and a fully working example accompanies the distribution and is gone over step by step in the manual.

McKae Technologies - DTraq 0.970 Available

From: Marc A. Criley <mc@mckae.com>
Date: Tue, 21 Oct 2003 00:28:22 GMT
Subject: Announce: DTraq 0.970 now available
Newsgroups: comp.lang.ada

DTraq 0.970 has been released and is now available on the Mckae Technologies website (www.mckae.com). In addition, the site's DTraq section [...] has been updated with a walkthrough of DTraq capabilities and screenshots illustrating its operation.

Modifications for DTraq 0.970 were mostly internal, fixing minor bugs and reducing system-dependencies so as to ease porting to other platforms.

The one visible change is that the Data Viewer can now display the raw memory content of logged data items, rather than just the interpreted forms. See the screenshots and the DTraq User Manual, (www.mckae.com/dtq_common/DTraq.pdf) for more information.

Marc A. Criley, McKae Technologies, "The Efficient Production of Reliable Software"

Excel Software - WinA&D Design Tool Adds Ada Modeling and Code Generation

From: Excel Software <excel@lobo.net>
Date: Mon, 22 Sep 03 18:28:17 -0600
Subject: WinA&D 3.5 Adds Ada Modeling and Code Generation
To: <Dirk.Craeynest@aubay.be>

Design Tool for Modeling Software and Managing Requirements

Placitas, NM - September 22, 2003 - Excel Software began shipping a major upgrade of WinA&D with advanced capabilities for modeling and generating

Ada code. WinA&D is a comprehensive software engineering tool for structured analysis and design, OOA/OOD with UML, multi-task design, data modeling, requirements management, code generation for popular programming languages and a built-in scriptable reporting engine.

Ada is a programming language used primarily on mission critical defence projects. Working closely with a large defence contractor, Excel Software has mapped a streamlined UML notation to the Ada programming language. When Ada is selected in WinA&D 3.5, tailored UML class models allow the designer to easily represent Ada packages and various types of relationships like aggregation (parent/child and parent/nested packages), generalization (generic instantiation) and dependency (Withed packages).

The UML modeling experience has been tailored to fit the conventions, standards and practices used by Ada developers. For example, dialogs with drop-down pick lists make it easy to create records, arrays and other data types, share data types between packages and construct subprogram arguments with minimal typing. Visual relationships between classes (Ada packages) in the UML model determine the code structure.

The scope of package components like types, variables, constants and subprograms are visually represented on the UML class model and automatically transferred to the Ada implementation during code generation.

WinA&D runs verification checks to ensure model consistency, completeness and design integrity prior to code generation. Ada code generated from the model is an exact implementation of the design including full package specifications and body files, subprogram files with arguments, data types and stubbed implementations. Ada's With dependencies between files are also included in the generated code. The code is linked directly to the design, enabling the developer to click on the class model and view the code in the integrated code browser.

For non-Ada developers, WinA&D 3.5 adds new features including a data types dialog, a synchronized contents view for diagram organization and navigation, new diagram presentation options, enhancements for listing diagrams and associated dictionary information between projects and new reengineering features.

WinA&D runs on Windows 95, 98, NT, 2000 or XP. It is available in four Single User License editions; Standard $495, Desktop $1295, Educational $845 and Developer $1995, or by 5-User and Unlimited User Site License. Contact Excel Software or visit www.excelsoftware.com for product information and online ordering.

Excel Software, info@excelsoftware.com

Excel Software - WinTranslator Adds Ada to UML Reengineering

From: Tools <excel@lobo.net>
Date: Sun, 28 Sep 2003 12:51:09 -0600
Subject: Generate UML Models from Ada Code
Newsgroups: comp.lang.ada

WinTranslator 2.2 Adds Ada Reengineering.

Generate UML Class Models from Ada Source Code.

Placitas, NM - September 29, 2003 - Excel Software began shipping a major upgrade of WinTranslator that adds the capability of scanning Ada source code and generating graphic UML class models in WinA&D.

WinTranslator is a reengineering tool that works with Excel Software's WinA&D, QuickUML and QuickCRC modeling tools. Object-oriented software written in C++, Java, Delphi or Ada creates UML class models using WinA&D or QuickUML. CRC cards are created with QuickCRC. Code written in C, Pascal, procedural Basic or Fortran generates structure charts in WinA&D. Rich data models are generated from SQL for popular RDBMS products. Generated models can be automatically organized into multiple diagram levels to easily accommodate very large software systems. Diagram objects click to source code using an integrated code browser.

Ada is a programming language used primarily on mission critical defence projects. Working closely with a large defence contractor, Excel Software has mapped a streamlined UML notation to the Ada programming language. WinA&D implements UML modeling that is tailored for Ada with automated code generation.

WinTranslator captures information about each Ada package, relationships between packages and package components like records, arrays and other type definitions, variables, constants, named numbers, exceptions and subprograms. Details captured for subprograms include return data types and argument lists. Generic package parameters and actual parameters of generic instantiations are also captured. The public, private and implementation scope of each package component is expressed on the generated UML class models.

The typical code reengineering process involves creating a list of code files, extracting dictionary information to a text file, importing that information into a modeling tool and generating graphic diagrams that represent the source code. WinTranslator fully automates that process with a multi-step dialog that guides the developer to enter project information like programming language and code folder locations and then generates and executes a script of commands to reengineer the project. WinTranslator outputs a dictionary entry list to a text file that's imported into WinA&D to populate its dictionary. WinA&D's new Class Model From Ada command lets a developer select a collection of classes (Ada packages) from the dictionary, then generates a rich UML class model.

WinTranslator is priced at $495 for a Single User License and runs on Windows 95 through XP. Contact Excel Software for site license and upgrade prices or visit www.excelsoftware.com for information and online ordering.

Excel Software, info@excelsoftware.com

Ada and CORBA

Status of the CORBA Mapping for Ada

From: Laurent Pautet <pautet@enst.fr>
Date: Mon, 20 Oct 2003 16:57:34 UTC
Organization: ENST, France
Subject: Is CORBA dead for Ada
Newsgroups: comp.lang.ada

It seems to me that the CORBA mapping for Ada is almost dead. The last official document is formal/01-10-42 for CORBA 2.3. The ada-rtf team does not seem to be very active in its job to update the mapping (the last closed issues are from 1999).

OIS which seems to lead this task force is very shy in its promotion of Ada on its web site.

TopGraph'X is still promoting its Ada products. If ORB-River is compatible with CORBA 2.6, it seems that it does not include new features from versions greater than 2.3.

Does anyone have fresh news on the OMG activities around the Ada mapping?

[And from another message: -- dc]

> [...] I wouldn't be surprised if the Ada team are waiting for an indication of how the mutual type reference problem is going to be solved in the new revision of Ada before undertaking any major review of the Ada binding.

This would mean that it is not possible to use CORBA > 2.3 until Ada0Y is complete. I do not believe that.

The last CORBA revision is CORBA 3.0. The last Ada mapping is for CORBA 2.3. In between, many important features have been introduced in CORBA and many issues have been raised in the (old) mapping (mostly by O. Kellogg). But the Ada mapping revision seems to be frozen for years. My belief is that there is no real CORBA market for Ada so the major tool

vendors focus on other markets like the Java and C++ ones.

From: Jean-Claude Mahieux <jeanclaude.mahieux@topgraphx.com>
Date: Tue, 21 Oct 2003 18:41:17 +0200
Organization: Top Graph'X
Subject: Re: Is CORBA dead for Ada
Newsgroups: comp.lang.ada

> TopGraph'X is still promoting its Ada products. If ORB-River is compatible with CORBA 2.6, it seems that it does not include new features from versions greater than 2.3.

Laurent, I entirely disagree with that (see our web site).

OrbRiver/Ada supports: Interoperable Naming Service, Corbaloc/Corbaname URLs, CORBA Messaging, RT CORBA, MIOP (with a reliable implementation to come soon), Notification Service (entirely implemented in Ada95).

OrbRiver/C++ is approximately at the same level and OrbRiver/Java a little bit behind.

[See also "Top Graph'X - PrismTech Adds CORBA Ada Support to its OpenFusion Product Line" in AUJ 24.3 (Sep 2003), p.150. -- dc]

Jean-Claude Mahieux, Top Graph'X Sales Manager, Marcoussis, France, http://www.topgraphx.com

From: pautet@antigone.enst.fr (Laurent Pautet)
Date: Tue, 21 Oct 2003 23:13:11 UTC
Subject: Re: Is CORBA dead for Ada
Newsgroups: comp.lang.ada

Fine, Jean-Claude ! We have some of these with PolyOrb as well for instance the two last ones. [See also "PolyORB 0.1 – Schizophrenic Object-Oriented Middleware" in AUJ 23.1 (Mar 2002), p.16. -- dc] But my question was about the status of the Ada mapping. CORBA/Ada is almost unusable for a user without the mapping. Has it been updated?

[And from a later message: -- dc]

> Strange... Mr. Pautet you are one [of the] persons behind PolyOrb... PolyOrb leads to ACT. If I can get support from ACT for PolyOrb so why does ACT (as an Orb vendor) not participate in the maintenance of the CORBA standard?

As far as I am concerned, I don't see the maintenance of the Ada mapping as a research interest (anyway, I would need a funding to participate to it). But in the context of schizophrenic middleware, I am interested in an up-to-date mapping.

Concerning ACT, ask them.

From: volkert@nivoba.de (Volkert)
Date: 22 Oct 2003 11:20:45 -0700
Subject: Re: Is CORBA dead for Ada
Newsgroups: comp.lang.ada

> My belief is that there is no real CORBA market for Ada so the major tool vendors focus on other markets like the Java and C++ ones.

I hope some Ada (compiler, Orb) vendors are clever enough to see the impact of middleware technologies for the future Ada market. For the most companies using Ada now it is really important to get/keep their systems accessible from other SW systems created with modern/mainstream technologies. [...]

We have a quite large Ada system running here [...] We are starting to evaluate one Ada ORB right now.

[And from a later message: -- dc]

> You may want to check out libre.act-europe.fr/polyorb/

Already done, but I had problems with connecting Java clients (SUN JDK 1.4.2/SUN ORB) to Ada servers. As long as PolyOrb is not a >visible<< supported product of ACT, it is not interesting for us ...

From: Lionel.Draghi@fr.thalesgroup.com
Date: Thu, 23 Oct 2003 10:50:43 +0200
Subject: Re: Is CORBA dead for Ada
Newsgroups: comp.lang.ada

> [...] Believe me, if there is no visible progress with the CORBA standard the things will getting harder for some of us ...

I agree with you.

Others posts here answered about actually existing Ada ORBs, and that's OK. But Laurent's original demand was about the standard. If Ada ORBs do not stick to the latest CORBA version, it will cause interoperability problems, and will also be interpreted as a further Ada influence loss.

[...] Maybe need we just that ORB vendors set our minds at rest by reaffirming their commitment in Ada for CORBA, as do by Jean-Claude in this thread.

From: dirk@cs.kuleuven.ac.be (Dirk Craeynest)
Date: 23 Oct 2003 12:46:30 +0200
Organization: Ada-Belgium, c/o Dept. of Computer Science, K.U.Leuven
Subject: Re: Is CORBA dead for Ada? No, it doesn't appear to be.
Newsgroups: comp.lang.ada

It may be useful to point out here that similar threads are ongoing in the comp.object.corba newsgroup. Victor Giddings of Objective Interface System posted there some responses to the issues mentioned above.

FYI, I've included the most relevant below. [See Victor Giddings' postings of Wed 22 Oct 2003 to comp.object.corba, included further in this thread. -- dc]

That said, it wouldn't hurt to *see* a bit more Ada related news from CORBA vendors, both in this newsgroup and elsewhere.

Dirk (Dirk.Craeynest@cs.kuleuven.ac.be for Ada-Belgium/Europe e-mail)

*** Intl. Conference on Reliable Software Technologies - Ada-Europe'2004

*** June 14-18, 2004, Palma de Mallorca, Spain ** www.ada-europe.org ***

From: Lionel.Draghi@fr.thalesgroup.com
Date: Thu, 23 Oct 2003 16:06:18 +0200
Subject: Re: Is CORBA dead for Ada? No, it doesn't appear to be.
Newsgroups: comp.lang.ada

Thank you. As I understand it, the job is ongoing. This is reassuring.

From: Laurent Pautet <pautet@enst.fr>
Date: Tue, 21 Oct 2003 09:15:48 UTC
Subject: CORBA mapping for Ada
Newsgroups: comp.object.corba

It seems to me that the last Ada mapping is for CORBA 2.3. But nothing seems to be done to update it for CORBA 2.X (X > 3) or CORBA 3.0. Could someone tell me what is going on? Is the ada-rtf still active?

From: gduzan@bbn.com (Gary D. Duzan)
Date: Tue, 21 Oct 2003 16:35:28 GMT
Subject: Re: CORBA mapping for Ada
Newsgroups: comp.object.corba

The OMG site shows a document number for a revised Ada mapping with this month's date on it, but no actual document yet. I would expect to see something soon, and the PTC will likely address it at the November meeting in London.

Gary Duzan, BBN Technologies, A Verizon Company

From: Victor Giddings <victor.giddings@ois.com>
Date: Wed, 22 Oct 2003 20:46:55 -0000
Organization: Objective Interface System
Subject: Re: CORBA mapping for Ada
Newsgroups: comp.object.corba

> It seems to me that the last Ada mapping is for CORBA 2.3. But nothing seems to be done to update it for CORBA 2.X (X > 3) or CORBA 3.0.

I always cringe a little when I see statements like this, even if it is the fault of the OMG in part. The fact is that IDL does not change very often, so the mapping for CORBA 2.3 is equally useful for 3.0, etc. The work of updating is more often adding a new operation to CORBA.Object or something like that, or clarifying the specification of existing features of the mapping. In fact, anything more major would be outside of the scope of what an RTF (REVISION Task Force) is allowed to do. Case in point was value types, which required a new RFP.

[And on Gary Duzan's response about a revised Ada mapping soon: -- dc]

This is correct. We are working to have an update to meet the 3-week deadline for the London meeting (due Monday!). Status can be tracked at

http://www.omg.org/techprocess/meetings/schedule/Ada_2003_RTF.html

Victor Giddings, Senior Product Engineer, Objective Interface Systems

From: Victor Giddings <victor.giddings@ois.com>
Date: Wed, 22 Oct 2003 22:06:04 -0000
Organization: Objective Interface System
Subject: Re: Ada CORBA intro?
Newsgroups: comp.object.corba

[On a request for a CORBA tutorial/intro for Ada programmers. --dc]

> A Google search for simply "CORBA Ada" turns up a surprising amount of material. A good start might be www.ois.com.

You can also refer to my Ada Letters article of five or six years ago, before I left MITRE. (I can't recall the specific issue, but could look it up for you). I've also done a more recent article for the COTS Journal (V 3, No. 5, May 2001), but it is more an advocacy article than a tutorial.

> Whether you can get really substantial material free of charge is another matter, as CORBA and Ada is a relatively exotic combination likely to be of interest mainly to engineers and military/aerospace developers.

You might be surprised at the range of application our Ada product has been used in. This includes an ocean-going yacht http://www.webriviera.fr/Technology/Kingcat/index.php, high-energy physics applications http://www.llnl.gov/nif/, ...

Victor Giddings, Senior Product Engineer, Objective Interface Systems

From: pautet@antigone.enst.fr (Laurent Pautet)
Date: Thu, 23 Oct 2003 16:47:59 UTC
Subject: Re: Is CORBA dead for Ada
Newsgroups: comp.lang.ada

[...] Indeed my question is the current status of the mapping.

I must apologize for having started this troll but at least one CORBA tool vendor is now on-line We had the advertising part about their technology, we are now waiting for an answer about the mapping :))

From: Victor Giddings <victor.giddings@ois.com>
Date: Thu, 23 Oct 2003 19:14:29 -0000
Organization: Objective Interface System
Subject: Re: Is CORBA dead for Ada
Newsgroups: comp.lang.ada

> [...] OIS which seems to lead this task force is very shy in its promotion of Ada on its web site.

I will feed this back to our marketing team. We are not shy in our promotion of Ada. In the last OMG Real-Time Workshop in response to a comment, I was accused of being "an unrepentent Ada programmer". At which point, my boss exclaimed "damned right we are"!

> TopGraph'X is still promoting its Ada products. If ORB-River is compatible with CORBA 2.6, it seems that it does not include new features from versions greater than 2.3. Does anyone have fresh news on the OMG activities around the Ada mapping?

Dirk Craeynest was kind enough to repost my responses to a similar thread on comp.object.corba and inform me of this thread. I would like to expand on some of the earlier responses.

There seems to be much apprehension and more than a little misunderstanding associated with CORBA "versions". CORBA is not a monolithic specification but a collection of adopted specifications that may or may not be consolidated into individual documents. The individual documents have associated Revision Task Forces (RTFs) and individual life cycles. The references such as CORBA 2.6 are more properly references to a particular version of the "CORBA Core" specification, a particular document that specifies the language-independent requirements of what an ORB product must implement. Changes to the CORBA Core specification may or may not require changes to the language mapping specifications, depending on whether there is a significant change to the IDL language. Therefore, a lot of revision of the CORBA Core would have been addressed by changing the last digit in the statement that the mapping was "aligned to CORBA version 2.x".

There is understandable confusion about this that has been partly caused by the OMG itself. First of all, the CORBA Core document contains a lot of things, e.g. CORBA/COM Interworking, that don't have to be implemented by an ORB product. The OMG staff have also issued press releases that claimed what the contents of CORBA x.x would contain. In general, this is a problem that needs to be fixed. In addition to being the chair of the Ada RTF, I am chair of a group in the OMG called the Product Specification Definition (psdef) subcommittee that is trying to straighten out publication organization and coordination of versioning. I urge you to participate.

The bottom line is that the fact that the current Ada Language Mapping specification is "aligned to CORBA 2.3" means very little. As I stated in the comp.object.corba post, the OMG IDL language is fairly stable, so few changes in the language mapping are needed. Most of the features added in the CORBA Core 2.4 and later versions are specified in a language independent manner and have not affected the language mapping. So, as Jean-Claude Mahieux was able to report, there has been no hindrance to advancing Ada ORB implementations.

That being said, the other responders are correct in that there has not been an active Ada Mapping RTF is almost 3 years. My only excuse for this is that we have been busy with other OMG specifications like Real-Time CORBA (1.0 and 2.0), Fault Tolerant CORBA, Data Distribution, etc., etc.. […] Nevertheless, there is currently an active RTF attempting to deal with the backlog. Anyone may participate in this activity by joining the email group (ada-rtf-request@omg.org). Non-OMG members may need to contact me to be added to the list. Voting membership (one per organization) in the RTF requires a minimal level of membership in the OMG but, in practice, most voting is pro-forma after a consensus has been worked out among us. Let me know if you are interested in formal membership, so this can be placed on the PTC agenda.

Victor Giddings, Senior Scientist, Objective Interface Systems

Ada and Linux

GNAT 3.15p Package for Debian

From: Ludovic Brenta <ludovic.brenta@insalien.org>
Date: 26 Aug 2003 00:56:45 +0200
Subject: Re: Ada Getting More Shelf Publicity
Newsgroups: comp.lang.ada

[...] Would you be willing to try and build the package I made for GNAT 3.15p on Debian? I don't have access to an Alpha machine to try it on.

You can get my package, as well as several Ada-related others, at http://users.skynet.be/ludovic.brenta. My packages were built on sarge but I think you can compile them on woody as well.

More Ada Packages for Debian

From: Arnaud Rolly <arnaud.rolly@eikonex.net>
Date: Tue, 9 Sep 2003 17:28:46 +0200
Organization: Eikonex
Subject: Paquets des LL Ada
To: ada-france@ada-france.org

[Translated from French: -- dc]

Is there a source of RPM/Deb packages with the most important free Ada software (such as Florist, AdaSocket, AWS, PolyOrb, GtkAda, AdaDoc, ...)? This is an important point for deployment of applications.

(To have a coherent set of packages would really be a plus; the waste of time to install workstations/servers is considerable: to install the correct compiler, the correct support libraries... This hinders the distribution of applications written in Ada.)

Arnaud Rolly, Eikonex, Open Source Engineering, http://www.eikonex.net

From: Lionel Draghi <lionel.draghi@free.fr>
Date: Tue, 09 Sep 2003 23:48:25 +0200
Subject: Re: Paquets des LL Ada
To: ada-france@ada-france.org

The main problem seems to be the lack of a "packager". In your list, half are not packages for Debian. The resumption of the GNAT, GtkAda, and Florist packages by Ludovic Brenta gives hope for a better future.

PS: I am not sure whether his packages are already available by the official ways. For the interested debianers, add "deb http://users.skynet.be/ludovic.brenta testing main" to your /etc/apt/sources.list

From: Lionel Draghi <Lionel.Draghi@Ada-France.org>
Date: Tue, 30 Sep 2003 23:19:23 +0200
Subject: [Ann] Dépot pour les packages Ada Debian
Newsgroups: fr.comp.lang.ada

New Ada packages (GNAT, GPS, GtkAda, ASIS, Florist, etc.) were realized by Ludovic Brenta. While waiting for their integration in the official depots, they are accessible on the server of the Ada-France organization (http://www.ada-france.org/).

The usual magic formula for sources.list:

"deb http://www.ada-france.org/debian ada main" and

"deb-src http://www.ada-france.org/ debian ada main".

File Location Conventions for Ada on Linux

From: Ludovic Brenta <ludovic.brenta@insalien.org>
Date: 10 Oct 2003 19:17:29 +0200
Subject: Re: POSIX File Structure Conventions for Ada
Newsgroups: comp.lang.ada

> I would love to hear what the generally accepted practice is for placing Ada packages on a Unix file system. The library files are already well defined, but where is the standard location to put Ada specs (and bodies for generics) and those GNAT *.ali files, in a POSIX compliant platform? On Linux? In /opt? In /usr/local? If so, where in there?

Florian Weimer has put together a proposal for this. Details are available at http://cert.uni-stuttgart.de/files/ada/gnae/gnae-0.5.html.

[From that page: "The GNU Ada Ada Environment Specification describes a set of requirements for conforming installations of GNAT, the Ada compiler of the GNU Project, and associated tools and libraries. -- dc]

I have followed the recommendations in my Debian packages. Basically:

*.ad[bs] go in /usr/share/ada/adainclude/<package_name>/

*.ali and *.o go in /usr/lib/ada/adalib/<package_name>/

*.a and *.so go in /usr/lib

And I've added:

*.gpr go in /usr/share/ada/adainclude/

GNAT 5.01a Like RPM for Redhat 9

From: Jeff C <jcreem@yahoo.com>
Date: Fri, 10 Oct 2003 17:26:08 GMT
Subject: Announce: GNAT 5.01a Like RPM for Redhat 9 Available
Newsgroups: comp.lang.ada

This is a first build of something that resembles GNAT 5.01a as an RPM for Redhat 9.

You can download it from Adaworld from the links page http://www.adaworld.com/linksmain.html [select "Ada Downloads Websites" -- dc]

GNAT 5.01a is based on gcc 3.2 technology but it is not the same as building an out of the box configuration of gcc 3.2.

GNAT 5.01a is created by patching gcc 3.2 with a set of Ada Core Technologies patches for the core gcc components and by replacing the entire Ada subdirectory with the contents of the CVS tag 5.01a from http://libre.act-europe.fr/GNAT/

This was built following the build instructions from the README.BUILD (which lives in the CVS archive) with the following exceptions:

1) The README.BUILD directs you to re-use 3 libraries from your existing binary distribution in order to create the final gnat tools. (libaddr2line.a libbfd.a and libiberty.a). I did not do this since I ran into linking issues on Redhat 9 when I tried this (although it appears to work under Solaris). In addition, I was not wild about not really building everything from source. So, I downloaded a copy of binutils-2.13.2.1, patched it with a patch file from the GNAT CVS archive (although in this case to get the patch I had to use a file from the CVS head - version 1.6 of the file called binutils-2.13.2.1.dif).

2) The instructions in README.BUILD do not actually create an RPM, they just do a standard install via make install. Since I have never created an RPM before I took the easy way out and used a program called checkinstall to build it. Basically checkinstall is a program that monitors the progress of a make install and creates an RPM semi-automatically from the files that get installed during the process.

I wanted this RPM to install so that it had no chance of breaking any existing installations so everything will be installed under gnat-501a-jmc

So, you MUST put /usr/gnat/gnat-501a-jmc/bin in your path for this to work. If you already have a gcc installed I recommend putting this path entry at the start of your path to be sure you really are running it.

As for testing...I have done hardly any testing of this at all. I did install the RPM on a separate machine and verified I could build and run hello world. I also built and ran (on the original machine) a simple tasking program that did I/O from a couple of tasks to see that the tasking runtime at least appeared to be semi-functional.

So, let me know if you run into anything.

Since I did not modify any of the sources I am not at this time actually including a source library of all these GPL components.

[And from a later message: -- dc]

> Works well for me! Tested with AWS-1.3, Adasockets-1.8 and XmlAda-0.7.1

Glad it is working!

Note that the RPM has been temporarily pulled so I can have this properly identify itself as something like 5.01p but other than some version/nomenclature stuff there is really nothing particularly wrong with the existing RPM.

I hope to get a new RPM (along with co-located source code instead of having the source on a different server) this weekend.

From: Jeff C <jcreem@yahoo.com>
Date: Fri, 17 Oct 2003 22:01:30 GMT
Subject: Re: GCC 5.01p (was: Re: gcc/gnat 3.3)
Newsgroups: comp.lang.ada

[In response to "when I re-release my properly marked GNAT 5.01p for Redhat 9 x86" Ludovic Brenta -- su]

> I am suspicious about this "5.01p" version number. In my understanding, the "p" stands for something like "published" or "public", i.e. it denotes one of ACT's stable releases. Since I am not aware of any 5.01p official release by ACT, I would suggest you use a version number that denotes that this is from CVS, such as e.g. 5.01.cvs.20031017.

Actually p stands for public. Since this is not just a random CVS grab at a random time but a pull of the tagged CVS files that were "released" for 5.01 it seems pretty reasonable. Note though that the RPM and the directory that it gets installed in is actually 5.01p-jmc to further

differentiate it from real real version that may or may not some day be publically released by ACT.

Finally, I have discussed it with ACT and they were OK with calling it 5.01p which seems good enough for me.

But of course you are right to be suspicious about this!! Who knows who I am or what evil lurks in the heart of men. So, if you want a real 5.01 where someone stands behind what is actually in it I suggest you pay several thousand dollars to ACT and get a 5.01a since even ACT does not promise anything in particular (other than it being good for student and research use) about the "p" releases.

RTLGnat 1.0 - GNAT Port for Real-Time Linux OS

From: Jorge Real <jorge@disca.upv.es>
Date: Tue, 02 Sep 2003 19:53:34 +0200
Organization: DISCA-UPV
Subject: Announce: RTLGnat 1.0 released
Newsgroups: comp.lang.ada

RTLGnat version 1.0 is now available from the following web site:

http://rtportal.upv.es/apps/rtl-gnat/. It works with Gnat 3.14p or 3.15p and RTLinux 3.1, 3.2pre1, 3.2pre2 or later (as long as the POSIX interface is preserved).

RTLGnat is a port of the GNAT Ada compiler for the Real-Time Linux operating system. An Ada program can be compiled as a Linux loadable kernel module, where the program's tasks run as RTLinux threads, with a higher priority than any other running Linux application.

[See also "RTLGnat - GNAT Port for Real-Time Linux OS" in AUJ 24.2 (Jun 2003), p.88. -- dc]

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

DLibs 0.1 - Binding to Linux Library Loader

From: Chris Campbell <chris.danx@ntlworld.com>
Date: Tue, 09 Sep 2003 18:09:39 +0200
Subject: [Ann]: Binding to Linux Library Loader - DLibs 0.1
Newsgroups: comp.lang.ada

Version 0.1 of DLibs is now available at www.cyberdanx.co.uk, complete with documentation and examples. DLibs is a binding to the Linux Library Loader, allowing Ada 95 programs to load and utilise libraries at runtime!

Suggestions, complaints, notes of omissions, ... are all welcome. [...] Thanks to Simon Knipe for helping with the binding way back in January (when it should have came out, had I not removed the wrong partition :().

Plugins in Ada Programs

From: kat-Zygfryd <6667@wp.pl>
Date: Sat, 4 Oct 2003 01:47:43 +0200
Subject: DLLs / shared objects
Newsgroups: comp.lang.ada

I'd like to have a plugin system in my Ada program – loading functions/procedures from dll's/so's unknown at compile time, but found by the app at runtime. All information I found concerned only Windows DLLs and involved statically linking parts of the dynamic libs, which is unacceptable. Is there any way in Ada to do so? (GNAT 3.14p, x86 Windows/Linux) [...]

[All following responses were sent on 4 Oct 2003 as well. -- dc]

[From: Jerry van Dijk <jdijk@acm.org>]

Yes, the same way as you would with C or C++. This is as you probably realize not a language but an OS question.

I never did this under Linux, but under Windows you can use the Win32 API to dynamically load DLL's and get pointers to their function. Or you can use GNATCOM if you want to use (D)COM objects instead. [...]

Jerry van Dijk, Leiden, Holland

[From: chris <chris.danx@ntlworld.com>]

Linux Dynamic Library support is provided by "dl" (man dlopen), a binding can be found at http://www.cyberdanx.co.uk [see previous news item -- dc].

[From: Patrice Freydiere <frett27@free.fr>]

There is a solution for each system supporting dynamic library loading. You can use the Win API to do so for Windows platform. There is an example in the AdaPower source repository.

There is a AdaPlugin project for Linux that uses Dynamic Glib Module functionality (I don't remember the URL) or Dlib (http://www.ada-france.org/article94.html)

I never saw an Ada project that makes an abstraction and proposes a common approach of "plugin" that manages the different platforms.

[From: chris <chris.danx@ntlworld.com>]

Try www.cyberdanx.co.uk and look for dlib. It's not high level but it does bind to Linux library loading facilities, and might serve as the basis for a higher level binding.

GNAT Tasking on Red Hat 9.0

From: mcq95@earthlink.net (Marc A. Criley)
Date: 7 Oct 2003 06:33:11 -0700
Subject: Re: Redhat Linux 9.0 and Gnat
Newsgroups: linux.redhat.install,comp.lang.ada

> [...] I believe there are some tasking issues with Redhat 9 and GNAT due to the threading changes [in] RH9.

Yes, at least for GNAT 3.15p. I've not tried any of the GCC 3.x versions of GNAT, so can't make any statements regarding that. Here's a link to the info you need for GNAT 3.15p and RH 9: http://groups.google.com/groups?selm=254c16a.0307261224.347f4fe2%40posting.google.com&output=gplain

Marc A. Criley, mc@mckae.com, www.mckae.com

From: Mike Card <thehouseofcards@mac.com>
Date: Wed, 08 Oct 2003 04:47:12 GMT
Subject: Re: Redhat Linux 9.0 and Gnat-tasking bug
Newsgroups: linux.redhat.install,comp.lang.ada

FWIW, I think you'll find that Ada tasking does not work with the GNAT that comes installed by default in RH9. I used the GNAT that came with RH8 and it worked great, but the size of the data structure for tasks changed in RH9 and so GNAT's task control blocks are the wrong size. So if you have a program that uses tasking you will have problems.

We fixed this on our RH9 installation by downloading LOTS of RPMs for gcc and installing them ourselves after finding a newsgroup posting when doing a Google search for information about this problem. I say LOTS because of dependencies we encountered that kept requiring more and more RPMs. Anyway, to our amazement we did as the posting recommended and voila our GNAT compiler is producing good code for programs with tasking. Unfortunately I couldn't find the link just now, but here is another thread that describes the problem:

http://www.geocrawler.com/archives/3/84/1996/1/50/273331/

Ada and Microsoft

Windex Binding Updated for Windows 2000

From: Stephen Leake <Stephe.Leake@nasa.gov>
Date: 27 Oct 2003 12:00:52 -0500
Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)
Subject: Windex
Newsgroups: comp.lang.ada

Well, in the spirit of "never say never", I got tired of waiting to have time to port my favorite Windex application to GtkAda, and decided to see how hard it would be to fix Windex on Windows 2000. Took about an hour; MS had added a field to the MenuItem record. Of course, I should have had a test function that checked the lengths of all C structs I imported, but that would have meant main

taining some C code with MS Visual Studio; one of the reasons I gave up on Windex.

[See also "Windex 1.04 - Thick Binding to Win32" in AUJ 21.1 (Apr 2000), p.30, and many references in more recent issues. Release 1.06 is at http://www.toadmail.com/~ada_wizard/ -- dc]

Anyway, if you want the fix, let me know. This is one way of gauging interest in Windex . If there is sufficient interest, I'll do a release (the fix is _not_ on my website).

[Stephane Richard <stephane.richard@verizon.net> responded: -- dc]

Well you got my interest. But I think you knew that already. . [...] I learned a lot of my Ada programming from Windex and SAL when I started along with a few other examples and some PDF files :-)... He's my mentor and he doesn't even know it .

Stéphane Richard, "Ada World" Webmaster, http://www.adaworld.com

[Stephen Leake replied: -- dc]

Thanks.

But this raises a general point. If you like/use an open source project, you should provide feedback to the authors. Often the only feedback we get is bugs. For all I know, there are hundreds of people using Windex, and they've never had a problem . I'd like to hear from them. [...]

References to Publications

An Interview with A. Stepanov

From: Matthew Heaney <matthewjheaney@earthlink.net>
Date: Sun, 03 Aug 2003 16:42:26 GMT
Subject: Re: XML DOM Binding for Ada 95 - matter of style
Newsgroups: comp.lang.ada

> I see tagged types as the way to fully implement the Object Oriented paradigm in Ada.

To help convince you that inheritance-oriented programming is not the solution to all problems, I suggest you read this interview with Stepanov: http://www.stlport.org/resources/StepanovUSA.html

Here are the money quotes:

"I find OOP technically unsound. It attempts to decompose the world in terms of interfaces that vary on a single type. To deal with the real problems you need multisorted algebras - families of interfaces that span multiple types. I find OOP philosophically unsound. It claims that everything is an object. Even if it is true it is not very interesting - saying that everything is an object is saying nothing at all. I find OOP methodologically wrong."

Asked what he thought of Java, he replied that "...It keeps all the stuff that I never use in C++ - inheritance, virtuals - OO gook – and removes the stuff that I find useful."

He says later that "...I spent years trying to find some use for inheritance and virtuals, before I understood why that mechanism was fundamentally flawed and should not be used."

Burns & Wellings - "Real-Time Systems and Programming Languages"

From: Richard Riehle <richard.riehle@adaworks.com>
Date: Sun, 03 Aug 2003 11:24:35 -0700
Organization: AdaWorks Software Engineering
Subject: Announcement: Burns and Wellings Book
Newsgroups: comp.lang.ada

I just acquired the current edition of the book by Alan Burns and Andy Wellings, Real-Time Systems and Programming Languages, Addison-Wesley, ISBN 0-201-72988-1, Third Edition. This update to their already great Second Edition is important for anyone interested in Real-time Systems. It covers Ada, Java RT, C, Occam, Posix, and others. Somehow, it has not been as well-known in our immediate and direct visibility as some other books. The price is $60.00 USD, but I think it is worth every penny.

[Although published in March 2001, this important book wasn't mentioned in the AUJ news section until now. -- dc]

New Version of "Ada Distilled" On-line Book

From: Richard Riehle <richard.riehle@adaworks.com>
Date: Mon, 04 Aug 2003 19:11:42 -0700
Organization: AdaWorks Software Engineering
Subject: Announcement: Revised Ada Distilled
Newsgroups: comp.lang.ada

A new version of Ada Distilled is available at http://www.adaic.org & http://www.adapower.com free for downloading. It is about 100 pages as a PDF file. There is also a ZIP file containing all the source code for the book. The newest version uses color text extensively in both comments and code examples. Also, I have commented, in color, several of the more commonly used packages in the ALRM and Annexes.

This is freeware. You may copy it, forward it, print it, or whatever.

I have been asked by people from several non-English language countries for permission to translate Ada Distilled. If you are one of those people currently translating Ada Distilled, please remind me to send you the MS-Word version so your task will be easier.

[See also "Ada Distilled - On-line Book Updated" in AUJ 24.2 (Jun 2003), pp.89-90, and "Feedback on Ada Distilled Online Book" in AUJ 24.1 (Mar 2003), p.8. -- dc]

Arthur Griffith - "GCC: The Complete Reference"

From: Warren W. Gay VE3WWG <ve3wwg@cogeco.ca>
Date: Tue, 19 Aug 2003 09:28:15 -0400
Subject: Ada Getting More Shelf Publicity
Newsgroups: comp.lang.ada

I was happy to see the beginnings of fruit from putting Ada (GNAT) into the GCC 3.x compiler. The book "GCC: The Complete Reference" will expose more GCC readers, to the existence of Ada and get the curious started in it. There is a table of contents available in the [information on www.amazon.com about the book -- dc].

The Ada sections are short, and the interfacing C to Ada section is very short, but at least readers will become more aware that it is there, and that it can coexist in a C world.

Warren W. Gay VE3WWG, http://home.cogeco.ca/~ve3wwg

From: Arthur Griffith <arthur@belugalake.com>
Date: Fri, 22 Aug 2003 18:04:56 GMT
Organization: ACS Internet - complaints to abuse@acsalaska.net
Subject: Re: Ada Getting More Shelf Publicity
Newsgroups: comp.lang.ada

The subject of the book is GCC and all of its languages. I didn't try to dig into any of the languages (other than some of the GCC variations to them), but only to explain how to install and run the compiler. Actually, I had to do a bit of extra work in describing Ada because the installation of it is in two phases--first by installing the GNAT compiler and then by installing the GCC Ada compiler. That was necessary because some of the source is in Ada. [...]

Warren, thanks for checking out the book.

From: Warren W. Gay VE3WWG <ve3wwg@cogeco.ca>

Date: Mon, 25 Aug 2003 12:27:26 -0400

Subject: Re: Ada Getting More Shelf Publicity

Newsgroups: comp.lang.ada

Nevertheless, it was encouraging to see a GCC book include sections about Ada, brief as they may have been. Half of the battle is giving people Ada to work with, and another significant portion is demon-

strating that it can be used with C. So I think that this is a good start. [...]

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, 2 Sep 2003 15:03:04 -0700 (MST)
Subject: Real-Time Industry Updates - News from DDC-I
To: T9DK Sept 2003 Online News <jcdk@ddci.com>

DDC-I Online News. August/September 2003, Volume 4, Number 6 - [http://www.ddci.com/news_vol4num6.shtml] A monthly news update dedicated to DDC-I customers & registered subscribers.

Embedded C++ Now Supported by DDC-I's SCORE IDE. More options for customers developing safety-critical systems.

3rd Party Update: The real-time operating system buying decision.

TADS-68xxx Windows Migration Package. A budget conscious solution for current TADS customers.

Keeping the Customer Happy. How to be successful in customer relationship management. [...]

From: jc <jcdk@ddci.com>
Date: Wed, 1 Oct 2003 15:53:40 -0700 (MST)
Subject: Real-time Industry Updates - News from DDC-I
To: U9DK Oct 2003 Online News <jcdk@ddci.com>

DDC-I Online News. October 2003, Volume 4, Number 7 - [http://www.ddci.com/news_vol4num7.shtml]

DDC-I's SCORE IDE Now Offers Windows Native Capability. Start Software Development and Testing Before Hardware is Available!

3rd Party Update: The CsLEOS RTOS - A Technical Overview.

Edison and The Phonograph - Lessons to Learn From. Being First or Being Best Does Not Matter. What Matters is What Your Customers Think & Need. [...]

From: jc <jcdk@ddci.com>
Date: Fri, 31 Oct 2003 11:29:59 -0700 (MST)
Subject: Real-time Industry Updates - News from DDC-I
To: V9DK Nov 2003 Online News <jcdk@ddci.com>

DDC-I Online News. November 2003, Volume 4, Number 8 - http://www.ddci.com/news_vol4num8.shtml]

Rod Chapman, SPARK Team, Praxis Critical Systems

Royalty Free Subscription Pricing Now Available from DDC-I. A flexible pricing option aimed directly at the customer's bottom line!

3rd Party Update: FAA DO-178B Training - November 6th & 7th. [...]

Tech Talk: A simple way to debug Ada elaboration code.

Pair Programming. A powerful addition to the software development toolbox.

For this months complete newsletter, go to http://www.ddci.com/news_current_issue.shtml [...]

Thomas Bergin et al - "History of Programming Languages"

From: Matthew Heaney <mheaney@on2.com>
Date: Wed, 8 Oct 2003 18:17:40 -0400
Subject: Re: Question about historical fairness
To: team-ada@acm.org

> [...] I feel it isn't possible to get an unbiased history of computer languages. I would like to see what contributions were made by whom at what time - and honestly - without hype, spin, marketing, or other lies. Does anyone know if this exists?

You can read about the development of the Ada programming language in History of Programming Languages, Volume 2, by Thomas J. Bergin et al.

[See http://www.csis.american.edu/tbergin/pubs/programming.html for the table of contents. -- dc]

Adrian Hoe - "Software Development Reengineering - An Experience Report"

From: Adrian Hoe <mailbox@adrianhoe.com>
Date: Thu, 23 Oct 2003 10:45:42 +0800
Subject: Solid facts to promote Ada
Newsgroups: comp.lang.ada

I've just made available my paper, "Software Development Reengineering - An Experience Report", on my web site adrianhoe.com [...].

This paper was published in the conference proceeding of Ada-Europe 2002. It has real numbers to proof that Ada is a better language over others, instead of statements claiming Ada is better.

This paper may complement Jack Ganssle's article "My love-hate relationship with C" in Embedded.com (see Volkert's post on Oct 21).

[See also "Embedded.com on Love-Hate Relationship with C" further in this AUJ issue. -- dc]

Hardcopy of Ada RM and Related Documents

From: dirk@cs.kuleuven.ac.be (Dirk Craeynest)
Date: 24 Oct 2003 08:24:55 +0200
Organization: Ada-Belgium, c/o Dept. of Computer Science, K.U.Leuven
Subject: Re: Hard copy of Ada RM (was: Hex ouput)
Newsgroups: comp.lang.ada

> [...] as I have yet to get a hardcopy one for '95. Do you know where I can purchase one? I hate trying to read PDF (et. al.) files on my computer. I much prefer books.

Both the Consolidated Ada Reference Manual and the Ada 95 Rationale have been reprinted by Springer in their Lecture Notes in Computer Science series, and are readily available.

Full information can be found on the AdaIC web-page "Accessing the Ada Language Reference Manuals" at URL http://www.adaic.com/standards/articles/lrm.html (see subsection "Ordering Print Copies").

Note that the document "Ada 95 Quality and Style Guide: Guidelines for Professional Programmers" is also reprinted by Springer, should you be interested.

Dirk (Dirk.Craeynest@cs.kuleuven.ac.be for Ada-Belgium/Europe e-mail)

*** Intl. Conference on Reliable Software Technologies - Ada-Europe'2004

*** June 14-18, 2004, Palma de Mallorca, Spain ** www.ada-europe.org ***

CrossTalk - Papers on Ravenscar and on Static Code Analysis

From: rod.chapman@praxis-cs.co.uk (Rod Chapman)
Date: 31 Oct 2003 00:55:33 -0800
Subject: Ada article in CrossTalk Journal
Newsgroups: comp.lang.ada

I'm sure c.l.a readers will find the latest issue of CrossTalk Journal of interest. There are two very Ada-friendly articles in there:

"The Ravenscar Profile for Real-Time and High Integrity Systems" by Dobbing and Burns,

"Software Static Code Analysis Lessons Learned" by Andy German of QinetiQ Boscombe Down.

The latter paper presents years of findings regarding the static analysis of all sorts of aircraft systems. Should be required reading, especially if your current project name starts with a "J" and ends with an "SF".

You can find these at www.stsc.hill.af.mil

From: Rasmussen Karen J Contr OO-ALC/MASEA <Karen.Rasmussen@hill.af.mil>
Date: Mon, 3 Nov 2003 08:41:52 -0700
Subject: The November 2003 Issue of CrossTalk is now available on-line.
To: Dirk.Craeynest@offis.be

The November 2003 issue of CrossTalk, The Journal of Defence Software Engineering is now available on our Web site at: <www.stsc.hill.af.mil>.

This month our theme is "Development of Real-Time Software."

[...] Next is "The Ravenscar Profile for Real-Time and High Integrity Systems" by Brian Dobbing and Alan Burns. These authors present the Ravenscar model for building safe and reliable real-time systems. They explain how developers using this profile can establish high confidence levels in concurrency properties and requirements within international standards early in the development life cycle.

In "Software Static Code Analysis Lessons Learned," author Andy German shares his experiences from developing safety-critical, real-time systems. He defines static code analysis, reviews some of the tools, and shares lessons learned at The United Kingdom Ministry of Defence. [...]

Pam Bowers-Palmer, Managing Editor

Ada Inside

France - Electronic Document Shelf

From: Arnaud Rolly <arnaud.rolly@eikonex.net>
Date: Fri, 5 Sep 2003 17:31:06 +0200
Organization: Eikonex
Subject: Base de données XML
To: ada-france@ada-france.org

[Extracts translated from French. -- dc]

Being new on this group, I present myself: my name is Arnaud Rolly, and I am an engineer (software) in an enterprise in Toulouse, Eikonex. Eikonex decided to use Ada for its internal R&D projects, projects which should lead thereafter to commercial offers, also developed in Ada. [...]

[And from a later message: -- dc]

> That's interesting news. Is it possible to know more? Is this an internal choice, or is this to score with respect to the potential customers of your products? Which platforms are envisaged?

The choice of Ada arises from several internal concerns on two points: quality (vast field, Ada is only one link in the chain here), maintenance (previous development experience showed large shortcomings in the preceding choices, RIP).

It is interesting to note that without certain tools like AWS, Ada would not have been selected: we support "mature" technologies directly bringing many integrable components; it is not viable to redevelop the low layers of the protocols, for example.

Ada could also become a commercial argument, but that is not on the agenda now.

Eikonex, as far as possible, only uses free/open source components to offer to its customers perennial solutions, on the Linux platform [...].

Arnaud Rolly, Eikonex, Open Source Engineering, http://www.eikonex.net

China - Super 7 Jet

From: Adrian Hoe <mailbox@adrianhoe.com>
Date: Sun, 07 Sep 2003 21:54:34 +0800
Subject: Re: F22 Raptor in slashdot.org
Newsgroups: comp.lang.ada

The Chinese new generation jet fighter rolled out recently in Cheng-du has its fly-by-wire and combat and target management systems developed in Ada. This magnificent Super 7 (formerly known as FC 1) has many modern architectures including software.

Indirect Information on Ada Usage

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

From: Patrick Farail <patrick.farail@airbus.com>
Date: Wed, 06 Aug 2003 16:53:19 +0200
Subject: TR: Protected operation and IT handling
To: ada-france@ada-france.org

[...] If somebody could help us it will be very nice. [...] We use GNAT 3.13p under Sun/Solaris 2.x. [...]

Patrick Farail, Airbus France, Software Engineering Methods

URL: http://www.jobscareer.be/[...]
Date: Sat, 27 Sep 2003 00:04:21 +0200 (MEST)

IT & Telecom - region Brussels - sector Telecommunication, ICT & Internet

New! 4 Ada Software Engineers [...]

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

Profile: [...] Knowledge of a programming language preferably Ada 83-95 or C++. Developer & analyst designer. Team spirit and good methodology.

From: oliderid@yahoo.co.uk (Olivier Laurent)
Date: 7 Oct 2003 06:10:58 -0700
Subject: a good book to start with?
Newsgroups: comp.lang.ada

My background: Perl, Java, C++. [...] I was impressed by some tutorials I've seen over the web about Ada. And most importantly the kind of application you can make with. I would like to know more, just a hobby. [...] On the linux OS, [...]

From: Stephen D. B. Wolthusen <wolt@igd.fhg.de>
Date: Tue, 7 Oct 2003 18:44:10 +0200
Subject: Anyone interested in Ada-related European research project?
To: team-ada@acm.org

The following is pertinent only for European Union and affiliated nations (e.g. Switzerland, Israel) because of funding regulations.

We are currently putting together a consortium for a submission to the European Union's 6th framework program in the area of secure mobile applications with significant portions of the systems to be developed in Ada (for reasons of high assurance / reliability). While the core of the consortium is already established, I would particularly welcome small and medium enterprise participants, but would also invite research/university participation.

Please contact either myself (wolt@igd.fhg.de) or Volker Roth at (vroth@igd.fhg.de) at your earliest convenience if you're interested.

Thanks!

Stephen Wolthusen, Fraunhofer-IGD, Darmstadt, Germany

From: tmoran@acm.org
Date: Wed, 08 Oct 2003 02:08:55 GMT
Subject: Re: graphics in ada
Newsgroups: comp.lang.ada

> My professor said that graphics were not possible in Ada. [...]

I guess the video editor I built in DOS with Ada 83 in 1992 must have not been possible. Or the teleprompter that I posted on Compuserve. Sigh.

From: Marc Pelletier <marc@goldak.ca>
Date: Fri, 24 Oct 2003 20:05:13 -0000
Organization: Goldak Exploration
Subject: Realtime and Ada - stupid newby question
Newsgroups: comp.lang.ada

I am planning a realtime application for data acquisition on a pc104 platform and investigating my OS options. I'm normally a Delphi programmer, and rather than the pain of boning up on my very poor C/C++ skills, I would like to learn Ada for this project. [...]

Marc Pelletier, Goldak Exploration

From: Adrian Hoe <mailbox@adrianhoe.com>
Date: Sat, 25 Oct 2003 11:48:27 +0800
Subject: Re: Solid facts to promote Ada
Newsgroups: comp.lang.ada

[...] Right now, I have some more important job to attend to. One of it is to prepare course material for a Basic Ada Hands-on Workshop in a university in Malaysia which I gave a seminar 10 days ago. [...]

From: MIKE/MAR-EL <mike@bestjobtoday.com>
Date: Thu, 30 Oct 2003 18:58:34 GMT
Subject: Sr Ada SW Engineering openings
Newsgroups: comp.lang.ada

Experienced Senior SW Engineers with Ada development skills needed to work on various defence related programs. Coding, integration, Team Lead etc.

Current permanent job openings in NY and RI. Relocation packages will be offered.

Other desirable skills include: C/C++, OO, CORBA, RTOS-VxWorks, Combat Control Systems experience, Rational Apex.

Active Secret clearance is a plus but not always necessary. US citizenship is required.

Applications Written in Ada

From: Lars <larsm@gmx.net>
Date: Wed, 29 Oct 2003 09:16:08 +0100
Subject: Applications written in Ada
Newsgroups: comp.lang.ada

I am aware of the (alleged) fact that Ada is mainly used for the development of safety-critical applications by governmental institutions. I found a list of military, space, and aviation projects realised in Ada, and it was not unimpressive.

[See also "Ada in Fielded Military Systems" in AUJ 24.2 (Jun 2003), pp.96-97, and "List of Real-World Projects Powered by Ada" in AUJ 23.3 (Sep 2002), p.154. -- dc]

However: Is there a list of applications written in Ada that are not intended for defence, space, aviation? Which Ada applications (especially for Win32) are there for which the source code is available?

From: Ross Higson <rosshigson@optusnet.com.au>
Date: Wed, 29 Oct 2003 20:49:05 +1100
Subject: Re: Applications written in Ada
Newsgroups: comp.lang.ada

> Is there a list of applications written in Ada that are not intended for defence, space, aviation?

Try AdaWorld (http://www.adaworld.com/) and look around the 'Ada Projects' section.

> Which Ada applications (especially for Win32) are there for which the source code is available?

I have telnet and serial communications applications specifically written for Win32 in Ada - see the 'libraries' section of Ada World (they're in that section because the applications are mainly intended as test programs for the terminal emulation library). However, they are both complete implementations, and the source is available.

[See also "Ada Terminal Emulator for Windows" in AUJ 24.3 (Sep 2003), pp.150-151. -- dc]

There are other Win32 applications listed on the site, but sometimes it is not obvious exactly what platforms are supported. If you feel this is a serious omission (I never thought of it before!) it might be worth making this point via the feedback link on the site.

From: Peter Amey <peter.amey@praxis-cs.co.uk>
Date: Wed, 29 Oct 2003 11:04:53
Subject: Re: Applications written in Ada
Newsgroups: comp.lang.ada

> [...] applications written in Ada that are not intended for defence, space, aviation?

You could take a look at the IEEE paper describing the Mondex smart card certification authority downloadable from http://www.sparkada.com/downloads/ieeesw.pdf. This describes a Windows NT hosted system written in Ada 95, SPARK (and C++ for the GUI) which is none of defence/space/aviation being financial/security.

[See also "Ada and Cryptography" in AUJ 23.3 (Sep 2002), pp.139-140. -- dc]

From: Randy Brukardt <randy@rrsoftware.com>
Date: Wed, 29 Oct 2003 14:17:35 -0600
Subject: Re: Applications written in Ada
Newsgroups: comp.lang.ada

> [...] Ada applications (especially for Win32) [...] for which the source code is available?

Win32-specific applications typically use a thick library like Claw, GWindows, or Windex. Look for apps using those bindings on the Ada-wide search engine: www.adaic.com/site/wide-search.html. I know that there are number of small applications using Claw out there. (The Claw GUI builder is a Claw application, in fact. But no source available on that.)

From: Jano <nono@celes.unizar.es>
Date: Wed, 29 Oct 2003 21:24:17 +0100
Subject: Re: Applications written in Ada
Newsgroups: comp.lang.ada

My current hobby project: http://agio.sf.net.

[From that page: "Adagio aims to provide a server-only solution for the recently published Gnutella2 protocol (www.gnutella2.com). -- dc]

From: Gautier de Montmollin <gdemont@hotmail.com>
Date: Wed, 29 Oct 2003 21:43:13 +0100
Subject: Re: Applications written in Ada
Newsgroups: comp.lang.ada

> Is there a list of applications written in Ada that are not intended for defence, space, aviation?

On these lists you'll find (also) non defence/space/aviation applications: http://www.seas.gwu.edu/~mfeldman/ada-project-summary.html and http://www.acm.org/sigada/education/pages/success.html

> [...] Ada applications (especially for Win32) [...] for which the source code is available?

Not looking too far I can cite: http://www.usafa.af.mil/dfcs/bios/mcc_html/adagide.html and http://www.mysunrise.ch/users/gdm/texcad.htm

From: Ed Falis <falis@verizon.net>
Date: Wed, 29 Oct 2003 21:55:01 GMT
Subject: Re: Applications written in Ada
Newsgroups: comp.lang.ada

AstroFrames, an astrology program that works on Windows and various Unices: http://mysite.verizon.net/vze6qirr/myindex.html

From: Jeffrey Carter <jrcarter@acm.org>
Date: Thu, 30 Oct 2003 04:49:56 GMT
Subject: Re: Applications written in Ada
Newsgroups: comp.lang.ada

Mine Detector, a mine-finding game for Windows and Linux: http://home.earthlink.net/~jrcarter010/mindet.html. [See also "Mine Detector Game" in AUJ 22.4 (Dec 2001), p.205. -- dc]

From: Erlo Haugen <elh@terma.com>
Date: Thu, 30 Oct 2003 11:28:29 +0100
Organization: Terma A/S
Subject: Re: Applications written in Ada
Newsgroups: comp.lang.ada

> Which Ada applications (especially for Win32) are there for which the source code is available?

The GNU visual debugger GVD is written in Ada [and is now integrated in the GNU Programming System GPS -- dc].

From: tmoran@acm.org
Date: Thu, 30 Oct 2003 19:19:57 GMT
Subject: Re: Applications written in Ada
Newsgroups: comp.lang.ada

> Which Ada applications (especially for Win32) are there for which the source code is available?

What constitutes an "application"? Is, for instance, a web crawler that reports dead links, an application? How about a program that reports on the Ada.Calendar.Clock timing characteristics of a particular compiler/OS? I would call a teleprompter program an application, but I'm not so sure about a program that

returns off the internet and displays them on a community cable TV system?

[See also "Finder - Ada Web Crawler" in AUJ 22.4 (Dec 2001), p.200, "Time the Clock", in AUJ 21.4 (Jan 2001), p.227, and "USA – Local Election Results on Saratoga Community Access Television" in AUJ 24.1 (Mar 2003), p.28. -- dc]

From: Patrice Freydiere <frett27@free.fr>
Date: Thu, 30 Oct 2003 23:22:01 +0100
Subject: Re: Applications written in Ada
Newsgroups: comp.lang.ada

My personal img server contribution. http://imgsvr.tuxfamily.org. [See also "Imgsvr - Personal Picture Web Server" earlier in this AUJ issue. -- dc]

From: Martin Dowie <martin.dowie@btopenworld.com>
Date: Fri, 31 Oct 2003 13:18:02 UTC
Subject: Re: Applications written in Ada
Newsgroups: comp.lang.ada

Off the top of my head:

1. MediaGuard from Canal+ is an Ada application used for secure pay-per-view. They do the broadcasting using Ada too I believe. [See also "Word-wide / MediaOne & Canal+ - TV Head-ends Systems" in AUJ 21.1 (Apr 2000), pp.36-37. -- dc]

2. Philips Semiconductors control/monitors their fabrication plants using Ada. [See also the "OESM Server" subtopic in "Ada Web Server" in AUJ 23.2 (Jun 2002), pp.71-74. -- dc]

No sources though of course - but then it wouldn't matter what the language used was for these things. Companies _rarely_ publish source code.

Ada in Context

On Ada Language Improvements

From: Tom Moran <tmoran@acm.org>
Date: Wed, 27 Aug 2003 04:46:25 GMT
Subject: comment: Ada 83-95
Newsgroups: comp.lang.ada

I recently needed to write some utilities to run under 16 bit MSDOS (for data recovery of a badly damaged W2K NTFS disk), so I dug out an antique PC-AT class computer and used its Ada 83 compiler.

I was surprised by the importance of some of the "little changes" in Ada 95 vs 83 like declaration order requirements, mixing named and "others" in aggregate assignments, limited 'image, and, of course, "use type".

From: Stephen Leake <Stephe.Leake@nasa.gov>
Date: 29 Aug 2003 12:58:25 -0400
Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)
Subject: Re: comment: Ada 83-95
Newsgroups: comp.lang.ada

Yes, Ada 95 is a much better language than Ada 83, in lots of little ways as well as the obvious big ones. But, you can compile almost all Ada 83 source with an Ada 95 compiler. I think that's a remarkable achievement by the Ada 95 team!

Ada for a Programming Newbie

From: Ludovic Brenta <ludovic.brenta@insalien.org>
Date: 14 Sep 2003 23:29:02 +0200
Subject: Re: Ada for a programming newb.
Newsgroups: comp.lang.ada

> I was wondering if Ada would be suitable for me. I've started to learn Java (and I actually understood most of it) but just gave up cause I found more interesting things to do and it seemed that I never could make time for it. But now I've got plenty of time and want to learn to program again. There doesn't seem to be as much documentation (so I'm a little confused as to how one'd go about learning it) as in some other languages but it is very easy to read and looks very Pascal-like (I know because I've been looking at quite a few languages). What do you think?

Programming is inherently difficult. Among other lessons, experienced programmers have learned that "the devil is in the details". You may get the general picture right, but you may (and indeed will) make small mistakes in the details and your programs will fail as a result.

Different languages handle this problem differently. Some languages choose to be lax and forgiving to the beginning programmers, so as to appear to be easy-going. The price to pay is that they do not detect your mistakes, and you have to look for them yourself when your program fails for obscure reasons (which is called "debugging"). This is for example the case of C and, to a lesser extent, Java.

By contrast, Ada is an old, picky lady who reviews your work very closely and tries to point out your mistakes (she was the first programmer in history, so she should know a good program from a bad one). If you choose Ada, you will naturally learn to think straight, and to be precise and systematic. Those are good skills for a programmer, which you can later apply to all programming languages. I contend that an Ada programmer can adapt to pretty much any language very easily, but the converse is not true. If you start with Java, the learning may at first seem easier but in fact you will learn less. I guess it is up to you to decide which style suits you best.

There are intermediates between Java and Ada. I would list just a few languages here, but there are many more that you may want to explore.

From the most lax to the most strict:

C, Java, C++, Pascal, Modula-3, Ada.

If after reading this you are still interested in Ada, there is an excellent book by John English, which is by the author's own words "aimed (...) squarely at the beginning programmer learning Ada 95 as a first language". Here is the URL: http://www.it.bton.ac.uk/staff/je/adacraft/.

Also look at the following portal sites, which will direct you to additional resources:

http://www.adaic.com

http://www.adapower.com

http://www.adaworld.com

Hope this helps.

Why Program in Ada?

From: Stephane Richard <stephane.richard@verizon.net>
Date: Thu, 18 Sep 2003 16:46:42 GMT
Subject: Re: Pourquoi programmer en Ada?
Newsgroups: fr.comp.lang.ada

[Translated from French. -- dc]

> I hope that the question in the subject does not shock you. I wonder in fact, regarding this language which I only know a little, what are the convincing reasons to use it, as many manufacturers offer more developed, more sophisticated solutions. In other words, what does Ada bring which would not be in C++, C#, Java or other more advanced languages? What are its advantages, its specificities?

Here is why I program in Ada.

For "Embedded" systems there's nothing better than Ada.

High precision clock (1 millionth of a second); very precise contrary to C and other languages.

Multi-tasking integrated into the language, offering much better control of the execution of one or more tasks with respect to another, including priorities of the tasks, communication between the tasks, etc, etc... No language offers this kind of control.

Very organized and optimal access to low level functions, especially when compared to Pascal or Delphi.

Portability of source code to all the platforms available, which requires relatively fewer code changes than with C or others.

Ada was created with the aim to offer an ideal language for the development of reliable applications (which are not allowed to crash), and it offers all that is necessary [?] to make sure that a program doesn't crash. It detects many errors during compilation that other languages (C, Pascal, Delphi, Modula, and others) cannot detect before the execution of the program [if at all -- dc].

I can continue for a long time on why. Ada is a superior language, whatever the "popular" languages say.

Stéphane Richard, Senior Software and Technology Supervisor

From: Patrice Freydiere <frett27@free.fr>
Date: Thu, 18 Sep 2003 20:12:15 +0200
Subject: Re: Pourquoi programmer en Ada?
Newsgroups: fr.comp.lang.ada

For my part,

I find the syntax of Ada very expressive, it brings a great legibility. It is thus not necessary to be a specialist to be able to reread Ada code, however technical it is.

Strongly typed language.

Ada gives much control over staticity and dynamicity of the data structures. (Ada makes it possible to go very far "statically"..)

Ada makes it possible to develop in a functional or an object-oriented way whenever that helps.

Ada is a very portable language, available on many platforms.

Ada allows to create high level abstractions, and thus, allows re-use of code and very thorough functionality.

The development of portable multi-tasking applications is a real pleasure!!

PS: A good example is to look at what ACT-Europe does. They do not have the labour force of Microsoft... Nevertheless their achievements are impressive: compilers, XML, "Ada-ptation" of graphic interfaces... And the Ada contributors are really impressive technically speaking; -)

From: doucet@laas.fr (Jean-Etienne Doucet)
Date: 19 Sep 2003 08:09:46 GMT
Organization: LAAS/CNRS, Toulouse, France
Subject: Re: Pourquoi programmer en Ada?
Newsgroups: fr.comp.lang.ada

And what to say about GPS (Gnat Programming System)? Once you tried it, it is difficult to leave it aside. (That's my case...)

[Patrice Freydiere responded: -- dc]

Indeed, I use it since some time and I find it very well done!!! And of an astonishing simplicity for development, compilation and debugging!! I gave up Emacs for GPS... (but I continue to use Emacs for other things).

From: Gautier de Montmollin <gdemont@hotmail.com>
Date: Thu, 18 Sep 2003 23:47:44 +0200
Subject: Re: Pourquoi programmer en Ada?
Newsgroups: fr.comp.lang.ada

Clear syntax, modularity, strong typing, professionalism, thorough standardization -> thorough portability, non-proprietary.

The question should not shock, on the contrary! But in fact, who said that (for example) C++ was more advanced?!

From: Jean-Yves Lenhof <jylenhof@pasdespam.fr>
Date: Tue, 23 Sep 2003 22:57:10 +0200
Subject: Re: Pourquoi programmer en Ada?
Newsgroups: fr.comp.lang.ada

Strong typing, it is very often possible not to use pointers, legibility of the code, excellent source level portability, many libraries, genericity, a complete reference manual, Glade to develop distributed programs.

[...] In the literature one often says: once compiled, an Ada program contains fewer bugs than a program developed in C, C++, Java, because the compiler is very strict.

In addition the denomination "Ada compiler" conforms to a standard. Whereas the C compilers implement one or more standards more or less.

On Generics in Java and Contributions from Ada

From: Gabe <depacegp@yahoo.com>
Date: Wed, 8 Oct 2003 13:19:25 -0700
Subject: Question about historical fairness
To: team-ada@acm.org

I am a huge fan of Ada. Unfortunately right now I am working in Java. I heard rumblings that the project I am working on will have to be rewritten to take advantage of generics (and thus have to wait until JDK 1.5 realistically - which is another release of the Java language, sometime in the future). My first thought was - "when will Java catch up with Ada?" I hesitated to say this to anyone else for a few reasons, but the first was because I feel it isn't possible to get an unbiased history of computer languages. I would like to see what contributions were made by whom at what time - and honestly - without hype, spin, marketing, or other lies. Does anyone know if this exists?

[See also "Thomas Bergin et al - History of Programming Languages" earlier in this AUJ issue. -- dc]

Also, it would be interesting to see just what contributions Ada made, because everything I hear now for why Java is good are exactly the same things I heard many years ago that made Ada good, only this time you'd be the fool to think they weren't true.

From: Jack Beidler <beidler@cs.scranton.edu>
Date: Wed, 8 Oct 2003 18:31:27 -0400
Subject: Re: Question about historical fairness
To: team-ada@acm.org

Take a look at the name of some of the people on the Generic Java committee and you will see a very definite Ada influence - e.g. Norm Cohen.

Actually, Generic Java is closer than you think, there is a beta version and its pretty solid. Because of the obvious syntactic relationship between Java and C++, the notion for Java generics is analogous to C++ generics. I've been using Java for a couple of years now after 15 years of playing with Ada. I must confess that my approach to Java has been biased by Ada, and that bias is for the better. I am saddened by the great hype over Java when I see the poor quality of the Java reference books that are written in a very shallow way and make very poor use of Java interfaces (like an Ada specification file) and very few clues as to how to create Java software so that it is extendable.

If you want to see a great example of bad software take a look at Java's API and its LinkedList class, which by definition of its methods MUST be inefficient, yet so many Java books use it.

Dr. John (Jack) Beidler, Professor of Computer Science, Computing Sciences Department, University of Scranton, Scranton, PA

On Java, Ada and C for ESA Projects

From: Stephane Richard <stephane.richard@verizon.net>
Date: Wed, 08 Oct 2003 10:32:03 GMT
Subject: Re: a good book to start with?
Newsgroups: comp.lang.ada

> [...] Is ESA (European Space Agency) still basing its core software developments on Ada? I read somewhere [...] that Java is emerging as the next ESA's standard language?

[...] I'm not one to bring down languages for the good of other languages. But I can't help but wonder how precise they want (or need) their realtime applications if they are considering Java. Sure Java has a form of Task and Task control mechanism, but well at least on my PC time differential benchmarks I've done (for a music application project of mine, so nothing mission critical per se, however music is all about time precision). Ada came out much more stable and regular than Java as far as timelaps calculation and the execution of tasks during the waiting of delays (as in to end a note or play the next note on a track or multiple tracks).

So precision wise if it's what they want, they might want to do some serious benchmarks. Depends on why they need it I suppose.

Me well I simply prefer Ada for more than one reason. Mainly because it still detects more errors at compile time than Java can, which means that by the time your code compiles, there's: 1. a much higher chance it will run, 2. a much smaller debugging time from the coding to the end of debugging phase.

It's designed with software engineering in mind and as such offers many advantage for large scale application development as far as a programming language goes.

Don't get me wrong, C++, Java probably have their specific place or they wouldn't be what they are today (popular) but from what I'm seeing, I don't think all the right reasons went into making them as popular as they are today. A lot of the reasons aren't based on language capacities, but on sheer popularity (a lot of companies go with Java and C++ because other companies went with Java and C++) that makes Java and C++ popular, but does it make them better, as languages? I think not .

That was my most humble opinion

Stéphane Richard, "Ada World" Webmaster, http://www.adaworld.com

From: Colin_Paul_Gloster@acm.org (Colin Paul Gloster)
Date: 14 Oct 2003 15:58:21 GMT
Organization: Dublin City University (DCU)
Subject: Re: a good book to start with?
Newsgroups: comp.lang.ada

Support for multiple interface inheritance may be one of the main reasons why Java may be a serious rival to Ada for future ESA Attitude and Orbit Control Systems ...

http://control.ee.ethz.ch/~pasetti/RealTimeJavaFramework/doc/index.html

[Note that multiple interface inheritance is scheduled for inclusion in Ada 2005: http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-00251.TXT "Abstract Interfaces to provide Multiple Inheritance"; see also "Multiple Inheritance in Ada95" in AUJ 23.1 (Mar 2002), p.32. -- dc]

From: Jacob Sparre Andersen <sparre@crs4.it>
Date: Thu, 09 Oct 2003 20:44:30 +0200
Organization: CRS4, Center for Adv. Studies, Research and Development in Sardinia
Subject: Re: a good book to start with?
Newsgroups: comp.lang.ada

[...] According to my sources (inside a company with ESA contracts), all new contracts come with a C mandate. I have had the pleasure of hearing quite a bit of complaints about not being allowed to use Ada.

[And from a later message: -- dc]

> Is there any logic reason for such a diktat?

I haven't been told of any, but there is at least one argument in favour: it limits how many programming languages ESA's code reviewers have to know. But so would mandating use of Ada or Visual Basic.

Considering that ESA is still paying for the development of Ada libraries for their next generation of satellites, it really seems weird. And as a tax payer in an ESA member country, I am rather worried about the way they spend my tax-money.

Embedded.com on Love-Hate Relationship with C

From: volkert@nivoba.de (Volkert)
Date: 21 Oct 2003 07:39:19 -0700
Subject: My love-hate relationship with C
Newsgroups: comp.lang.ada

Read Mr. Ganssles Embedded Pulse: http://www.embedded.com/showArticle.jhtml?articleID=15306089

[The article is entitled "My love-hate relationship with C" and starts with: "C, the most popular of all embedded languages, is an utter disaster, a bizarre hodgepodge meant to give the programmer far too much control over the computer. C++ isn't much better. The languages are designed to provide infinite flexibility, to let the developer do anything that can be done on the computer." and "But no language should allow stupid mistakes like buffer overruns or undetected array overflows." -- dc]

From: snarflemike@yahoo.com (Mike Silva)
Date: 21 Oct 2003 18:30:28 -0700
Subject: Re: My love-hate relationship with C
Newsgroups: comp.lang.ada

Way to go, Jack. He's expressed his frustration with C before, but never like this.

Some of the responses are just so typical [cf. "Reader Feedback" on the web-site -- dc]. Stop whining. Learn discipline. Don't limit my technical prowess. Cars can hurt people, should we stop driving too? (Talk about a broken analogy!) Not stated, but sensed: wha'samatta, ain't you MAN enough for C? What surprises me, though, are the number of responses that agree with him.

From: Wes Groleau <groleau@freeshell.org>
Date: Tue, 21 Oct 2003 20:09:58 -0500
Subject: Re: My love-hate relationship with C
Newsgroups: comp.lang.ada

> Reader feedback shows 58 times the word "Ada"

I liked this one (edited): "... haven't seen an article on Ada for years! ... Maybe Ada based projects don't require calling in outside help to save the day." - James Munn

From: volkert@nivoba.de (Volkert)
Date: 22 Oct 2003 11:40:58 -0700
Subject: Re: My love-hate relationship with C
Newsgroups: comp.lang.ada

Anyway, when I read some of the responses to the article I feel that there is a strong desire for some language like Ada.

On Ada and Mainstream Technology

From: Carlisle Martin C Dr USAFA/DFCS <Martin.Carlisle@usafa.af.mil>
Date: Mon, 18 Aug 2003 14:48:48 -0600
Subject: FW: What about Ada?
To: team-ada@listserv.acm.org

> One recent article about the "state" of Ada use: http://www.sei.cmu.edu/publications/documents/03.reports/03tn021.html

[A technical note by Jim Smith, entitled "What About Ada? The State of the Technology in 2003". Based on fragmentary and outdated information the author paints a "bleak future for Ada"... -- dc]

FYI. I've already corrected his misstatements about the Air Force Academy and West Point (we still use Ada), and he indicated he would publish a correction.

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

From: Rod Chapman <rod.chapman@praxis-cs.co.uk>
Date: Tue, 19 Aug 2003 16:02:06 +0100
Subject: The "What about Ada?" article...
To: team-ada@listserv.acm.org

I've also responded to the author, pointing out some of the recent Ada-related articles we've written (you know those "minor" journals like IEEE Software, IEEE Transactions, and CrossTalk...), the SPARK Book, the STC Conference (I counted 5 Ada compiler vendors plus Praxis and PolySpace there in 2003...) and a few other matters...

[See many announcements in this and previous AUJ issues. -- dc]

Rod Chapman, SPARK Team, Praxis Critical Systems

From: Alan and Carmel Brain <aebrain@webone.com.au>
Date: Tue, 19 Aug 2003 22:39:55 +1000
Subject: Re: FW: What about Ada?
To: team-ada@listserv.acm.org

At the risk of "preaching to the converted"... I think he's right with his conclusions.

The future belongs to Visual Basic, Microsoft, Java, Worms, Virusses, Patches, buffer-overflows and the consequent weekly and monthly Software Chernobyls.

That way lies guaranteed employment for software engineers doing maintenance - increasingly in India these days.

Unless and until packs of trained attack-lawyers go after the software industry for its criminal negligence in not using appropriate languages and techniques for both safety-critical and non-safety-critical work, we're stuck with a worsening situation. I used to think that the hard empirical evidence of a language's superiority in productivity would help. But while that's necessary, it's not sufficient.

It's not about being the cheapest and best, it's about being the most popular. [...] Suggestions on how to get us out of this hole welcome.

From: Robert C. Leif <rleif@rleif.com>
Date: Tue, 19 Aug 2003 08:06:37 -0700
Subject: Re: FW: What about Ada?
To: team-ada@listserv.acm.org

From a simple business marketing point of view, Jim Smith is correct. Unfortunately, if US DoD continues to use this type of reasoning, it will purchase the cheapest parts made in China or better yet in the third world for equipment including the weapons systems. Unfortunately this article has no significant mention of software engineering. Unfortunately, the SEI in this case has behaved as do many Government contractors and produced propaganda instead of an engineering study. A study would have compared Ada versus other languages in terms: of the cost of development and maintenance, the defect levels in the products, and the reliability of the products.

I like your suggestion about trained attack-lawyers. I believe we should give a free seminar for these lawyers and provide expert witnesses. A few billion dollars in malpractice judgments will greatly increase the popularity of Ada. I might note that there have been two major problems with promoting the use of Ada. 1) The Ada community includes few if any entrepreneurs that are interested in commercial products. 2) The US DoD goofed when it did not tell NSF to do its job and promote software engineering including Ada and then did not tell Congress of the dire consequences that would result if NSF did not do its mandated job.

More about Ada

From: Jean-Pierre Rosen <rosen@adalog.fr>
Date: Mon, 1 Sep 2003 14:41:49 +0200
Organization: Adalog
Subject: More about Ada...
Newsgroups: comp.lang.ada

See http://www.catholic.org/saints/saint.php?saint_id=1106

[Info on the Catholic saint Ada who lived in the 7th century. -- dc]

Ada Cities

From: Rod Haper <rhaper@houston.rr.com>
Date: Sat, 20 Sep 2003 06:21:27 GMT
Subject: Re: Ada the acronym - amusement
Newsgroups: comp.lang.ada

US placename data from the MIT Geograpic Nameserver
http://www.mit.edu:8001/geo

Placename State County/Parish Lat/Long

Ada County Idaho Ada
43:26:00 N 116:44:00 W

Ada Alabama Montgomery
32:06:19 N 086:16:35 W

Ada Arkansas Conway
35:06:25 N 092:52:24 W

Ada Kansas Ottawa
39:09:05 N 097:53:20 W

Ada Louisiana Bienville
32:32:45 N 093:08:25 W

Ada Michigan Kent
42:57:15 N 085:29:20 W

Ada Minnesota Carlton/Norman
47:17:59 N 096:30:54 W

Ada Ohio Hardin
40:46:10 N 083:49:22 W

From: Randy Brukardt <randy@rrsoftware.com>
Date: Mon, 22 Sep 2003 16:15:51 -0500
Subject: Re: Ada the acronym - amusement
Newsgroups: comp.lang.ada

There is an Ada in Wisconsin, too, but it is an unincorporated community between Howards Grove and Kiel. (That's near my hometown, which is why I know this - my 6th grade teacher lived there.) You can find it on a map (at least on the Rand McNally Road Atlas) by following hwy 42 northwest out of Sheboygan.

On Tricks and Techniques, Bugs and Features

From: Frank J. Lhota <lhota.adarose@verizon.net>
Date: Wed, 30 Jul 2003 18:47:13 GMT
Subject: Re: Non-philosophical definition of Eiffel?
Newsgroups: comp.lang.ada

> As you said it is a trick. Perhaps I should have said "technique" instead.

Can you clarify the difference between the terms "trick" and "technique"? Is "technique" necessarily something else other than a dressed-up trick?

This reminds me of a cartoon I saw at the Alsys Waltham office. The top part of this cartoon has an anthropomorphized, fat, ugly insect wearing grubby clothes and a bowler, and smoking a cigar. Beneath this unappealing anthropod is the caption "BUG". The bottom part of the cartoon shows exactly the same insect, but with a few vital differences. In place of the bowler, he is wearing a top hat. In place of the cigar, he now has a cigarette holder. In place of the grubby clothes, the anthropod is wearing a tuxedo. The caption under this second insect drawing reads "FEATURE".

Conference Calendar

This is a list of European and large, worldwide events that may be of interest to the Ada community. Further 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.

2004

March 18 — 2004 **Ada-Belgium General Assembly** + Technical Presentation: "An Invitation to Ada 2005", ULB, Brussels, Belgium. A technical presentation by Pascal Leroy (IBM France), chairman of the ISO Ada Rapporteur Group

March 22-26 — 3rd **International Conference on Aspect-Oriented Software Development** (AOSD'2004), Lancaster, UK

March 22-26 — **International Conference on Practical Software Quality Techniques & Testing Techniques** (PSQT/PSTT'2004 East), Washington DC, USA

March 24-26 — 8th **European Conference on Software Maintenance and Reengineering** (CSMR'2004), Tampere, Finland. Topics include: Experience reports (successes and failures); Tools and enabling technologies for evolution, maintenance and reengineering tools; Migration, wrapping and interfacing legacy systems; Dealing with legacy systems towards new technologies; etc.

March 25-26 — 8th **IEEE International Symposium on High Assurance Systems Engineering** (HASE'2004), Tampa, Florida, USA. Topics include: Formal Methods; Safety analysis, reliability evaluation and enhancement techniques; Fault-tolerant software design; Evolutionary design of complex systems; Software engineering for embedded systems; etc.

March 27-April 04 — **European Joint Conferences on Theory and Practice of Software** (ETAPS'2004), Barcelona, Spain. Event includes: conferences from 29 March to 2 April, 2004, affiliated workshops on 27-28 March and 3-4 April, 2004. Includes a.o the following events:

- March 27-28 — **Workshop on Foundations of Unanticipated Software Evolution** (FUSE'2004). Topics include: Formal approaches, language concepts and implementation techniques for USE; USE support in programming languages, component models and related infrastructures; Consistency, safety, integrity, constraint enforcement and dependency management issues; etc.
- April 03 — 3rd **International Workshop on Compiler Optimization Meets Compiler Verification** (COCV'2004). Topics include: optimizing and verifying compilation, translation validation, certifying and credible compilation, programming language design and programming language semantics, etc.
- April 03 — **Workshop on Software Composition** (SC'2004)
- April 03 — 4th **Workshop on Language Descriptions, Tools and Applications** (LDTA'2004)

April 04-07 — 4th **International Conference on Integrated Formal Methods 2004** (IFM'2004), Canterbury, Kent, England.

April 14-16 — 9th **IEEE International Conference on Engineering of Complex Computer Systems** (ICECCS'2004), Florence, Italy. Topics include: Tools, environments, and languages for complex systems; Formal methods and approaches to manage and control complex systems; Integration of heterogeneous technologies; Human factors and collaborative aspects; Interoperability and standardization; Systems and software safety and security; Industrial automation, embedded and/or real time systems; etc.

April 19-21 — 11th Annual **European Concurrent Engineering Conference** (ECEC'2004), Hasselt, Belgium. Topics include: Supporting Technologies; Formal Methods and Techniques; Engineering of

embedded systems (HW/SW co-design, specification languages, ...); Collaborative CE Environments for Virtual Teams (CORBA based environments, CE languages and tools, Distributed computing environments, ...); Practical Applications and Experiences (Practical solutions, Systematic guide-lines, Pitfalls and success stories, Case studies, pilot projects and experiments, ...); etc.

April 19-22 **Systems and Software Technology Conference** (SSTC'2004), Salt Lake City, Utah, USA

April 21-23 5th **International Conference on Software Testing** (ICSTEST'2004), Duesseldorf, Germany

☺ April 26-30 **International Parallel and Distributed Processing Symposium** (IPDPS'2004), Santa Fe, New Mexico, USA. Topics include: Applications of parallel and distributed computing; Parallel and distributed software, including parallel programming languages and compilers, operating systems, runtime, middleware, libraries, programming environments and tools for parallel and distributed computing; etc. Includes a.o the following events:

- ☺ April 26 9th **International Workshop on High-Level Parallel Programming Models and Supportive Environments** (HIPS'2004). Topics include: Concepts and languages for parallel and Grid computing (Language interopability, Concurrent object-oriented programming, ...), Supportive techniques for component environments and testbeds (Runtime systems, Compiler techniques,...), etc.
- April 26 2nd **International Workshop on Parallel and Distributed Systems: Testing and Debugging** (PADTAD'2004)

April 27-29 4rd **International SPICE Conference on Software Process Improvement and Capability dEtermination** (SPICE'2004), Lisbon, Portugal

April 28-30 5th **Recent Object-Oriented Trends Conference** (ROOTS'2004), Bergen, Norway. Deadline for early registration: March 22, 2004

May 02-06 **Conference on Design, Analysis, and Simulation of Distributed Systems** (DASD'2004), Washington DC, USA. Topics include: Application oriented methods and tools; Aspects of real-time systems; Case studies, best practices and lessons learned; Fault tolerance / reliability; Modeling of distributed systems including analysis and simulation; New formal concepts and methods for validation and testing; Security and safety; Support for HW/SW codesign; etc.

☺ May 12-14 7th **IEEE International Symposium on Object-oriented Real-time distributed Computing** (ISORC'2004), Vienna, Austria

May 14-17 **International Conference on Computational Science and its Applications** (ICCSA'2004), Assisi, Italy. Topics include: Parallel and Distributed Computing; Reliability Engineering; Software Engineering; etc.

☺ May 23-28 26th **International Conference on Software Engineering** (ICSE'2004), Edinburgh, Scotland, UK. Includes a.o the following events:

- May 24 **International Workshop on Software Engineering for High Performance Computing System Applications** (HPCS'2004)
- May 24 2nd **Workshop on Software Quality** (WoSQ'2004). Topics include: Software Product Evaluation and Certification; Software Quality Education; Methods and Tools for Quality Assurance; Quality Metrics - in-process quality and customer views of quality; Software Quality for Object-Oriented development; Building quality into software products; Combining Quality and Rapid Development; etc.
- May 24-25 7th **Workshop on Component-Based Software Engineering** (CBSE7)
- ☺ May 25 4th **Workshop on Open Source Software Engineering** (OSSE'2004)
- ☺ May 25 **Workshop on Software Engineering for Automotive Systems** (ASE'2004). Topics include: High-level Languages, Specification, SW Architectures and Communication Infrastructures, etc.
- ☺ May 25 **Twin Workshops on Architecting Dependable Systems** (WADS'2004)

May 24-25 — 8th **International Conference on Evaluation and Assessment in Software Engineering** (EASE'2004), Edinburgh, UK. Co-located with ICSE'2004. Deadline for submissions: April 12, 2004 (posters)

May 24-25 — 3rd **International Workshop on Distributed Event-Based Systems** (DEBS'2004), Edinburgh, UK. Co-located with ICSE'2004

May 24-28 — 11th **IEEE Symposium and Workshops on the Engineering of Computer Based Systems** (ECBS'2004), Brno, Czech Republic

☺ May 25-28 — 10th **IEEE Real-Time and Embedded Technology and Applications Symposium** (RTAS'2004), Toronto, Canada. Topics include: QoS issues in systems integration, software engineering, programming languages, system development tools, etc. Special focus is on embedded and real-time applications.

May 26-28 — 10th **International Workshop on Future Trends of Distributed Computing Systems** (FTDCS'2004), Suzhou, China. Topics include: Real-time, Pervasive, and Embedded Systems; Security and Trust in Distributed Systems; Highly Available Distributed Systems; Novel Distributed Applications; etc.

June 06-10 — 5th **International Conference on eXtreme Programming and Agile Processes in Software Engineering** (XP'2004), Garmisch-Partenkirchen, Germany

June 07-09 — 4rd **International Conference on Computational Science** (ICCS'2004), Krakow, Poland. Includes a.o the following event:

- ☺ June 07-09 — **Workshop on Practical Aspects of High-level Parallel Programming** (PAPP'2004). Topics include: high-level parallel language design, implementation and optimisation; applications in all fields of high-performance computing (using high-level tools); benchmarks, experiments using such languages and tools; etc.

June 11-13 — **ACM SIGPLAN** 2004 **Conference on Programming Language Design and Implementation** (PLDI'2004), Washington, DC, USA. Includes a.o the following event:

- ☺ June 11-13 — **ACM SIGPLAN Conference on Languages, Compilers, and Tools for Embedded Systems** (LCTES'2004). Topics include: Programming languages for embedded applications; Real-time UML; Object-oriented modeling and design; Validation and verification techniques for embedded software; Real-time scheduling analysis; Exception and interrupt handling for real-time; Software design for multiprocessor embedded systems; Concurrent+distributed embedded environments/runtime systems; Support for partitioning; etc.

June 12-15 — 4th **IEEE/IFIP Working Conference on Software Architecture** (WICSA'2004), Oslo, Norway. Co-located with ECOOP'2004

♦ June 14-18 — 9th **International Conference on Reliable Software Technologies** - Ada-Europe'2004, Palma de Mallorca, Spain. Sponsored by Ada-Europe, in cooperation with ACM SIGAda (approval pending) and Ada-Spain.

☺ June 14-18 — 18th **European Conference on Object-Oriented Programming** (ECOOP'2004), Oslo, Norway. Topics include: Analysis and design methods; Concurrent, real-time, and parallel systems; Design patterns; Distributed and mobile systems; Language design and implementation; Programming environments; Versioning, compatibility, software evolution; etc. Deadline for submissions: April 1, 2004 (demos, posters, exhibits, practictioners reports). Includes a.o the following event:

- June 14 — **Workshop on Communication Abstractions for Distributed Systems** (CADS'2004). Topics include: Embodiments of communication abstractions, such as middleware services, communications-centric programming languages, communication frameworks, and communication components, such as run-time system and protocol evolution; etc. Deadline for position paper submissions: April 5, 2004

☺ June 14 ECOOP2004 - **Workshop on Practical Problems of Programming in the Large** (PPPL'2004). Topics include: Structuring systems with large amounts of classes/objects/components; Software Architecture; Software Composition; Refactoring, Software Evolution and Migration; Enterprise Application Integration; etc. Deadline for position paper submissions: April 5, 2004

☺ June 15 ECOOP2004 - **Workshop on Programming Languages and Operating Systems** (PLOS'2004). Topics include: type-safe languages for OS; domain-specific languages for OS development; language support for OS verification, testing, and debugging; etc. Deadline for position paper submissions: April 5, 2004

June 14-18 2nd **International Conference on Software Process Improvement** (ICSPI'2004), Washington, DC, USA. Deadline for submissions: March 20, 2004 (papers, presentations)

June 21-22 **Working Conference on Model Driven Architecture: Foundations and Applications** (MDAFA'2004), Linköping, Sweden. Includes special session on UML Profiles for Embedded and Real-Time Systems. Deadline for submissions: April 17, 2004

June 21-24 2004 **International Multiconference in Computer Science and Computer Engineering**, Las Vegas, Nevada, USA. Includes conferences on: Parallel and Distributed Processing Techniques and Applications (PDPTA), Software Engineering Research and Practice (SERP), Embedded Systems and Applications (ESA), etc. Includes a.o the following event:

June 21-24 **International Conference on Software Engineering Research and Practice** (SERP'2004). Topics include: Programming, Software Maintenance, Distributed Systems, Software Quality, etc.

June 27-July 02 29th Annual **USENIX Technical Conference** (USENIX'2004), Boston, Massachusetts. Topics include: Distributed and parallel systems, Embedded systems, Reliability and availability, etc., plus FreeNix Track on free and open source software.

☺ June 28-July 01 **DAta Systems In Aerospace** (DASIA'2004), Nice, France. DASIA'2004 was moved from May 24-27 in Istanbul, Turkey, to May 28 to July 1 in Nice, France.

June 28-July 01 **International Conference on Dependable Systems and Networks** (DSN'2004), Florence, Italy. Deadline for submissions: April 26, 2004 (student forum, fast abstracts). Includes a.o the following event:

☺ June 29 **Twin Workshops on Architecting Dependable Systems** (DSN 2004 WADS)

☺ June 30-July 02 16th **Euromicro Conference on Real-Time Systems** (ECRTS'04), Catania, Italy. Topics include: embedded real-time systems; real-time control applications; frameworks and tools for development and analysis; software architectures and languages; design, scheduling, timing and execution-time analysis; validation; etc.

July 05-09 8th **International Conference on Software Reuse** (ICSR-8), Madrid, Spain. Theme: "Software Variability Management for Reusable Software". Topics include: Software generators and domain-specific languages; Quality aspects of reuse, e.g. security and reliability; Success and failure stories of reuse approaches from industrial context; etc.

☺ July 07-09 10th **International Conference on Parallel and Distributed Systems** (ICPADS'2004), Newport Beach, California. Topics include: Parallel and Distributed Systems, Parallel and Distributed Applications and Algorithms, Distributed Operating Systems, Security and Privacy, Dependable Computing and Systems, Real-Time Systems, etc.

July 12-13 **Foundations of Computer Security** (FCS'2004), Turku, Finland Affiliated with LICS'2004 and ICALP'2004. Topics include: Formal specification, Language-based security, Static analysis, etc. Deadline for submissions: April 2, 2004

☺ July 12-15 **OMG** Annual **Workshop on Real-Time and Embedded Distributed Object Computing**, Washington, DC, USA. Topics include: Applying CORBA in any real-time or embedded system; High-confidence, high-availability or safety-critical CORBA applications; Security considerations in real-time or embedded CORBA deployments; Real-Time & Embedded Specifications and

	Standards; Real-Time & Embedded Product Issues; Real-Time and Embedded Advanced R&D Topics, such as advanced scheduling techniques and high-level real-time programming models; etc. Deadline for abstract submissions: March 19, 2004
July 12-16	10th **International Conference on Algebraic Methodology And Software Technology** (AMAST'2004), Stirling, Scotland, UK
☺ July 25-28	23rd Annual **ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing** (PODC'2004), St. John's, Newfoundland, Canada. Topics include: all areas of distributed systems; any aspect of distributed computing, including systems, design, verification, implementation, application, ...; implementation, analysis, evaluation, and deployment of real systems; intersection of security and distributed computing; etc. Deadline for submissions: May 1, 2004 (nominations for Edsger W. Dijkstra Prize in Distributed Computing)
August 19-20	3rd **International ACM-IEEE Symposium on Empirical Software Engineering** (ISESE'2004), Redondo Beach, CA, USA. Topics include: strengths and weaknesses of software engineering technologies; empirical studies of software processes and products; evaluation and comparison of techniques and models (cost estimation, analysis and design methods, testing); reports on benefits derived from using certain technologies; experience management; etc. Deadline for submissions: May 15, 2004 (posters)
August 19-20	9th **Australian Workshop on Safety Related Programmable Systems**, Brisbane, Australia. Theme: "Transport - Can we trust programmable technology?" Deadline for submissions: April 16, 2004 (papers)
August 26-28	11th **International Static Analysis Symposium** (SAS'2004), Verona, Italy. Co-located with LOPSTR'04, PEPM'04, and PPDP'04. Deadline for submissions: April 4, 2004
August 27	WCC04 - **Workshop on Architecture Description Languages** (WADL'2004), Toulouse, France. Part of the IFIP World Computer Congress 2004. Topics include: Components, Connectors, Composition; Semantics, Formalization; Verification, Simulation, Test; Tools and Development Environments; Standardization; Industrial Projects.
☺ August 31-September 03	10th **International Conference on Parallel and Distributed Computing** (Euro-Par'2004), Pisa, Italy. Topics include: Support tools and environments; Compilers for high performance; Distributed systems and algorithms; Parallel programming models, methods and languages; etc.
August 31-September 03	30th **EUROMICRO Conference** (EUROMICRO'2004), Rennes, France
September 06-10	12th IEEE **International Requirements Engineering Conference** (RE'2004), Kyoto, Japan. Deadline for submissions: April 16, 2004 (doctoral symposium), April 23, 2004 (research demonstrations)
September 08-10	4th **International Conference on Quality Software** (QSIC'2004), Braunschweig, Germany. Topics include: economics of software quality, review, inspection and walkthrough, reliability, safety and security, quality tools, formal methods, static and dynamic analysis, validation and verification, distributed systems, embedded systems, enterprise applications, etc. Deadline for submissions: March 19, 2004 (papers)
September 11-17	20th IEEE **International Conference on Software Maintenance** (ICSM'2004), Chicago, IL, USA.
☺ September 15-17	17th **International Conference on Parallel and Distributed Computing Systems** (PDCS'2004), San Francisco, California, USA. Topics include: Reliable Distributed Computing; Languages, Compilers, and Operating Systems; Libraries and Programming Environments; Software Development, Services, Support, Tools; Middleware for Parallel and Distributed Computing; Embedded Systems; Parallel and Distributed Applications; etc. Deadline for submissions: March 31, 2004 (session proposals), April 9, 2004 (papers), May 1, 2004 (tutorials)
September 20-22	5th **Argentine Symposium in Software Engineering** (ASSE'2004), Córdoba, Argentina. Topics include: Software Quality; Object Oriented Technology and Theory; Design Patterns; Reuse; Software Understanding; Maintenance and Reverse Engineering; Reliability, Safety and Security; Formal methods; Tools and Development Environments; Education in software engineering;

Software Engineering Techniques for Challenging Application Areas such as Distributed Systems, Real-Time Systems, etc. Deadline for paper submissions: April 1, 2004

September 20-24 — 8th **International IEEE Enterprise Distributed Object Computing Conference** (EDOC'2004), Monterey, California, USA. Topics include: Use and enhancement of middleware platforms; Practical experiences with enterprise distributed object computing; etc. Deadline for submissions: March 19, 2004 (papers), April 19, 2004 (workshop submissions)

September 20-25 — 19th IEEE **International Conference Automated Software Engineering** (ASE'2004), Linz, Austria. Deadline for submissions: April 9, 2004 (papers), May 7, 2004 (doctoral symposium)

☺ September 21-24 — 23rd **International Conference on Computer Safety, Reliability and Security** (Safecomp'2004), Potsdam, Germany. Topics include: Safety Foundations (Fault Tolerance; Distributed and Real-time Systems; Maintenance; Reliability; Formal Methods; Risk Analysis; Open Source Software and Safety; Standards, Guidelines and Certification; Commercial-Off-The-Shelf; Verification, Validation and Testing; ...); Safety Applications (Aerospace and Avionics; Automotive; Medical Systems; Power Plants; Railways; Robotics; Chemical Industry; Process Industry; Programmable Electronic Systems; Accident Reports and Management); Security in Safety-Critical Systems; etc. Deadline for submissions: March 28, 2004 (tutorials)

☺ September 22-24 — GI-Jahrestagung Informatik 2004 - **Workshop "Automotive Software Engineering & Concepts"**, Ulm, Germany. Deadline for submissions: April 31, 2004 (workshop submissions)

September 22-24 — 8th **Conference on Formal Techniques in Real-Time and Fault Tolerant Systems** (FTRTFT'2004), Grenoble, France. Deadline for submissions: April 15, 2004

September 27-30 — 24th **IFIP WG 6.1 International Conference on Formal Techniques for Networked and Distributed Systems** (FORTE'2004), Madrid, Spain

September 27-30 — 28th IEEE Annual **International Computer Software and Applications Conference** (COMPSAC'2004), Hong Kong, China. Theme: "Developing Trustworthy Software Systems". Topics include: Software safety; Trustworthy software; Software fault tolerance; High performance software; Component-based software development; Design patterns; Software certification; Software standards; Software engineering education; Distributed systems; Embedded systems; Enterprise systems; High dependable systems; etc.

☺ October 04-08 — 18th **International Symposium on DIStributed Computing** (DISC'2004), Amsterdam, Netherlands. Topics include: distributed programing languages; distributed applications; specification, semantics, and verification of distributed systems; fault-tolerance of distributed systems; cryptographic and security protocols for distributed systems; etc. Deadline for submissions: May 15, 2004

♦ October 07-08 — **Ada-Deutschland Tagung** 2004, Stuttgart, Germany. Co-located with the Automotive - Safety and Security 2004 Workshop, October 6-7, 2004. Topics include (in German): Methoden und Werkzeuge für zuverlässige Softwaresysteme; Beherrschung der Komplexität in SW-Projekten; UML Profile für zuverlässige Software; Vorgehensmodelle und Lifecycle Management von Systemen; Echtzeitsysteme mit Ada; Sichere Software mit Ada; Ravenscar und weitere Sprachprofile; Erfahrungsberichte über Produktivität, Performance und Kosten in Ada-Projekten; Interoperabilität von Ada und anderen Programmiersprachen; Ada in der Ausbildung; etc. Deadline for submissions: May 15, 2004

October 11-15 — 7th **International Conference on the Unified Modeling Language** (UML'2004), Lisbon, Portugal. Deadline for submissions: March 21, 2004 (abstracts), March 31, 2004 (submissions), July 19, 2004 (workshop papers)

October 18-22 — **ACM/IFIP/USENIX International Middleware Conference** (Middleware'2004), Toronto, Ontario, Canada. Topics include: Distributed real-time and embedded middleware platforms; Reliable and fault-tolerant middleware platforms; Applications of middleware technologies, including telematics, command and control, avionics, and e-commerce; Novel paradigms, APIs, and languages for distributed systems; Impact of emerging Internet technologies and standards on

middleware platforms; etc. Deadline for submissions: March 30, 2004 (abstracts), April 6, 2004 (research & work in progress papers), TBA (posters), March 30, 2004 (workshops), May 11, 2004 (tutorials)

☺ October 24-28 — 19th Annual **ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications** (OOPSLA'2004), Vancouver, Canada. Topics include: object technology and its offshoots. Deadline for submissions: March 19, 2004 (technical papers, Onward! submissions, practitioner reports, tutorials, panel proposals, workshop proposals, DesignFest(R) proposals, and educators' symposium), July 2, 2004 (posters, demonstration proposals, and Doctorial Symposium and Student Volunteers submissions)

October 24-28 — 3rd **International Conference on Generative Programming and Component Engineering** (GPCE'2004), Vancouver, Canada. Topics include: Generative techniques for Product lines and architectures, Embedded systems, etc.; Component-based software engineering (Reuse, distributed platforms, distributed systems, evolution, analysis and design patterns, development methods, formal methods); Integration of generative and component-based approaches; Industrial applications; etc. Deadline for submissions: March 19, 2004 (papers, workshops), April 30, 2004 (practitioners, tutorials), July 2, 2004 (demonstrations)

October 31-November 06 — ACM SIGSOFT 2004 12th **International Symposium on the Foundations of Software Engineering** (FSE-12), Newport Beach, California, USA. Topics include: Component-Based Software Engineering; Empirical Studies of Software Tools and Methods; Generic Programming and Software Reuse; Software Engineering and Security; Software Engineering Tools and Environments; Software Metrics; Software Reliability Engineering; Software Safety; Specification and Verification; etc. Deadline for submissions: March 24, 2004 (abstracts), March 29, 2004 (papers), April 15, 2004 (workshops)

November 03-05 — 15th IEEE **International Symposium on Software Reliability Engineering** (ISSRE'2004), Saint-Malo, Bretagne, France. Theme: "Achieving Software Dependability through Model-Driven Engineering". Deadline for submissions: April 2, 2004 (abstracts), April 18, 2004 (regular papers), May 1, 2004 (tutorials), July 1, 2004 (industry practice), July 10, 2004 (student papers, fast abstracts)

♦ November 14-18 — 2004 **ACM SIGAda Annual International Conference** (SIGAda'2004), Atlanta, Georgia, USA. Topics include: safety and high integrity issues, real-time and embedded applications, Ada & software engineering education, Ada in other environments such as XML and .NET, Ada and other languages, metrics, standards, analysis, testing, validation, and quality assurance, etc. Deadline for submissions: May 2, 2004 (technical articles, extended abstracts, experience reports, workshops, panels, and tutorials)

November 22-26 — John Robinson & Associates - **Public Ada Programming Course**, Cheltenham, UK. A practical course with two streams covering Ada 83 and Ada 95.

☺ December 05-08 — 25th **IEEE Real-Time Systems Symposium** (RTSS'2004), Lisbon, Portugal. Topics include: QoS support; Real-time systems middleware; Security and survivability; Real-time and dependability; Compiler support; Embedded operating systems; Software engineering; RT programming languages; Scheduling; Formal methods; Case-studies; etc. Deadline for submissions: May 1, 2004

December 06-08 — 6th **Symposium on Operating Systems Design and Implementation** (OSDI'2004), San Francisco, California. Topics include: distributed systems, embedded systems, etc. Deadline for paper submissions: May 14, 2004

December 10 — Birthday of Lady Ada Lovelace, born in 1815. Happy Programmers' Day!

2005

April 02-10 — European Joint Conferences on Theory and Practice of Software (ETAPS'2005), Edinburgh, Scotland, United Kingdom. Event includes: conferences from 4-8 April, 2005, satellite events on 2-3 and 9-10 April, 2005

12th International Real-Time Ada Workshop

Brian Dobbing[1], John Barnes[2], Miguel Pinho[3]

[1] *Praxis Critical Systems Ltd, 20 Manvers Street, Bath BA1 1PX email: brian.dobbing@praxis-cs.co.uk*

[2] *John Barnes Informatics, 11 Albert Road, Caversham, Reading RG4 7AN email: jgpb@jbinfo.demon.co.uk*

[3] *ISEP, Polytechnic Institute of Porto, Rua Dr. Ant. Ber. Almeida 431, 4200-072 Porto email: lpinho@dei.isep.ipp.pt*

Abstract

The 12th International Real-Time Ada Workshop was held in northern Portugal. The main focus was on developing proposals that relate to real-time or high-integrity systems for the Amendment to Ada that is scheduled for 2005. The workshop was very successful both in refining existing proposals, and in identifying important new ones. The delegates also had a thoroughly enjoyable time, and are very grateful to the organizers Miguel Pinho and Tullio Vardanega for all their efforts

Keywords: Real-Time Ada.

1 Introduction

The 12th International Real-Time Ada Workshop[1] was held in the beautiful setting of Viana do Castelo in northern Portugal between 15th and 19th September 2003. The location was the Pousada Monte de Santa Luzia (Figure 1) situated near the top of a hill above the town, with breathtaking views over dense trees, past the medieval church, and across to the Lima River and the Atlantic ocean (Figure 2).

Figure 1. Pousada Monte de Santa Luzia.

A "Pousada" is a hotel of a unique Portuguese chain, but the name could also be read as pressing on with Ada (from "pousser" – to press in French). This was particularly appropriate, since the greater part of the workshop was devoted to Ada Issues (AIs) for the upcoming revision of the language that relate to real-time or high integrity systems.

[1] The workshop agenda, as well as some photos (including the ones presented in this report) and videos are available at http://www.hurray.isep.ipp.pt/irtaw2003

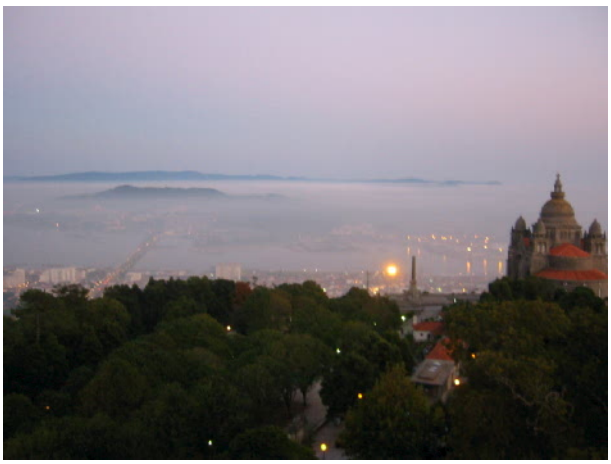

Figure 2. Overlooking the Lima River and the City.

2 Fun Times

The first big issue to be addressed on arrival at Porto airport was one of spatial management. There was no great difficulty in getting six into Miguel Pinho's people carrier, but the luggage space was woefully inadequate, requiring the application of elaborate compacting garbage collection algorithms before the rear door could be closed. This allocation of space however did not accommodate dynamic environmental change, such as the effect of sharp left-hand bends, which resulted in an undesirable reconfiguration and a breach of the firewall between the luggage area and one unfortunate delegate seated in the rear.

However, one of us who arrived at a more sensible time had the luxury of a personal limousine all the way marred only by some sticky roadworks.

Having arrived, we were greeted with a luxurious hotel in lovely grounds that included both an outdoor swimming pool and a tennis court. With temperatures close to 30°C, the pool and its surroundings proved to be a favourite watering hole for many of the delegates. Some even managed the exertions of tennis, although one aging delegate was somewhat the worse for wear after playing only six games. The hotel contained a small snooker table – approximately pool table sized but with the tighter snooker-

style pockets – but provided pool balls. Alan Burns once again demonstrated his prowess for this game, obliterating all and sundry. It was particularly galling not only to be totally outplayed, but then having to pick up the tab in full for the hire of the table.

For those of a more cultural inclination there were very extensive iron-age/Roman ruins within a short stroll of the hotel as well as the Templo-Monumento de Santa Luzia from the roof of which the views were especially stunning. And for the romantics, there were gorgeous sunsets (Figure 3). The new sport of step was also accounted for in the Santa Luzia surroundings, with more than 600 steps all the way down to the city. Going down was easy. Difficult was to come back, particularly if you stop occasionally to have a quick beer.

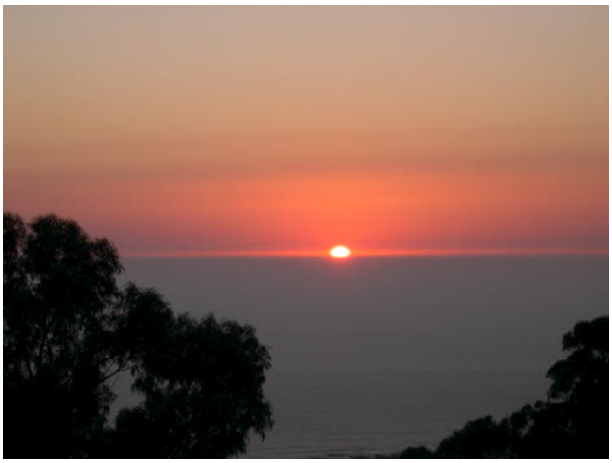

Figure 3. Sunset over the sea.

Tullio and Miguel did a wonderful job as hosts in keeping everything running smoothly. Clearly their job was facilitated and the organization duties were easily carried, due to the high spirits and openness of the delegates.

The meals were taken at the hotel's restaurant, which had peculiar timetables. Breakfast was only available after 8 a.m. and lunch was only after 1 p.m., which was quite inadequate for those early riser delegates. Miguel tried hard to ensure that the vegetarians were catered for, although the local culinary skills in this area were limited. One such meal in particular closely resembled a cow pat, and did not smell or taste much better either. Nevertheless, especially for the many carnivores, the food was generally excellent and was accompanied by an appropriate quantity of wine.

On the final evening there was a trip to a local restaurant. Before the meal, a quick stop was made at a local museum, where some of the participants were taken through a journey of two local traditions: goldsmith and dresses. Particularly, goldsmith was very interesting; however it seemed that the interest quickly cooled off when prices were inquired. In the meanwhile the rest of the group was getting interested in another local tradition: that of beer drinking.

In the restaurant, after settling in to an initial diet of red wine in traditional blue jugs plus various appetizers we were suddenly invaded by a troupe in traditional dress (Figure 4). They included two piano accordian players, a large and small fiddle and an equilateral triangle. There were also three pairs of dancers.

Figure 4. Regional folk dance.

Towards the end of the evening (after a welcome break) they wantonly enveigled members of the workshop at a vulnerably positioned table to dance with them (Figure 5). In particular this enabled one person (who shall remain anonymous) to give a demonstration of the rare but quite famous old-fashioned Portugese Jug Dance. This is particularly applicable to occasions where there is an odd number of dancers. The odd person is enabled to join in the fun by dancing solo with a jug of wine in each hand. Maybe the phrase "A nice pair of jugs" emanates from this traditional dance which is of course performed best when the jugs are of equal size and have handles arranged enantiomorphically.

Figure 5. Guess who's who?

After the dinner, while some of the participants were taking a stroll back to the bus, others were lost in the city. They were finally found enjoying *una última copa* in a local pub. The evening ended with a fascinating bus tour of the locality brought on by the fact that the driver didn't know the best way back to the hotel, since a new roadwork had popped up (a recurrent event during the travel in Portugal).

3 Serious Stuff

The days were organised into slots of three-and-a-half hours, comprising the morning session, an extended lunch break, and the afternoon/evening session. This is the normal style for IRTAWs – the extended lunch period providing plenty of time for delegates to unwind and enjoy the lovely surroundings, so that they would be fully refreshed for the final work session of the day.

Figure 6. Relaxing in the middle of a productive session?

During the sessions, a small break was always welcomed in order to refuel with some coffee or tea (Figure 6). Eventually, in the course of the workshop, breaks were more used for offline collaborative work, to prepare the rest of the session or other sessions of the workshop (Figure 7).

Figure 7. Breaks were also for collaborative work.

The main topics for the sessions were:

- a summary of the new language features for Ada 2005, as expressed in AIs that are being discussed by the ARG;
- the current state of the RTSJ (real-time specification for Java);
- review and update of the current AIs that relate to real-time and high-integrity (Figure 8 shows Andy Wellings presenting his intended alterations on the execution budgets AI);
- a discussion on what language changes could be proposed in the area of flexible scheduling;
- the status and experiences of the Ravenscar Profile and any changes that might be needed to the Ravenscar Guide;
- a discussion on the production of new AIs that the workshop supported.

In this last technical session, a summary of the work performed within the workshop was made by the session chair (Figure 9), in order to prioritise the new (and some of the changed) AIs for the revision process.

Rapporteur notes for each of these sessions are provided separately.

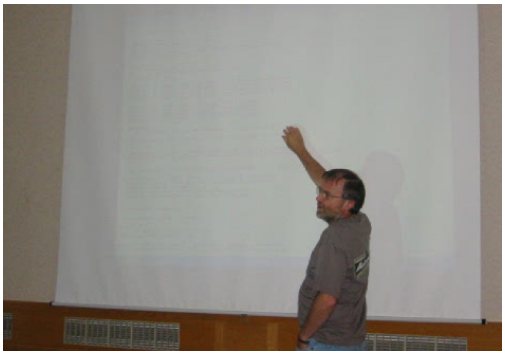

Figure 8. Some code finally: but was it Ada? Yes!

Figure 9. Last technical session: summarizing the work.

4 Future IRTAWs

The closing session of the workshop was to assess the success of the 12th edition of IRTAW, and to decide whether the workshop series should continue and, in case, the time frame and the venue of it.

The Program Chair reported that the workshop this year attracted an acceptable level of interest, regarding both the number and quality submissions and the number of committed participants. All those familiar with the IRTAW series know that the special style of the workshop (intense, discussion-based, working sessions interspersed with long breaks, inter-session preparatory work, informal environment) works somewhat against widening its scope and its selected participation. Obviously however the workshop must stay relevant and significant to fulfil its

intended role, which does require visibility, and, consequently, the injection of fresh blood and/or the inclusion of new attractive discussion topics.

Furthermore, the usual 18-month span occurring between successive IRTAW events causes this edition to be the last one to be able to contribute proposals to the ongoing Ada revision process, due for completion in 2005. Similarly, Q1-2 2005 would also be too early for language technology incorporating significant new features to be available for evaluation.

This notwithstanding, the IRTAW group expressed their wish to continue to follow very closely the progress of the language revision process, and therefore to remain active as such for the whole duration of the process.

The IRTAW group felt they could continue to play an important role towards Ada and real-time programming also beyond the horizon of the current revision process. Worthwhile contributions could be made regarding the quality of implementation of the relevant language changes and their applicability to real-time, distributed and fault-tolerant programming paradigms as well as partitioning and interoperability with other languages.

All aspects considered, the group deemed it wiser to defer any decision on future editions of IRTAW to a later time and yet agreed to:

- maintain the IRTAW group electronically active, so as to continue to monitor the Ada language revision process, and;
- hold an IRTAW Program Committee meeting on the occasion of the Ada Europe 2005 conference, to actually decide on the opportunity, and possibly on the logistics, of a new edition of the workshop.

My First Ada-Europe Conference

Miguel Masmano

Technical University of Valencia, Valencia, Spain; email: mmasmano@disca.upv.es

Abstract

This paper presents the impressions of a young researcher that comes to his first conference for presenting his first published paper. The most positive experience has been to discover that there are a lot of people researching "with" and "for" the Ada language.

Keywords: Ada-Europe, Ada language, real-time systems.

1 Introduction

Ada is a general-purpose programming language that was originally commissioned by the U.S. Department of Defence (DoD) for finishing with the then existing software crisis. But with the recent abandonment of the support of the DoD to the Ada Language, people began to think that the Ada language would be forgotten. However there are still a lot of people and companies that use it intensively for developing projects, education, and so on.

In fact, there exists a European association, called Ada-Europe, which promotes the use of Ada. According to [1]:

"Ada-Europe is an international organisation set up to promote the use of Ada. It aims to spread the use and the knowledge of Ada and to promote its introduction into academic and research establishments. Above all, Ada-Europe intends to represent European interests in Ada and Ada-related matters".

This great effort for conserving and promoting the Ada language demonstrates that Ada is not dying but is completely alive.

One of the activities organised for such purpose is the annual International Conference on Reliable Software Technologies, where researchers from all over the world present work related with the Ada language. The Ada-Europe Conference is also a good place to find a number of companies related with the Ada world, demonstrating that Ada is not a simple academic language, but a professional language useful for all kinds of projects. Although this is an Ada-oriented conference, the background of the Conference is to build reliable software, therefore it is not monographically oriented to Ada issues, but has a more general scope on software quality.

2 The 8th International Conference on Reliable Software Technologies. Toulouse, France

The 8th International Conference on Reliable Software Technologies was held in Toulouse, France, on the 17th, 18th and 19th of June, 2003. This city is important for its aerospace industry, so the place where the Conference was held, had an important meaning by itself.

Toulouse is a beautiful city with lots of places to visit; it made the Conference still nicer.

2.1 Accommodation and transport to the Conference

Several two- and three-star hotels were available for accommodation; all of them seemed to be nice (they say!). Not all the hotels were near of the Conference place, but the organisation also provided a coach which gave us a lift to the Conference at 8 o'clock all the days.

2.2 Tutorials

Several tutorials were given the day before and the day after the Conference (16th and 20th of June), where different new tools and techniques were dealt with.

Tutorials are a good way to quickly learn about hot topics, and are normally given by quite competent experts. For example, an ORK tutorial was given by the creators of ORK themselves, Juan Antonio de la Puente and Juan Zamorano.

2.3 Parallel Workshop: QoS in CBSE 2003

On Friday 20th of June a workshop about quality of service in Component-Based Software Engineering (CBSE) was held. I'm sorry, but I can't say more about this workshop: I just could not attend!

2.4 Invited speakers

The Conference also had the participation of relevant invited speakers, who gave interesting talks about important issues. The invited speakers, and their corresponding talks, were:

- The first day of the Conference, Pascal Leroy gave a very interesting presentation about the current revision of the Ada standard. He explained the new features, some of which were presented in this or in previous editions of the Ada-Europe Conference.
- The second day, Mira Mezini, presented aspect-oriented programming, a new technique for modularising concerns whose modular structures are not hierarchical but rather crosscutting in nature.

- In the morning of the last day, Jörg Kienzle, gave an overview about software fault tolerance. It was an interesting survey.
- Finally, in the afternoon of the last day, Patrick Farail, presented the design process of the Airbus A380. The presentation finished with a nice film about the building process of the A380.

All invited talks where interesting, but to me, the most interesting one was Pascal Leroy's because the new revision of Ada will be used by all of us.

2.5 About the accepted papers and their presentation

There were 29 papers accepted to be presented at the Conference and also printed in the proceedings. Before acceptance, papers were meticulously reviewed by 5 reviewers. For a paper to be accepted it needed to be original, meaning that the paper must not have been published in other conferences.

The Conference proceedings are printed in the "*Lecture Notes in Computer Science*". In particular, the proceedings of the 8th Ada-Europe International Conference on Reliable Software Technologies have the volume number LNCS 2655.

For the presentations, papers are grouped together according to the topic they deal with, creating a session. Each session has a duration of 90 minutes, where three different papers are presented. Since there are 10 different sessions and only three days for the presentations, there are two different parallel tracks. The sessions were the following:

First day (Tuesday June 17th)

- Ravenscar: An interesting session about this Ada profile for real-time systems. To me, this was one of the most interesting sessions, since my research interest is mainly focused on real-time systems.
- Static analysis: This session presented some interesting papers to improve and analyse Ada programs and even the Ada compiler itself.

In parallel

- Language issues: If you wanted to learn more about Ada and new features for the language, this was your session.
- Vendor session.

Second day (Wednesday June 18th)

- Distributed Information Systems: Another important use of the Ada language is distributed systems. Some new techniques and tools were presented in this session.
- Software components: New software components for the Ada language were presented. For example *Charles*, a new set of standard structures, like lists, queues, stacks, and so on, which has been proposed for being added to the Ada standard.
- Formal specification: This session is special for addicts to UML and the like, which is not my case, at least today!

In parallel

- Metrics: In this session some metrics for measuring and observing the behaviour of Ada applications were presented.
- Vendor sessions.

Third day (Thursday June 19th)

- Real-Time kernel: this session presents advances on Ada to be used in real-time. In this session Jorge Real presented our paper [2], called "Running Ada on Real-Time Linux", winning the best paper and also the best presentation awards.
- Real-Time systems design: Another important branch of real-time systems is how real-time systems must be specified. In this session, several approaches using specification languages for real-time systems were presented.

In parallel

- Testing: When an Ada application has been implemented, next step is to test it. The question is how to probe it in an extensive way, with enough coverage. In this session several new tools were presented.
- Vendor session.

In the vendor sessions, several companies with Ada-related products presented them.

The hall was also used between sessions and after lunch for exhibitions about these new products.

2.6 The gala dinner

The gala dinner of the Conference took place in a typical French farm called "ferme Lauragaise", close to a beautiful lake. It was a quite relaxed and pleasant dinner where you could meet very representative people of the Ada world, and share your ideas with them, not necessarily about Ada!

The menu was wonderful, with some local dishes like "foie gras", "cassoulet", duck "magrets", cheeses and also the fabulous French wines.

The farm was located far from the Conference place, but two buses take us to the banquet place.

2.7 Civic reception and visit to the Bemberg Foundation

The AE Conference is a very important event, so the mayor of Toulouse himself offered a cocktail reception for the Conference attendees in the Toulouse town hall and also spoke about the importance of Computer Science in the future. The reception allowed us to visit the splendid town

hall. Later, a short walk let us see some of the emblematic buildings in Toulouse.

The last day of the Conference we visited the Bemberg Foundation, a collection of paintings and art collected throughout Bemberg's lifetime.

Conclusions

My conclusions about the Conference are:

- The Conference was wonderful, it was well organised, and the visits to the city were very instructive.
- Although my research interest is mainly focused on real-time systems, almost all the presented papers were very interesting to me.
- There exists a number of researchers and also companies supporting and using Ada for all kind of projects.
- The Ada-Europe Association does a big effort to promote Ada, it can be seen in this Conference.

Finally, after this wonderful experience I wish I assisted next year to the 9th International Conference on Reliable Software Technologies – Ada-Europe 2004 which will be held in Palma de Mallorca, Spain, on the 14th to 18th of June.

References

[1] www.ada-europe.org

[2] J. P. Rosen and A. Strohmeier (eds) (2003), *Reliable Software Technologies Ada-Europe 2003*, LNCS 2655, Springer-Verlag.

Foreword

As we have bundled issue 24-4 with issue 25-1 in a single shipment, pages 1 to 3 of the 25-1 part will not be repeated. This page is therefore intentionally left blank and the subsequent pages 2-3 will simply be omitted. The News section will therefore resume at its usual position, on page 4 of the new volume.

News – 25-1

Santiago Urueña (ed.) with support from Dirk Craeynest, former News editor

Technical University of Madrid. Email suruena@datsi.fi.upm.es

Preface

When I started working as Ada User Journal News editor more than six years ago, the News section was of rather moderate size and ambition. The first AUJ issue jointly published by Ada-UK and Ada-Europe (19-1) had a News section of only ten widely spaced pages mainly containing items selected from the AdaIC's mailing list.

Since then a lot has changed. Many additional news sources were systematically monitored, interesting items were selected, cataloged, and summarized from newsgroups, mailing lists, web sites, direct mailings, etc. Moreover, the volume of Ada-related and –relevant news increased steadily over the years: a healthy sign for the Ada community! Consequently, the AUJ News section has grown in importance and size, until typically taking at least 32 dense pages, half of each issue.

After producing the news and conference sections in six Journal volumes, i.e. the 24 issues since early 1998, it is now time for me to pass the torch to new and younger talent. I am very pleased that Santiago Urueña stepped forward, and that he, based on a collection of preselected news items, managed to produce the News section for this 25-1 issue in record time!

I'll continue preparing the Conference Calendar, but from now on Santiago is responsible for the News section. I wish him a lot of success and much energy, and am looking forward to reading the forthcoming issues.

Dirk Craeynest, Ada User Journal News Editor 1998-2003

Contents

Ada-related Organizations

Ada-Europe and Ada UK to Merge

From: www.adauk.org.uk

Latest News - 1st December 2003

Ada-Europe and Ada UK to Merge

We are delighted to announce that we have reached agreement with Ada-Europe to merge the two organisations with effect from April 2004.

All future membership applications and renewals will be dealt with directly by Ada-Europe. All current Ada UK members will be contacted in early 2004 with membership renewal information.

Ada-Europe also offer a range of sponsorship packages, and many Ada UK sponsors are already sponsors of Ada-Europe.

This merger reinforces Ada-Europe's position as a successful and active software engineering community, enhancing the profile of the Ada language in the European software engineering community.

In particular we would draw your attention to the 9th International Conference on Reliable Software Technologies being organised by Ada-Europe in Palma de Mallorca in the week 14-18 June 2004. The Ada-Europe conference is now the principal Ada-related event in Europe and a "must attend" event for all Ada practitioners.

There are plans to bring the Ada-Europe conference into the UK for 2005.

Please visit www.ada-europe.org for further information on this community and future conferences.

Adalog agreed as an ACAL

From: Jean-Pierre Rosen <rosen@adalog.fr>
Date: Fri, 16 Jan 2004 12:23:38 +0100
Organization: Adalog
Subject: Press release: Adalog agreed ACAL
Newsgroups: comp.lang.ada

Adalog is pleased to announce that it has been agreed by the ACAA (Ada Compiler Assesment Authority) as an ACAL (Ada Compiler Assesment Laboratory).

In non ISO-speak, this means that Adalog is now an agreed laboratory for the validation of Ada compilers.

This is a recognition of Adalog's long lasting expertise in Ada, and is good news for the Ada community, as it guarantees the continuation of the validation process, one of Ada's great strengths over other programming languages.

For more information, please contact J-P. Rosen, technical director, at rosen@adalog.fr

Praxis Critical Systems joins Ada Resource Association

From: AdaIC Technical Webmaster <webmaster@adaic.com>
Date: Wed, 28 Jan 2004 16:52:18 -0600
Subject: [AdaIC] Praxis joins the Ada Resource Association
To: <announce@adaic.com>

Burlington, Mass. [January 27, 2004]

The Ada Resource Association (ARA) announced today the addition of a new member, Praxis Critical Systems of Bath, England, which represents a major share of the European Ada high-integrity software market.

"Praxis is part of the Ada programming language's cutting edge with its SPARK toolset," S. Tucker Taft, president of the ARA, said from his SofCheck, Inc., offices. "The ARA has been very interested in having them come aboard to help guide Ada into the new millennium."

Praxis Critical Systems specializes in those markets that must ensure reliability and avoid failure. The company concentrates on aerospace and defense, finance, transport, telecommunications and media, energy and utilities, and pharmaceuticals. It provides a range of services, including requirements engineering, systems engineering, software development, safety

assurance, and information security. In the Ada software engineering community, Praxis is best known for its SPARK-Ada programming language toolsets, which are designed for the engineering of high-integrity software.

"The ARA has been a tremendous resource for us already in promoting and supporting the Ada language," said Praxis' products manager Rod Chapman. "We look forward to a profitable and technically innovative partnership with the other members."

The Ada Resource Association (http://www.adaresource.com) is an international trade group comprising the principal vendors of Ada-related technology. The ARA promotes and publicizes Ada technology usage (http://www.adaic.org), and it sponsors the ongoing development and maintenance of the Ada language standard and supporting infrastructure.

The Operation of the ARG

From: Robert I. Eachus <rieachus@comcast.net>
Date: Tue, 16 Dec 2003 22:19:59 -0500
Subject: Re: SIGada Conference
Newsgroups: comp.lang.ada

> Since the ARG people are *very* smart, and *very* motivated to conserve implementors time, I suspect you are wrong :). Personally, I think Java-style interfaces will let me do things I simply cannot do now.

Thanks for the compliment. But I think everyone should be aware that we don't care whether or not a feature is "necessary." What we really agonize about is how to make programming in Ada easier and make it more likely that Ada programs will satisfy all the "ilities."

The interfaces proposal allows a style of multiple inheritance that is currently not well supported in Ada. Wonderful. But the reason that it is a slam dunk that it will be in there, even if it does result in adding a new reserved word, is that it makes in much easier in many cases for a programmer to document what he is really trying to do.

It is possible to make fancy use of generics and dispatching so that you can have one specification matched by different bodies. But the fact that different members of a class are handled in very different ways is hidden in the structure of the program. Interfaces allow a programmer to say he is doing just that, and doing it intentionally.

For example you can have an indexed list type that calls different sort routines, hash table, radix sort, b-tree, etc. depending on the type of the index. With interfaces the "documentation" of this occurs where the interface is implemented for a specific type of index, rather than where the index types are declared. (And you don't need to manufacture a tagged parent type to make it work.)

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. -- su]

December 10 - SIGAda Awards 2003

From: Currie Colket <colket@mitre.org>
Date: Tue, 4 Nov 2003 12:31:33 -0500
Organization: The MITRE Corporation
Subject: Invitation to Nominate Candidates for SIGAda Awards; Nominations Due 16 November 2003
To: team-ada@acm.org

Dear SIGAda Members,

SIGAda Extended Executive Committee (EEC),

SIGAda Working Groups,

Team-Ada, and

Members of the Ada Community:

On Wednesday, December 10, 2003, the 2003 SIGAda Awards will be presented in a special morning plenary session at the SIGAda 2003 conference in San Diego, California. (See http://www.acm.org/sigada/conf/sigada2003/ if you have somehow missed announcements of this year's annual SIGAda international conference.)

We welcome your nominations of deserving recipients.

The ACM SIGAda Awards recognize individuals and organizations who have made outstanding contributions to the Ada community and to SIGAda. The two categories of awards are:

(1) Outstanding Ada Community Contribution Award

-- For broad, lasting contributions to Ada technology & usage.

(2) ACM SIGAda Distinguished Service Award

-- For exceptional contributions to SIGAda activities & products.

Please consider who should be nominated this year. You may nominate a person for either or both awards, and as many people as you think worthy. One or more awards will be made in both categories.

Please visit http://www.acm.org/sigada/exec/awards/awards.html#Recipients and peruse the names of past winners. This may help you think about the measure of accomplishment that is appropriate. You may be aware of people who have made substantial contributions that have not yet been acknowledged.

[…]

From: Stephen Leake <stephen_leake@acm.org>
Date: 15 Dec 2003 19:29:39 -0500
Subject: Re: SIGada Conference
To: chrismiller677@hotmail.com (Chris Miller)
Newsgroups: comp.lang.ada

> Anyone care to post a summary of what went on at this conference. Anything interesting ???

The most interesting thing was that there are several companies making money using Ada, and they are all growing :).

I've been more inspired to look into using SPARK to write solid code.

The Ada 200Y features sound fun, too; aggregates for limited types, a solution for mutually recursive types, Java-style interfaces, no more silent overloading when overriding was meant.

If you subscribe to Ada Letters, the proceedings should appear in a few months ☺

Ada 2005 Presentation

From: dirk@heli.cs.kuleuven.ac.be (Dirk Craeynest)
Date: 16 Feb 2004 21:11:21 +0100
Organization: Ada-Belgium, c/o Dept. of Computer Science, K.U.Leuven
Subject: Ada 2005 presentation, Thu 18 Mar 2004 20:00, Ada-Belgium
Newsgroups: comp.lang.ada,fr.comp.lang.ada,be.comp.programming,nl.comp.programmeren

Ada-Belgium will hold its 11th annual General Assembly on Thursday, March 18, 2004, 19:00, at the U.L.B., Department of Computer Science, Boulevard du Triomphe / Triomflaan, B-1050 Brussels. The official convocation is distributed separately to members and is also available on the Ada-Belgium web-server.

There will be refreshments and pizza for Ada-Belgium members at 18:15.

Please notify us if you are a current or new member and intend to participate at this informal "pre-meeting".

"An Invitation to Ada 2005"

At 20:00 the General Assembly will be followed by a technical presentation plus Q&A, by Pascal Leroy from IBM France.

Abstract

Starting in 2000, the ISO technical group in charge of maintaining the Ada language has been 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 talk will give a technical overview of the most important improvements that are currently under consideration for inclusion in Ada 2005.

Speaker

Pascal Leroy is a Senior Software Engineer with IBM France, and the chairman of the ISO Ada Rapporteur Group, the expert group in charge of revising and maintaining the Ada standard. He has more than 19 years of experience in Ada and has been involved in language design, in compiler and tool development and in consultancy with very large Ada projects, notably in the area of command and control systems. Pascal holds degrees from Ecole Polytechnique and Ecole Nationale des Télécommunications in Paris, France.

More information

The Ada Issues Database of the Ada Rapporteur Group is available on the Ada Conformity Assessment Authority's web site at http://www.ada-auth.org/.

[...] Looking forward to meeting many of you in Brussels!

Dirk Craeynest, President Ada-Belgium

Dirk.Craeynest@cs.kuleuven.ac.be

November 14-18 SIGAda 2004

From: ricky.sward@ix.netcom.com (Ricky E. Sward)
Date: 7 Feb 2004 11:50:19 -0800
Subject: CFP for SIGAda 2004
Newsgroups: comp.lang.ada

Call for Participation - SIGAda 2004

14-18 November 2004, Atlanta, Georgia, USA

Sponsored by ACM SIGAda

http://www.acm.org/sigada/conf/sigada2004 (Approval pending by ACM)

Constructing highly reliable software is an engineering challenge that can now be met in many domains. The SIGAda 2004 conference focuses on how the application of software engineering methods, tools and languages interrelate and on how features in Ada affect the quality of the resulting software. Papers that analyze Ada with respect to these factors or in comparison to other languages are especially welcome. SIGAda 2004 gathers industry experts, educators, software engineers, and researchers interested in developing, analyzing, and certifying reliable, cost-effective software. Technical or theoretical papers as well as experience reports with a focus on Ada are solicited. […] The full Call for Participation on the SIGAda 2004 web site for submission details: http://www.acm.org/sigada/conf/sigada2004 The deadline for submission is 2 May 2004.

[Also see the full Call for Papers at page 240 of the 24-4 part of this combined AUJ issue. -- su]

Ada and Education

University and Industry

From: Chad R. Meiners <crmeiners@hotmail.com>
Date: Mon, 17 Nov 2003 15:17:08 -0500
Organization: Michigan State University
Subject: Re: Re-Marketing Ada (was "With and use")
Newsgroups: comp.lang.ada

> I think they *know* about it, but they are not much interested in teaching a language that appears to be on the decline if not exactly "dead". They want to be able to prepare students for the kinds of things they'll likely see when they get out in industry and don't want to teach something that might be perceived as of purely academic interest with no practical use.

Well first of all the point of a computer science degree is not to prepare you for industry. If you want to be prep'ed for industrial use, go to a technical college, it will suffice. Now I am not saying that universities do not prepare you for industry, but I am saying that university programs have (and should have) concerns other than satisfying the demands of industry. In short, the point of a university degree is to develop your mind; teaching you a trade is secondary. I will admit that it is easy to find professors that have lost touch with this objective. Furthermore, I will admit that collaboration between universities and industry can have a very positive effect. However, the fact remains that the stated goals of universities are to discover truth and to develop minds. (Can you tell that I have had this argument before ;-)

That being said, I agree that the main reason computer science faculty resist teaching Ada is that they believe it to be a dying language. However, I believe that we have a good sellable argument for the language with the following pedagogical reasons: (This is off the top of my head. Please feel free to add more to the list.)

1. Ada is subsetable. Initial student will start out with a small but workable subset of Ada. As the student grows and develops so can the subset of Ada. When teaching new concepts (such as OO programming, or multi-tasking programs) new features within Ada can be introduced and added to the subset.

2. Ada compilers produce helpful and informative compile errors. Ergo, it is easier to spot and correct misunderstandings about programming and software development and such detection usually happens earlier.

3. Ada is versatile. Ada contains enough features to properly facilitate any type of computer science course.

4. Ada is designed via the method of least surprise. This allows the professors to concentrate on programming issues as opposed to programming language issues.

The problem is to convince the professors that the above benefits overweight the benefits of using old lecture note about a language that they already know.

Note that I don't think that universities should not teach C/C++ or any other languages. They should of course offer them as programming language courses. But also as has been expressed many times before in this newsgroup, learning a second language is not nearly as difficult as learning the first one. Learning Ada as a first language really helps students absorb programming concepts faster. (These are my observations; I wish we could do a study on this since such a conclusion would be wonderful press for Ada ;-)

Educational Virtues of Ada

From: Daniel Feneuille <feneuile@romarin.univ-aix.fr>
Date: Wed, 03 Dec 2003 10:54:13 +0100
Subject: Enseigner Ada
To: Ada France <ada-france@ada-france.org>

[Translated from French] The paper «Teaching Ada» [Enseigner Ada, all in French], which I earlier announced to be in the making, was finally born!

[See AUJ 24-2 (June 2003), p.71 --su]

It consists of 35 pages in HTML format that can be accessed via a URL posted at: http://www.ada-france.org/article103.html

I am grateful to Ada France for their support to this project, and I am indebted to those who have provided contributions: they are all cited at the beginning of the article). Finally, I wish to thank those, without supplying textual contributions, have sent me their encouragements, comments and observations.

The paper is not cast in concrete, yet. You may therefore react to its contents and possibly trigger some final addtions to it.

Ada-related Resources

Ada IRC Channel

From: Genro Kane Gupta <genro@niestu.com>
Date: 26 Nov 2003 19:16:17 GMT
Organization: Uncensored-News.Com $6.95 Uncensored Newsgroups.
Subject: [Announce] #Ada IRC channel on Freenode
Newsgroups: comp.lang.ada

This is the annual reminder of the existence of the #Ada channel on the Freenode IRC network. Now entering its third year, the channel is open to all discussions related to the Ada language and its use. We welcome beginners and pros alike, and do our best to maintain a friendly, productive, and informative atmosphere.

The latest open question seems to be, "Why isn't Ada used in more free and open-source programming projects?" Some reasons have been proposed, but you're welcome to come on over and offer your own opinion. We also spend a lot of time on specific programming questions, so come to get help or to offer it.

Point your IRC client to irc.freenode.net and join the #Ada channel. Come one, come all!

Ada Employment Opportunities

From: mcq95@earthlink.net (Marc A. Criley)
Date: 9 Dec 2003 05:37:09 -0800
Subject: Re: Question
Newsgroups: comp.lang.ada

> Is there a Ada job mailing list?

I don't know of a mailing list, but the Ada IC (www.adaic.org) has a jobs section under the "Help Wanted" tab. And it does appear to be at least somewhat active.

Marc, mc@mckae.com, McKae Technologies

"The Efficient Production of High Quality Software" www.mckae.com

From: Jeffrey Carter <jrcarter@acm.org>
Date: Tue, 09 Dec 2003 18:25:00 GMT
Subject: Re: Question
Newsgroups: comp.lang.ada

> I don't know of a mailing list, but the Ada IC (www.adaic.org) has a jobs section under the "Help Wanted" tab. And it does appear to be at least somewhat active.

There's also the SIGAda/ARA Ada Employment Opportunities Database at http://www.seas.gwu edu/~adajobs/

From: Randy Brukardt <randy@rrsoftware.com>
Date: Tue, 9 Dec 2003 13:56:33 -0600
Subject: Re: Question
Newsgroups: comp.lang.ada

> Is there a Ada job mailing list?

If you have a job posting, send it to joblistings@adaic.com. To read job listings, look at http://www.adaic.org/jobs/jobs.html

Ada-related Tools

Booch Components

From: Simon Wright <simon@pushface.org>
Date: Sun, 23 Nov 2003 19:51:36 GMT
Subject: Booch Components 20031123
To: team-ada@acm.org

This release has been uploaded to http://www.pushface.org/components/bc/

Please note that the mirror at http://www.adapower.net/booch/ is out of date, which is why I've not announced the 20030815 release (included here).

Major features since 20030309:

Added BC.Support.High_Resolution_Time, which supports sub-microsecond interval timing. This version is for GNAT on Linux or Windows on x86 processors.

Added BC.Support.Statistics, which supports collecting running mean, variance, minimum and maximum of sequences of values.

BC.Support.Synchronization no longer exposes the controlledness of Semaphore.

Ada now in Cygwin

From: Stephen Leake <Stephe.Leake@nasa.gov>
Date: 30 Oct 2003 17:03:14 -0500
Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)
Subject: Ada now in cygwin
Newsgroups: comp.lang.ada

A message just posted on the Cygwin mailing list:

> From: "Gerrit P. Haase" cygwin@cygwin.com
>
> Subject: Updated: gcc-3.3.1-3
>
> To: cygwin-announce@cygwin.com
>
> I've made a new version of gcc available for download.
>
> This release includes some changes in the package layout. There are now several packages, one package including the core components and one package for each additional front end.
>
> Available front ends now: Ada, C, C++, Fortran, Java, ObjC, Pascal.

I assume this is the somewhat broken GNAT. But it's a step in the right direction! I'll give it a try on the SAL tests.

GNAT for Mac OS X 10.3

From: James E. Hopper <hopperj@macconnect.com>
Date: Tue, 18 Nov 2003 03:07:55 GMT
Organization: Road Runner High Speed Online http://www.rr.com
Subject: Re: GNAT for Mac OS X 10.3?
Newsgroups: comp.lang.ada

> Is a version of GNAT available yet for Mac OS X 10.3? I took a look at http://adapower.net/macos/compiler.html and it looks like all they have is versions for 10.1.5 and 10.2. Can the 10.2 version be used safely on a G5 running 10.3.1?

[...] we have a mailing list that you might want to join.

We are working on 10.3 specific version of the compiler we hope to have it real soon now but no dates yet. The current 10.2 version works fine on 10.3 but not with xcode. What several people have done is done an update to 10.3 but not to xcode and all runs well. If you already have xcode IDE installed, you should be able to install the compiler but the integration with Apples IDE won't work.

From: Andrew Reynolds <awreynolds@mac.com>
Date: Tue, 18 Nov 2003 03:16:34 GMT
Subject: Re: GNAT for Mac OS X 10.3?
Newsgroups: comp.lang.ada

Yes, the 10.2 compiler can be used with 10.3. The procedure for doing so is a little tricky right now. We are in the process of fixing the one remaining problem with the 10.3 compiler.

First install the developer tools from the CD provided with Panther. Next, do the following:

$ sudo gcc_select 3.1

password: <your password>

Then install the 10.2 version of GNAT from the MacAda.org web site.

If you need more help E-mail me directly.

Andrew W. Reynolds

GNAT for MacOS X Development Team
Compiler/Run-Time System/Bindings/ Sample Code

GNAT Compiler for AVR Targets

From: rolf.ebert_nospam_@gmx.net (Rolf Ebert)
Date: 5 Dec 2003 04:48:52 -0800
Subject: [Announce] AVR-Ada V0.1 released
Newsgroups: comp.lang.ada,comp.arch.embedded

We are proud to announce the first release of AVR-Ada, one of the first GCC based Ada compilers targeting 8-bit microcontrollers.

You can get the project description and some (limited) documentation at avr-ada.sourceforge.net

The SF development pages with the download section are at www.sourceforge.net/projects/avr-ada

AVR-Ada is available in source form only. Binary packages of the cross compiler for Linux and Windows are expected to appear with future releases of cdk4avr (cdk4avr.sourceforge.net) and WinAVR (winavr.sourceforge.net).

Feel free to join the mailing list at https://lists.sourceforge.net/mailman/listinfo/avr-ada-devel.

It has quite low traffic. Please use SF's bug reporting system for guiding future development of AVR-Ada.

The goal of the AVR-Ada project is make the gcc based Ada compiler GNAT available for the AVR microcontrollers.

More specifically the project wants to provide:

- a working compiler based on the existing AVR and Ada support in gcc

- a minimalistic Ada runtime system

- a useful AVR specific support library

Although the compiler and the library have considerably improved in the last few months they still have some problems. Do not base a commercial project on this tool chain. Or if you do, do it at your own risk :-).

Most of Ada's static features can be used with AVR-Ada. A typical Ada run time system is practically non-existent (and will probably never be). As a consequence we have to sacrifice some of the typical useful Ada features like run time checks, exception handling, timing commands (no delay statement), tasking, etc. Some of them are on our todo list for future releases, though.

We provide an Avr package hierarchy with some useful type and interface definitions and most importantly the necessary definitions for most AVR parts.

Some sample programs in the apps/ directory show how to use the compiler and the library. This includes the tutorial program from the avr-libc distribution translated to Ada.

The documentation is very low and consists only of the pages at avr-ada.sourceforge.net. A copy of the pages is in the directory AVR-Ada/web/ for offline reading.

Summary of Ada Graphic Tools

From: Peter Hermann <ica2ph@sinus.csv.ica.uni-stuttgart.de>
Date: Wed, 12 Nov 2003 15:17:08
Organization: Comp.Center (RUS), U of Stuttgart, FRG
Subject: Ada graphic tools' summary
Newsgroups: comp.lang.ada

Is there somewhere an Ada graphic tools' summary with abstracts and/or evaluations? Which tools/libraries should I recommend to somebody who asks for doing graphics with Ada? I would like to beef up my www.csv.ica.uni-stuttgart.de/ [web site, placed at:] homes/ph/resources_on_ada.html with any of your inputs.

From: lemchens@lemmies.lb.bawue.de (arvids lemchens)
Date: 13 Nov 2003 08:42:00 +0100
Organization: A poorly-maintained Debian GNU/Linux InterNetNews site
Subject: Re: Ada graphic tools' summary
Newsgroups: comp.lang.ada

[...]

> Which tools/libraries should I recommend to somebody who asks for doing graphics with Ada?

Assuming you want to open a window under X11 and plot some dots, there are

a. GTKAda at http://libre.act-europe.fr/GtkAda/main.html

b. in combination with a. http://www.ctr.unican.es/win_io/ which makes the hole thing a lot simpler.

c. binding to XLib, x11ada. There are different versions on the web. Try google to find the best/newest/....

d. possibly much more that I don't know of.

From: Chad R. Meiners <crmeiners@hotmail.com>
Date: Thu, 13 Nov 2003 14:28:44 -0500
Organization: Michigan State University
Subject: Re: Ada graphic tools' summary
Newsgroups: comp.lang.ada

[...]

> - real-time dynamic displays as for a video game or visualization of a physical process?

Ada thin binding for SDL http://sourceforge.net/projects/adasdl/

There is a thick binding in the CVS tree, also.

From: James E. Hopper <hopperj@macconnect.com>
Date: Thu, 13 Nov 2003 23:57:19 GMT
Organization: Road Runner High Speed Online http://www.rr.com
Subject: Re: Ada graphic tools' summary
Newsgroups: comp.lang.ada

> - a graphic UI in the sense of Windows or Macintosh, in which case a "graphic library" is a binding to the Windows or Mac OS?

For Mac OS X, there are carbon bindings at http://MacAda.org as well as examples for using them.

> - Pixar type photo-realistic rendering of complex scenes?

Ada Open Gl works great on Mac. http://adaopengl.sourceforge.net/

> - real-time dynamic displays as for a video game or visualization of a physical process?

The SDL that someone already pointed out works on Mac OS x as well.

> - a hardware-independent set of routines to draw straight lines and ellipses?

http://adaopengl.sourceforge.net/

> A further category would be image processing as is done by tools like, e.g. photoshop or the gimp.

I have been working on an Ada binding to the c++ class library at www.ossim.org which does image processing and geographic stuff. For instance I wrote a nice little Mac drag and drop app that translates files like dted, dem, uncompressed nitf, etc to jpeg, raw, and tiff.

Have done a number of other nice apps to do things like cropping to geographic rectangles, resample to different degrees per pixel, etc.

We hope to put these up on the ossim website soon.

AWS 1.4 - Ada Web Server

From: Pascal Obry <p.obry@wanadoo.fr>
Date: 13 Nov 2003 17:54:53 +0100
Organization: Home - http://perso.wanadoo.fr/pascal.obry
Subject: Announce: AWS 1.4 released
Newsgroups: fr.comp.lang.ada,comp.lang.ada
Followup-To: fr.comp.lang.ada,comp.lang.ada

AWS - Ada Web Server 1.4 release / SOAP 1.2

Authors: Dmitriy Anisimkov, Pascal Obry

We are happy to announce the availability of the AWS 1.4 release. The API could change slightly at this stage but should be fairly stable now.

AWS stand for Ada Web Server. It is not a real Web Server like Apache. It is a small yet powerful HTTP component to embed in any applications. It means that you can communicate with your application using a standard Web browser and this without the need for a Web Server. AWS is fully developed in Ada with GNAT.

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

The SOAP implementation has been validated on http://validator.soapware.org/.

Validation: AWS 1.4 has been compiled and has passed all tests on:

Windows XP, GNAT 3.16a1 and 5.01a

GNU/Linux x86, GNAT 3.16a1 and 5.01a

SPARC Solaris 8, GNAT 3.16a1

Others platforms / compiler version combinations have not been tested, it does not mean that it's not working.

Previous version of AWS have been build on FreeBSD 4.1 and MacOS X.

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

FTP, HTTP & SMTP

From: tmoran@acm.org
Date: Mon, 29 Dec 2003 20:19:15 GMT
Subject: Announce: Ada Internet tools
Newsgroups: comp.lang.ada

FTP, HTTP, and SMTP packages, with example programs, are now available at www.adaworld.com, thanks to Stephane Richard. Click on Ada Projects, then Ada Internet Projects, then Internet Protocols. A .zip file has the specs. Another zip file has bodies implemented with NC, a high level winsock binding based on, but Not requiring, Claw. The third zip file has NC, with a short example program that fetches a webcam picture of current Silicon Valley weather.

ZLib Bindings

From: anisimkov@yahoo.com (Dmitriy Anisimkov)
Date: 1 Dec 2003 23:28:32 -0800
Subject: An: ZLib-Ada
Newsgroups: comp.lang.ada

Latest version of the popular compression/decompression library ZLib v 1.2.1 contains thick Ada binding in the contrib/ada folder.

http://www.zlib.org

Fortran to Ada Translators

From: Randy Brukardt <randy@rrsoftware.com>
Date: Wed, 3 Dec 2003 14:08:01 -0600
Subject: Re: Fortran to Ada
Newsgroups: comp.lang.ada

> I have been looking for a (preferably free) Fortran (77, 90) to Ada translation tool. Any ideas where I can find such a tool?

Ada Solutions has such a tool, but it isn't free. http://www.ada-solutions.com/

From: Gautier Write-only <gautier@fakeaddress.nil>
Date: Wed, 03 Dec 2003 21:53:38 +0100
Organization: Cablecom Newsserver
Subject: Re: Fortran to Ada
Newsgroups: comp.lang.ada

f2a is quite good. Translated a many numerics sources with it. You need pre- and after- translation work of course. And you discover often a good load of bugs never detected in the Fortran code...

URL: ftp://ftp.usafa.af.mil/pub/dfcs/carlisle/usafa/for2ada95/

MaRTE OS 1.4 - Minimal Real-Time Operating System for Embedded Applications

From: Mario Aldea Rivas <aldeam@unican.es>
Date: Thu, 08 Jan 2004 13:08:06 +0100
Subject: MaRTE OS V1.4 released

MaRTE OS version 1.4 released on December 22nd!

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

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

In this version, new functionality has been added:

- New POSIX functionality included:
* Semaphores.
* pthread_setschedprio() function.
* "Dynamic Package Initialization" (pthread_once() function).
- Application-defined scheduling interface improved.
- Added PCI bus support (Developed by José María Martínez).
- Improved dynamic memory management: Included the "Two Levels Segregate Fit memory allocator" (TLSF) algorithm developed by Miguel Masmano Tello (Universidad Politécnica de Valencia)
- Created scripts 'mgnatbind' and 'mgnatlink'. Now 'mgnatmake' can be used to compile applications split into several directories.
- Hardware interrupts management operations. Based on a POSIX draft for "Interrupt Control API" (P1003.2X/D1.0, February 2001).
- Drivers for standard output, input and error are now integrated in the general MaRTE drivers framework.
- Other minor improvements and bug fixes.

Coming soon:

- Implementation of RT-GLADE (modification of GLADE, the current GNAT implementation of the Ada 95 Distributed Systems Annex), to support the development of distributed applications with real-time requirements. Project developed by Juan López Campos or J. Javier Gutiérrez García (Universidad de Cantabria).
- Graphics library (port of the SVGAlib). Project developed by José Luis Mantecón (Universidad de Cantabria).

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

For comments, suggestions, doubts, problems, etc., send an e-mail to Mario Aldea Rivas: mailto:aldeam@unican.es
[…]

Ini File Reader Package

From: Stephen Leake <Stephe.Leake@nasa.gov>
Date: 29 Oct 2003 15:22:06 -0500
Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)
Subject: Re: Ini reader package
Newsgroups: comp.lang.ada

> Does anyone know if there is some kind of ini file reader/writer package? Otherwise i'll write one myself.

Grace.Config_Files reads Java properties syntax; that satisfies "some kind of ini file", unless you meant "exactly Windows ini file format".

http://savannah.nongnu.org/projects/grace

[...]

Auto_Text_IO & SAL

From: Stephen Leake <Stephe.Leake@nasa.gov>
Date: 18 Nov 2003 11:25:15 -0500
Organization: NASA Goddard Space Flight Center (skates.gsfc.nasa.gov)
Subject: SAL, Auto_Text_IO versions released
Newsgroups: comp.lang.ada

Driven by the release of "webcheck", I've released new versions of SAL (now 1.51; my Ada library of useful stuff, mainly data structures) and Auto_Text_IO (now 3.01; automatically generates Put and Get Text_IO subprograms for Ada types).

The changes mostly involve the build structures; they are cleaner.

Auto_Text_IO can now be nicely integrated with ACT's GPS, thanks to some kind sole whose name I've lost :(.

Grace.Config_Files is now also supported as SAL.Config_Files; that's easier for me to use and maintain.

http://www.toadmail.com/~ada_wizard/

AdaCL - Ada Class Library

From: Martin Krischik <krischik@users.sourceforge.net>
Date: Wed, 21 Jan 2004 20:20:16 +0100
Organization: AdaCL
Subject: [Announcement] AdaCL 4.0.0 (cgi) released.
Newsgroups: comp.lang.ada

CGI Suport has been added. Apart from all CGI features AdaCL can read normal HTML files, change them on the fly to add dynamic content and send the result to the web server.

Changes:

A bug in GNAT eventually forced me to change all filenames to lower case.

Abstract:

AdaCL provides library services for scripting in Ada:

- A text search and replace library.
- a cgi binding.
- execution of external programs including I/O redirection with Ada.Text_IO. Unlike GNAT.OS_Lib AdaCL lets you wait on a given asynchronous process instead of just the first to end.
- a garbage collector.
- Booch components for indefinite elements.
- trace feature - very handy for CGI (no debugger, no console output).

Regards, Martin Krischik,
krischik@users.sourceforge.net

http://www.ada.krischik.com

Ada-related Products

ACT - XML/Ada 1.0

From: Emmanuel Briot <briot+spam@act-europe.fr>
Date: Fri, 07 Nov 2003 10:38:32 +0100
Subject: Announce: XML/Ada 1.0 released
Newsgroups: comp.lang.ada,fr.comp.lang.ada

We are happy to announce the release of a new version of XML/Ada (1.0). This is a set of Ada packages that can be used to manipulate XML streams. It includes a full XML parser (including for the DTD part), as well as SAX 2.0 and DOM 2.0 compliant interfaces (please see the web page and the documentation for more information on these interfaces). It also includes a Unicode module to manipulate and convert Unicode streams.

It passes all of the applicable tests of the official XML conformance test suite.

This new release includes a number of fixes for the parser itself and the DOM interface, as well as a number of new features:

- More efficient parser
- Better support for encodings

See the files features and known-problems in the distribution for a more detailed list.

Ada Core Technologies and ACT Europe are providing full support for this tool set. Let us know at sales@gnat.com or sales@act-europe.fr if you are interested in evaluating this library for commercial use.

The XML/Ada team

ACT - GNAT Programming System IDE (GPS)

From: Arnaud Charlet <charlet@gnat.com>
Date: Thu, 20 Nov 2003 18:23:34 +0100
Subject: ANNONCE: GNAT Programming System 1.4.0
Newsgroups: fr.comp.lang.ada

Ada Core Technologies and ACT Europe are pleased to announce the release of GPS 1.4.0, the GNAT Programming System IDE, including binaries for the GNU/Linux, Solaris and Windows platforms.

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

GPS is available at
http://libre.act-europe.fr/gps

New features include, among other things:

- Clickable CVS annotations: you can now click on CVS annotations to jump directly to the corresponding changelog in the history revision of the file.
- Improvements in the entity browser for C and C++
- Browsers can be exported as png images
- Ability to derive a language from another language in xml custom files
- Contextual menus no longer show xref entries when no xref information is available
- Possibility to see the differences between the current file and a specified VCS revision.
- It is now possible to interrupt background tasks using the task manager, or the contextual menus on the progress bars.
- When defining custom actions, the output of a shell command can now be used to launch an external command (for instance to query switches from the project first).
- Compilation output is highlighted in different colors based on the type of the error (Error/Warning/Style)
- Support for Ada, C and C++ switches is now defined through XML files. These files can be modified to add more switches
- New customization capabilities
- Support for preprocessor options in syntax checking
- Improved MDI
- Ability to reformat more Ada construct
- Auto casing of Ada reserved words and identifiers
- Navigation to the next/previous subprogram and start/end of statement
- New casing mode "Smart Mixed"

ACT - PolyORB 1.0p

From: Laurent Pautet <pautet@inf.enst.fr>
Date: 26 Nov 2003 17:41:18 +0100
Organization: ENST, France
Subject: [ANNOUNCE] Release of PolyORB 1.0p
Newsgroups: comp.lang.ada

Ada Core Technologies, ACT Europe, and the PolyORB team are proud to announce the first public release of PolyORB:

PolyORB 1.0p
(http://libre.act-europe.fr/polyorb/).

This release contains a CORBA-compliant instantiation of the PolyORB generic middleware. It includes:

* an IDL to Ada 95 compiler,
* implementations of the Portable Object Adapter (POA), Dynamic Skeleton Interface (DSI), Dynamic Invocation Interface (DII), and Interface Repository (IR),
* implementations of the COS Naming, COS Event and COS Time services,
* implementations of GIOP 1.0, 1.1, and 1.2.

This CORBA implementation can be configured for full tasking, Ravenscar tasking or no tasking runtime, depending on the level of desired functionality for the application, and on the resource constraints for the target.

This release should be considered as a stable implementation of CORBA middleware over PolyORB.

Other instantiations of PolyORB are available in the public PolyORB CVS repository for testing purposes. They will be included in future releases.

PolyORB is primarily developed by Jérôme Hugues, Thomas Vergnaud, and Laurent Pautet (Télécom Paris), and Thomas Quinot (ACT Europe). Fabrice Kordon (LIP6) also participates in the project.

POLYORB Distributed applications and middleware

PolyORB aims at providing a uniform solution to build distributed applications; relying either on industrial-strength middleware standards such as CORBA, the Distributed System Annex of Ada 95, distribution programming paradigms such as Web Services, Message Oriented Middleware (MOM), or to implement application-specific middleware.

Middleware provides a framework that hides the complex issues of distribution, and offers the programmer high-level abstractions that allow easy and transparent construction of distributed applications. A number of different standards exist for creating object-oriented distributed applications. These standards define two subsystems that enable interaction between application partitions:

* the API seen by the developer's application objects;
* the protocol used by the middleware environment to interact with other nodes in the distributed application.

Middleware implementations also offer programming guidelines as well as development tools to ease the construction of large heterogeneous distributed systems. Many issues typical to distributed programming may still arise: application architectural choice, configuration or deployment. Since there is no "one size fits all" architecture, choosing the adequate distribution middleware in its most appropriate configuration is a key design point that dramatically impacts the design and performance of an application.

Consequently, applications need to rapidly tailor middleware to the specific distribution model they require. A distribution model is defined by the combination of distribution mechanisms made available to the application. Common examples of such mechanisms are Remote Procedure Call (RPC), Distributed Objects or Message Passing. A distribution infrastructure or middleware refers to software that supports one (or several) distribution model, e.g.: OMG CORBA, Java Remote Method Invocation (RMI), the Distributed System Annex of Ada 95, Java Message Service (MOM).

PolyORB: a generic middleware with an instance per distribution model.

Typical middleware implementations for one platform support only one set of such interfaces, pre-defined configuration capabilities and cannot interoperate with other platforms. In addition to traditional middleware implementations, PolyORB proposes an original architecture to enable support for multiple interoperating distribution models in a uniform canvas.

PolyORB is a polymorphic, reusable infrastructure for building or prototyping new middleware adapted to specific application needs. It provides a set of components on top of which various instances can be elaborated. These instances (or personalities) are views on PolyORB facilities that are compliant to existing standards, either at the API level (application personality) or at the protocol level (protocol personality). These personalities are mutually exclusive views of the same architecture.

The decoupling of application and protocol personalities, and the support for multiple simultaneous personalities within the same running middleware, are key features required for the construction of interoperable distributed applications. This allows PolyORB to communicate with middleware that implement different distribution standards: PolyORB provides middleware-to-middleware interoperability (M2M).

PolyORB's modularity allows for easy extension and replacement of its core and personality components, in order to meet specific requirements. In this way, standard or application-specific personalities can be created in a streamlined process, from early stage prototyping to full-featured implementation. The PolyORB architecture also allows the automatic, just-in-time creation of proxies between incompatible environments.

PolyORB currently supports the following personalities (in the main development branch, available through CVS access):

* application personalities: CORBA, Distributed System Annex of Ada 95 (DSA), MOMA - MOM for Ada. Interaction between CORBA and DSA partitions has been successfully tested;
* protocol personalities: SOAP, GIOP (CORBA generic protocol layer) and the following instantiations: IIOP (over TCP/IP), DIOP (over UDP/IP for one-way requests), and MIOP/UIPMC (group communication over multicast/IP)
* under development: Web Services personality, an adaptation of the AWS API to PolyORB.

Note: PolyORB is the project formerly known as DROOPI, a Distributed Reusable Object-Oriented Polymorphic Infrastructure.

PegaSoft - BUSH AdaScript Business Shell

From: Ken O. Burtch <kburtch@sympatico.ca>
Date: Thu, 01 Jan 2004 11:59:36 -0500
Subject: ANN: BUSH 0.9.3
Newsgroups: comp.lang.ada

I'm dropping by comp.lang.ada to announce that a new version of BUSH, the AdaScript Business Shell, was released at the end of November. The new version features better script stability and preliminary FreeBSD and PostgreSQL database support. There is also a new BUSH mailing list.

We're looking for examples of BUSH scripts to post to the BUSH Source Code Treasury page. Email contributions to me or post them to the mailing list.

The BUSH home page is http://www.pegasoft.ca/bush.html

To learn more about PegaSoft Canada, visit http://www.pegasoft.ca

Ada and Linux

Debian Ada Packages reached "Testing"

From: Ludovic Brenta <ludovic.brenta@insalien.org>
Date: 27 Jan 2004 21:04:51 +0100
Subject: Update on Ada packages in Debian GNU/Linux
Newsgroups: comp.lang.ada

Yesterday, my Ada packages have all reached "testing" in Debian. In a nutshell, this means that:

- they will be in the next stable release of Debian (i.e. on the CD-ROMS and DVDs). This release is code-named "Sarge" and will become the new "stable" release Real Soon Now(tm). But as always, the only release date you will ever get from a Debian developer is, "when it's ready" :)
- they have passed certain quality tests, most notably to ensure no package is broken by them.
- they are all built using GNAT 3.15p.

For more details, please see http://www.debian.org/devel/testing.

Here is a list of the Ada-related packages that are in Debian:

- 20021112	ada-reference-manual
1.4 1.6	adacgi
1.0 1.8.4.2	adasockets
3.14p 3.15p	asis
19990519	gch (GNAT Style checker)
1.2.0 1.4.3a	gnade (GNU Ada Database Environment)
3.14p 3.15p	gnat (GNU Ada Translator)
3.14p 3.15p	gnat-doc
- 5.3.gnat0.0.20030225	gnat-gdb (Ada-aware version of GDB)
3.14p 3.15p	gnat-glade (GLADE distributed systems)
3.14p 3.15p	gnat-glade-doc
- 1.4.0	gnat-gps (the GNAT Programming System)
- 1.2.0	gvd (the GNU Visual Debugger)

- 20030813	libcharles0
3.14p 3.15p	libflorist-3.15p-1
- 1.2.8	libgtkada1 (GtkAda v.1)
- 2.2.1	libgtkada2 (GtkAda v. 2)
- 1.0	libxmlada1 (XML/Ada)
- 0.7.13.3	topal

In addition, the two packages gnat-3.2 and gnat-3.3 provide the FSF version of GNAT.

I believe that these packages together provide a good foundation to build on. With Debian, you now get a complete Ada development system including compiler, debugger, IDEs, and several libraries, all prepackaged and precompiled for you. Most, if not all, packages are available on three architectures: i386, sparc, and powerpc.

Several packages provide libraries. For them, I always distribute their files (*.ads, *.adb, *.ali, *.so and *.a) according to Florian Weimer's proposed GNU Ada Environment Specification (http://cert.uni-stuttgart.de/ under: projects/ada/gnae.php). In addition to this, I always provide a GNAT project file that makes it very easy to build your applications against the libraries.

If there is software (library or application) that you would like me to package for Debian, please tell me. I will try to provide some of the most popular packages. The more people request a particular package, the more I will consider it popular :) I will also tend to favour well-tested, mature software over experimental releases. However, as Debian is a volunteer effort, I cannot make any promises unless you pay me :)

The release of Sarge is nigh!

From: Ludovic Brenta <ludovic.brenta@insalien.org>
Date: 28 Jan 2004 15:14:14 +0100
Subject: Re: Update on Ada packages in Debian GNU/Linux
Newsgroups: comp.lang.ada

> What happened to GVD? Not that I would use a debugger nearly as much with Ada as with (Visual) C(++), but if I have to, GVD looks nicer than using gbd directly.

GVD was removed from Debian. It was the only package using libgtkada1, and libgtkada1 was orphaned (i.e. no longer maintained) and also removed. The maintainer for GVD did not have time to either adopt libgtkada1, or port GVD to libgtkada2.

This is not a big problem tough, since GVD has been superceded by the GNAT Programming System, which is in testing now. If, however, you would like to contribute some time to re-include GVD into Debian, you are most welcome to do so.

From: Ludovic Brenta <ludovic.brenta@insalien.org>
Date: 28 Jan 2004 15:18:46 +0100
Subject: Re: Update on Ada packages in Debian GNU/Linux
Newsgroups: comp.lang.ada

> Actually you shouldn't use testing too much. Testing does not need to be consistent so it can break your setup. It is better to use unstable or wait for sarge. However, now as we are close to a stable release I think upgrading to testing is quite safe. But after the stable has been released, don't use testing.

Testing is safer than unstable. I use testing all the time. Testing is good for you.

Packages migrate from unstable to testing only after they pass certain sanity checks. In particular they must have no more bugs than the version already in testing, must not break any packages, and must install on all platforms. The discussion you are referring to is one year old. At that time there was change from glibc 2.2 to 2.3, which is a large change and caused some people (not me) to have problems. But the problems were there in unstable too. Now, testing is very close to being frozen for final release and is suitable for daily use.

From: Preben Randhol <randhol+valid_for_reply_from_news@pvv.org>
Date: Wed, 28 Jan 2004 15:52:59
Organization: PVV
Subject: Re: Update on Ada packages in Debian GNU/Linux
Newsgroups: comp.lang.ada

> Testing is safer than unstable. I use testing all the time. Testing is good for you. […]

As I said, "However, now as we are close to a stable release I think upgrading to testing is quite safe."

But testing is never meant to be consistent in between stable releases. Testing completely broke tetex for my while writing my thesis (this August/September) and I had to get back to the stable release (and also delete files to get it work again). Same problem with Gnome. The reason was that not all the package that one needed were available in testing.

Ada and Microsoft

Ada Terminal Emulator for Windows

From: Ross Higson <rosshigson@optusnet.com.au>
Date: Thu, 06 Nov 2003 23:19:32 +1100
Subject: Announce: Ada Terminal Emulator 1.5
Newsgroups: comp.lang.ada

Version 1.5 of the Ada Terminal Emulator is now available. It can be downloaded from:
http://www.members.optusnet.com.au/
under: rosshigson/terminal

The emulator provides a set of terminal emulation capabilities implemented in Ada 95 to run under Windows 95/98/NT/2000, along with various demonstration and application programs.

All source code for the emulator is provided under the GNU General Public License. The package was developed using GNAT and GWindows.

Version 1.5 is a significant improvement over the initial release of this package. Modifications include bug fixes, improvements in the VTxxx and ANSI emulation capabilities, and improvements in the compatibility of the telnet application with various Win32, Unix and VMS based telnet servers.

A summary of all changes is available at the above URL.

Special thanks to Simon Clubley for testing the emulator against several different VMS platforms, and suggesting valuable improvements.

[See also "Ada Terminal Emulator for Windows" in AUJ 24-3 (Sep 2003), pp.150-151. -- su]

SYSAPI - Windows 64-bit File I/O Bindings

From: Ekkehard Morgenstern <ekkehard.morgenstern@onlinehome.de>
Date: Mon, 15 Dec 2003 15:18:55 +0100
Organization: 1&1 Internet AG
Subject: [announcement] SYSAPI and SYSSVC for Windows
Newsgroups: comp.lang.ada

I'd like to announce the new version of my SYSAPI / SYSSVC packages for Windows.

This version provides the following features:

- uses features of Windows 2000 and XP (if present).
- synchronous 64-bit file I/O.
- asynchronous 64-bit file I/O.

The package contains:

- the SYSAPI DLL and import library for GNAT 3.15p or higher
- the source code and makefile for SYSAPI DLL (Visual C/C++ 7.0)
- the Ada spec and body files for SYSAPI and SYSSVC.
- two test programs with Ada source and executables.

The package requires:

- GNAT 3.15p or higher, might also work with other Ada compilers.
- Visual C/C++ 7.0 (only for recompilation of SYSAPI), might also work with other compilers.

You can download the package here:
http://www.ekkehardmorgenstern.de/
under: ada-sysapisvc-03121501.zip

Since I'm a relative Ada newbie, it'd be great if someone could look over it and tell me about inconsistencies, style problems or errors.

The package has been thoroughly tested, but I cannot give any guarantee that it will always work under any circumstance. Use at your own risk!

Note: The Windows Platform SDK documentation now (02/2003) states that asynchronous file I/O is possible with Windows 95, 98 and ME (i.e. the comments from some years before where it said it wasn't possible have been removed). It could be that it's still not possible with those OSes, I cannot verify this. If you can, please notify me of the results.

From: Ekkehard Morgenstern <ekkehard.morgenstern@onlinehome.de>
Date: Mon, 15 Dec 2003 19:38:21 +0100
Organization: 1&1 Internet AG
Subject: Re: [announcement] SYSAPI and SYSSVC for Windows
Newsgroups: comp.lang.ada

> What are they? What do they do?

SYSAPI is a package that interfaces directly to the Microsoft Windows API. It's not a direct binding to the Windows DLLs however; it is a DLL that interfaces to Windows. The reason for this is that I'd like to encapsulate access to the Windows API in a way that can bridge platform dependencies. There are different platforms of Windows: Windows 95, 98, ME, NT, 2000 and XP; all are distinct with distinct features, and they need specialized code to handle them properly.

I tried using the Win32Ada binding, but it covers only Windows NT 4.0 and Windows 95, which are both very outdated (from 1995). So I got the idea to write my own. GNAT's tools for Windows DLL binding also seem to be outdated, and so that's another reason for providing my own abstraction. And while it would be possible perhaps to create import libraries to use directly from Ada, I think it's better to encapsulate the system dependencies away.

SYSSVC provides an Ada server task to access the SYSAPI library from a central point, and with an Ada interface. So it's possible to write Ada programs that use the features of SYSAPI without knowing of the underlying platform (Windows in this case). I plan to port SYSSVC to other platforms as well.

I will also add windowing and graphics support (especially for DirectX Graphics, formerly known as DirectDraw and Direct 3D), and also audio and video support.

I know this will take some time, but I think it's a good project to contribute something to the Ada community as well as my own projects.

For now, it contains only file services, namely 64-bit synchronous and asynchronous services. I've seen that the GNAT libraries do not provide those, and hence I had to write my own. I need 64-bit file I/O for a database project of my own (to allow for files bigger than 4 GB).

References to Publications

Crosstalk Articles

From: Rasmussen Karen J Contr OO-ALC/MASEA <Karen.Rasmussen@hill.af.mil>
Date: Mon, 3 Nov 2003 08:41:52 -0700
Subject: The November 2003 Issue of CrossTalk is now available on-line.

The November 2003 issue of CrossTalk, The Journal of Defense Software Engineering is now available on our Web site at: <www.stsc.hill.af.mil>. This month our theme is "Development of Real-Time Software." We examine the additional development requirements, testing, and maintenance factors that come into play with the nature of this often life- or mission-critical software.

First, we begin with a primer in easy-to-understand terms by Dennis Ludwig titled "An Introduction to Real-Time Programming." The author provides a thorough walkthrough of the considerations real-time programmers make regarding hardware, operating system, and programming language options. Readers will learn about the different world in which real-time developers work.

Next is "The Ravenscar Profile for Real-Time and High Integrity Systems" by Brian Dobbing and Alan Burns. These authors present the Ravenscar model for building safe and reliable real-time systems. They explain how developers using this profile can establish high confidence levels in concurrency properties and requirements within international standards early in the development life cycle.

In "Software Static Code Analysis Lessons Learned," author Andy German shares his experiences from developing safety-critical, real-time systems. He defines static code analysis, reviews some of the tools, and shares lessons learned at The United Kingdom Ministry of Defense.

We end our theme article line up with Timothy J. Budden's article "Decision Point: Will Using a COTS Component Help or Hinder Your DO-178B Certification Effort?' Budden describes how the demands of DO-178B certification can be achieved with commercial off-the-shelf modules if the vendor is a willing partner who understands what is expected under this type of grueling development and verification process.

We begin our supporting articles this month with "Defining a Process for Simulation Software Vulnerability Assessments" by Dr. John A. Hamilton Jr., Col. Kevin J. Greaney, and Gordon Evans. These authors describe the process developed by the U.S. Missile Defense Agency and Auburn University to evaluate the potential vulnerabilities in shared simulation software as a means of risk mitigation.

Next Dan W. Christenson and Lynn Silver discuss issues for tying software and hardware together in "Developing a Stable Architecture to Interface Aircraft to Commercial PCs." The authors introduce a development architecture that maintains the strengths of traditional architectures and eliminates some of the weaknesses and inefficiencies.

Lastly, in our online article "The Probability of Success," Walt Lipke explains the statistical methods applied to the earned value indicators and cost and schedule performance indexes, and introduces a Performance Window graphic as the outcome of this application.

We hope this issue of CrossTalk adds to your knowledge of developing real-time software, including the additional software requirements inherent in critical, real-time applications. Whether a beginner or a seasoned developer, these articles are intended to address the differences in non-real-time and real-time software that will help you and your team build and buy software better.

Pam Bowers-Palmer, Managing Editor

Static Verification and Extreme Programming

From: rod.chapman@praxis-cs.co.uk (Rod Chapman)
Subject: ANN: XP and Static Verification paper from SIGAda now on-line
Date: 14 Jan 2004 09:43:34 -0800
Newsgroups: comp.lang.ada

I'm pleased to say that our paper from SIGAda 2003 "Static Verification and Extreme Programming" is now on-line at www.sparkada.com

There have been some threads recently in this and other groups pertaining to Praxis, SPARK, and the use of XP in some of our projects such as SHOLIS and the MULTOS CA, so I hope people find this interesting. I also hope this paper might stir up the XP community a bit... :-)

DDC-I Online News

From: jc <jcus@ddci.com>
Date: Fri, 9 Jan 2004 12:27:32
Subject: Real-Time Industry Updates - News from DDC-I
To: Y9DK Jan 2004 Online News <jcus@ddci.com>

DDC-I Online News. January 2004, Volume 5, Number 1 - [http://www.ddci.com/ under: news_vol5num1.shtml] A monthly

news update dedicated to DDC-I customers & registered subscribers.

Let Us Help Make Your World a Little Easier!

XTOFF-to-ELF Conversion Tool Now Available for TADS Customers

Thoughts from Thorkil - Endianism: Byte and Bit Numbering

A Tale of Two Successful Projects: Pairing Re-visited

From: jc <jcdk@ddci.com>
Date: Fri, 30 Jan 2004 17:18:56
Organization: DDC-I
Subject: Real-Time Industry Updates - News from DDC-I
To: Z9 Feb2004 Online News DK <jcdk@ddci.com>

DDC-I Online News. February 2004, Volume 5, Number 2 - [http://www.ddci.com/ under: news_vol5num2.shtml] A monthly news update dedicated to DDC-I customers & registered subscribers.

Thoughts from Thorkil - Bit Testing in Integer Values

3rd Party Update - Ada Distilled: A Compact Introduction to the Ada Programming Language

A Few Good Patterns - To Help Deal with Change

SPARK Book Review

From: Rick Maffei <richard.a.maffei@lmco.com>
Date: Tue, 11 Nov 2003 12:38:38 -0500
Organization: Lockheed Martin Corp, Valley Forge PA
Subject: John Barnes' "High Integrity Software" Book Review
Newsgroups: comp.lang.ada

The latest edition of The Risks Digest (newsgroup comp.risks or http://catless.ncl.ac.uk/Risks) contains a short review of John Barnes' book "High Integrity Software". The reviewer is a regular in The Risks Digest, and in the years I have been reading it, I have never read a positive review of a book by Mr. Slade...except until today!

Ada Inside

Ada Binding to the Java Native Interface

From: Wojtek Narczynski <wojtek@power.com.pl>
Date: Tue, 13 Jan 2004 23:56:35 +0100
Organization: Power Media Sp. z o.o.
Subject: Re: interfacing Java from Ada
To: team-ada@acm.org

> which is currently the easiest way to interface a Java program from Ada code using GNAT? Any examples available?

Please tell us more about your interfacing needs.

You may also check out: http://www.acenet.com.au/~gbull/

This is something like JNI for Ada.

[See alsa AUJ 20-4: Interfacing Ada to Java Technology -- su]

Indirect Information on Ada Usage

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

From: Cas <esacas@SNET.NET>
Date: Thu, 18 Dec 2003 16:05:57 -0500
Subject: Sr. Embedded Software (Ada) Engineering Opportunity
To: team-ada@acm.org

[...] Embedded Software Engineer. Preferred Education: Master of Science in Computer Science or Software Engineering. Preferred Experience: Preferred Skills: - Experience with Ada-83 and other high level languages - Knowledge of SEI CMM Level 4 standard work processes. Familiarity with avionics systems. Knowledge of MIL-STD-1553B. Familiarity with cockpit display software. [...]

http:// job.monster.be/
Subject: BE-Brussels-Brussels-Ada 95 Software developer (on Unix)

[...] Currently we are looking for a: Ada 95 Software developer (on Unix) in an Air Traffic Management environment. [...]

http:// job.monster.be/
Subject: BE-Brussel-Ada Software Engineers

[...] Education: Civil Engineer, Industrial Engineer, Graduate in Computer Sciences or similar experience. Knowledge of a programming language preferably Ada 83-95 or C++. Developer & analyst designer, team spirit and good methodology [...]

On Creating More Ada Jobs

From: clubley@remove_me.eisner.decus.org-Earth.UFP (Simon Clubley)
Date: 16 Dec 2003 12:27:07 -0600
Organization: Encompasserve
Subject: Ada Job Market, Was: Re: SIGada Conference
Newsgroups: comp.lang.ada

> I'd like to see Ada get somewhere, but there needs to be a better metric than the business growth of some few companies. Jobs in Ada, new project starts in Ada, classes taught in Ada, etc.

If you have a copy of Dr Dobb's Journal for October 2003, have a look at page 52.

The author has done an analysis of job offers for the period July 2002 to June 2003 that specified programming language requirements.

I won't post the table (because I'm not sure what the copyright issues are, and anyway it's a detailed table), but the author's table claims[1] that Ada is running around the 5% mark (+/- 1%) for much of that period.

It was interesting to note however that it increased a couple of percent in May/June 2003.

For comparison C++ is running at around the 50% (+/- several %) mark, and Pascal is running at around 0.2%.

[1] I use "claims" because I am finding it hard to believe that 5% of all programming jobs specify Ada, however nice that would be. My viewpoint here may be tainted by the fact that I am in the UK, and closer to 0% appears to be a more accurate figure here. [Fortunately, my interest in Ada is more of a personal interest ...]

Simon Clubley, clubley@remove_me.eisner.decus.org-Earth.UFP

From: Marin David Condic <nobody@noplace.com>
Date: Wed, 17 Dec 2003 13:35:12 GMT
Subject: Re: Ada Job Market, Was: Re: SIGada Conference
Newsgroups: comp.lang.ada

My gut reaction based on prior searches of Monster, et alia, would make 5% sound optimistic. It would also likely be *very* location sensitive since Ada has traditionally been employed largely with defense contractors and they're not spread across the U.S. evenly like peanut butter.

But assume for a moment that this is true. C++ gets 50% and Ada gets 5%. Students exiting college are going to be looking for jobs. What skill set will they likely want to have in order to face that market? When companies start development of software related tools, will they be looking to satisfy the 5% market or the 50% market?

This has the potential to be a self-fulfilling prophecy, so one should not look at it as doom and gloom, but rather as a warning. Unless Ada starts getting bigger numbers, all the incentives are to go with something else. People will go with a niche language if they think it is on the way *UP*, but not if it appears to be on the way *DOWN*. Hence, Ada needs to do something to excite the potential user community and create the impression of going somewhere *new* so it starts a "get on board" trend.

From: Steve <nospam_steved94@comcast.net>
Date: Thu, 18 Dec 2003 04:49:40 GMT
Subject: Re: Ada Job Market, Was: Re: SIGada Conference
Newsgroups: comp.lang.ada

Do you realize you're trashing the language with your statements? Taking news that might be encouraging to Ada enthusiasts (the people you're likely to find on this group) and saying it's garbage. It doesn't help.

From: Marin David Condic <nobody@noplace.com>
Date: Thu, 18 Dec 2003 13:10:47 GMT
Subject: Re: Ada Job Market, Was: Re: SIGada Conference
Newsgroups: comp.lang.ada

We will not help Ada by deluding ourselves into thinking everything is just fine. Seeing a handful of companies experiencing some growth - while good for those companies - doesn't imply that Ada is on a nice, healthy, upward spiral. If we want to live in a rainbow, gumdrop, pastel unicorn, sugar-sweet, fairy-story land, then we can believe that Ada is doing just great and will be around and growing for years to come and nothing really needs to be done.

OTOH, we could get serious about what is happening in industry and understand that the bulk of the software population sees Ada as something old and beyond its useful life and not to be taken seriously for the future. Given that understanding of reality, we might want to *save* Ada by doing something that would create a "New and Revitalized" impression of the language. Give Ada some major new capabilities that might actually reverse that impression people have of it being over the hill.

Ada in Context

Reaffirmation of POSIX Ada Binding

From: Ted Baker <baker@cs.fsu.edu>
Date: Tue, 20 Jan 2004 18:03:05
Subject: [Ada-Comment] reaffirmation ballot on posix ada binding, need balloters
To: ada-comment@ada-auth.org

The POSIX 1003.5 a-c standards are due to expire unless reaffirmed. If you would like to see these standards continue (with no changes) I hope you will consider signing up for the balloting group, and pass this information along to other members of the Ada community. [...]

***** BALLOT *****

The Computer Society/Portable Applications Standards Committee invites you to ballot on the following:

Title: Std. 1003.5-1999 Reaffirmation: IEEE Standard for Information Technology—POSIX ® Ada Language Interfaces-Part

1: Binding for System Application Program Interface (API)

Scope: This standard is part of the POSIX ® series of standards for applications and user interfaces to open systems. It defines the Ada language bindings as package specifications and accompanying textual descriptions of the application program interface (API). This standard supports application portability at the source code level through the binding between ISO 8652:1995 (Ada) and ISO/IEC 9945-1:1996 (IEEE Std 1003.1-1996) (POSIX) as amended by IEEE P1003.1g/D6.6. Terminology and general requirements, process primitives, the process environment, files and directories, input and output primaries, device- and class-specific functions, language-specific services for Ada, system databases, synchronization, memory management, execution scheduling, clocks and timers, message passing, task management, the XTI and socket detailed network interfaces, event management, network support functions, and protocol-specific mappings are covered. It also specifies behavior to support the binding that must be provided by the Ada language.

Purpose: See Scope.

If you are interested in participating in this electronic ballot, you must respond by 2004-02-19, 11:59 PM Eastern Time by filling out the web-based form at http://standards.ieee.org/ under: cgi-bin/bsignup/0000709

Please Note: For your information, this standard is approximately 900 pages. [...]

Please email your questions to the IEEE-SA Balloting Center at saballot@ieee.org

Number Bases Greater than 16

From: Peter Hermann <ica2ph@sinus.csv.ica.uni-stuttgart.de>
Date: Mon, 3 Nov 2003 12:52:58
Organization: Comp.Center (RUS), U of Stuttgart, FRG
Subject: Re: number_base
Newsgroups: comp.lang.ada

>> because of an immense practical value in data processing I would like to have an enlargement of the number_base

> What are some examples of the practical value of number bases greater than 16?

quite a lot of ASCII-encoding/decoding purposes.

There are many examples, e.g. in GPS and moving map usage etc. Only two of them: e.g. look at *.igc-files of e.g. http://www.segelflugszene.de/ under: olcphp/olc-i.php?olc=olc-i
or look at time6stamp of http://www.csv.ica.uni-stuttgart.de/ under: homes/ph/adapilotresources/basic_tools/calenday.ads

From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>
Date: Mon, 03 Nov 2003 11:32:02 +0100
Subject: Re: number_base
Newsgroups: comp.lang.ada

>> Has anyone in this NG written an Ada program that used a number base other than 2, 8, 10, or 16? These days, it would be hard to find an example that uses base 8. Looking at current usage, it is hard to make the case for more number bases.

> Ha, USSR produced first and last(but don't sure) non-binary computer with base 3
Really!!! ;-)

Wasn't it actually e or Pi? I remotely remember something like that. [...]

> I think as space-oriented, etc, ada should support all aliens bases, who know, how many fingers they have?

Or whether that number be countable! ☺

Dmitry Kazakov, www.dmitry-kazakov.de

From: Lutz Donnerhacke <lutz@iks-jena.de>
Date: Mon, 3 Nov 2003 10:33:06
Organization: IKS GmbH Jena
Subject: Re: number_base
Newsgroups: comp.lang.ada

> Wasn't it actually e or Pi? I remotely remember something like that.

e is the most efficient base to store large numbers. Unfortunately hardware to the base e is difficult to develop. That's why most systems choose the next natural number around e to store date to this base.

From: Frank J. Lhota <lhota.adarose@verizon.net>
Date: Mon, 03 Nov 2003 17:11:01 GMT
Subject: Re: number_base
Newsgroups: comp.lang.ada

> Seriously, What's about base-60 ? Our seconds/minutes/etc? (still joking) Base 12?

The ancient Babylonians counted using groups of 12. The countries that Babylon traded with, of course, counted using base 10, creating an early standards conflict. The Babylonians overcame this conflict somewhat by making frequent use of the number 60, the least common multiple of 10 and 12. The use of 60 in Babylonian mathematics lead directly to our current system of measuring both time (60 minutes in an hour, 60 seconds in a minute) and angles (360 degrees in a full circle).

Speaking of the ancients, should Ada be extended to support Roman numerals? Why can't I print 2003 as "MMIII"?

Making Changes to the Language

From: Robert I. Eachus <rieachus@comcast.net>
Date: Mon, 03 Nov 2003 11:26:39 -0500
Subject: Re: Clause "with and use"
Newsgroups: comp.lang.ada

> OK, I think I'm starting to get it. Redundant keystrokes are what make Ada the great language it is! I guess I was just too stubborn to see that sooner. Now that I see the light, let me propose a way to use even more redundant keystrokes:

No, what makes Ada the wonderful language that it is, is the effort put into scrutinizing every proposed change to see what effects it will have on all the "-ilities," especially in this case readability and maintainability, to decide whether a proposed improvement really will be an improvement overall. [...]

Little Endian, Big Endian

From: Peter Hermann <ica2ph@csv.ica.uni-stuttgart.de>
Date: Wed, 17 Dec 2003 11:43:16 +0100
Subject: Re: Little/big endian
To: ada-france@ada-france.org

supplementary info:

for quick and dirty actions:
http://www.csv.ica.uni-stuttgart.de/homes/ph/adakurs/stream2ph.adb

http://www.csv.ica.uni-stuttgart.de/homes/ph/adakurs/invert4bytes.adb

for more insight: Endian-independent record representation clauses

Source: ACM SIGAda Ada Letters archive Volume XIV, Issue 1 Jan./Feb. 1994 table of contents. Pages: 27 – 29. Year of Publication: 1994. ISSN:1094-3641. Author: Norman H. Cohen. Publisher: ACM Press, New York, NY, USA

Porting from Ada83 to Ada95

From: Jeffrey Carter <jrcarter@acm.org>
Date: Thu, 29 Jan 2004 18:18:39 GMT
Subject: Re: Ada 83 to 95
Newsgroups: comp.lang.ada

> Hello! I'd like to know if there is a list of old commands in Ada 83 that can't work in Ada 95 (Also the new ones, if present) since I need to do a porting, thanks in advance!!

The first step is to try recompiling your code with an Ada-95 compiler and see what happens. Backward compatibility was an important goal of the Ada-95 effort. Recompiling is all you have to do most of the time. If you encounter specific problems we can address them.

You can also review an article about the main problem areas at http://www.adaic.org/ under: learn/tech/8395comp.html

Multitasking Without an OS

From: Tarjei T. Jensen <tarjei@online.no>
Date: Sun, 02 Nov 2003 10:45:06 +0100
Subject: Re: GNAT tasking without an OS?
To: GNAT Discussion List <gnatlist@lyris.seas.gwu.edu>

> I'd like to run my multitasking programs on CPU without an OS. (A small subset of Ada's tasking features would be just fine for me). I've already found out that I have to adjust the gnat run-time-system in a way that it no longer depends on the OS's tasking system.

Have a look at RTEMS at http://www.oarcorp.com/.

From: Jose Gonzalez <correo_ada@hotmail.com>
Date: Sun, 02 Nov 2003 23:13:44 +0100
Subject: Re: GNAT tasking without an OS?
To: GNAT Discussion List <gnatlist@lyris.seas.gwu.edu>

See http://marte.unican.es/:

"MaRTE OS is a real-time kernel for embedded applications that follows the Minimal Real-Time POSIX.13 subset. Most of its code is written in Ada with some C and assembler parts."

Cross-compilers for 8-bit Embedded Targets

From: Peter Milliken <peterm@resmed.com.au>
Date: Fri, 14 Nov 2003 06:50:54 +1100
Subject: Is it possible to build an Ada cross-compiler for an 8-bit embedded target now that gcc 3.X has support for Ada?
Newsgroups: comp.lang.ada

Subject pretty much says it all.

Now that you can build an Ada compiler using gcc 3.X (well, according to the build instructions of gcc 3.x :-)), I began to wonder whether it was possible to build a cross-compiler for an 8-bit target processor. The 8-bit processor family in question is already supported by gcc (I believe).

When I say "possible", I just mean "follow the cross-compiling instructions and out pops an Ada compiler" - I don't mean, "spend months of work patching various files and then you might have an Ada compiler" :-)

I would dearly like to convince some managers here that Ada is a reasonable alternative to C/C++ for product development. I don't mind spending a lot of my own time re-writing my current C code in Ada to help prove the point, but I need the tools to at least contemplate giving it a go :-).

The cross-compiler would be built and run on a Windoze box (just answering (one of) the obvious questions :-)).

From: Peter Amey <peter.amey@praxis-cs.co.uk>
Date: Fri, 14 Nov 2003 09:53:09 +0000
Subject: Re: Is it possible to build an Ada cross-compiler for an 8-bit embedded target now that gcc 3.X has support for Ada?
Newsgroups: comp.lang.ada

Given that there is already a C compiler for your target, one approach might be to _design_ in Ada and then use an Ada->C compiler/translator to convert the design to C for compilation. In practice you can tie the translator and C compiler so tightly together that it looks like an Ada compiler.

The biggest problem with this approach in general is the size of the run-time library you need to support all of Ada (esp. tasking, exception handling etc.) and writing such a library for a small processor. We have had some success in avoiding these problems by using SPARK as the design language. This gives several benefits:

1. SPARK is designed to need zero (or at least very little) run-time library support so all the generated C is directly and easily traceable back to some SPARK source statement.

2. Using the SPARK tools you can easily prove the code to be exception free (e.g. no range violations) and this means the translation can be greatly simplified because it does not have to include the run-time checks that a full Ada compiler would place in the code. The SPARK tools can be told how big your C int is, for example, and can ensure there are no overflows in the translated code.

3. You can do a great deal of strong verification on the SPARK "design model" before generating the C and therefore greatly reduce the cost of the testing process.

We have submitted a paper to Ada Europe 2004 on this topic.

From: Randy Brukardt <randy@rrsoftware.com>
Date: Mon, 17 Nov 2003 15:57:57 -0600
Subject: Re: Is it possible to build an Ada cross-compiler for an 8-bit embedded target now that gcc 3.X has support for Ada?
Newsgroups: comp.lang.ada

To not answer your actual question at all, but...

As someone else pointed out, the biggest cost/overhead is porting the runtime libraries. Speaking for Janus/Ada, porting the code generator/compiler usually only takes a month or so (and that's for building a code generator from scratch). But building the basic runtime support for exceptions, tasking, floating point (which usually has to be emulated on small proc-

essors), etc. can take a lot of time. In addition, that support can make programs quite large.

Because of this, I think you'd have trouble making acceptable programs with a general-purpose compiler like GCC.

Janus/Ada was originally built for Z-80 and 8086 processors without much memory. Thus, we only load floating point and tasking support when they are used. That isn't very valuable on standard systems like Windows these days, but it matters a lot on 8-bit systems.

The only reason that we don't support 8-bit processors is a lack of demand. Everyone seems to use a different processor, and thus we cannot really justify the investment building code generators (easy) and runtime libraries (not as easy) for each one.

Of course, we're always interested in customization projects, but that won't help you convince your manager...

Promoting Ada

From: Warren W. Gay VE3WWG <ve3wwg@cogeco.ca>
Date: Thu, 13 Nov 2003 12:49:02 -0500
Subject: Re: Re-Marketing Ada (was "With and use")
Newsgroups: comp.lang.ada

> So we've got another language revision coming out. The problem is it is more of an "incremental" revision - relatively minor enhancements to the language rather than anything new and big with respect to capabilities. If Ada had some revolutionary new thing to offer, it might have something to hype within the media. (Like if it had a library that went beyond what people traditionally expect?) If it had something *new*

I think most of us would agree that Ada was ahead of its time when it came out. [...] Even in 1995 (including up to present) the kind of thing you hear now is "I remember Ada". Immediately the association is that it is "old", even though C is just as old (younger folks often haven't even heard of it, unless the University taught it).

Even though Ada has been updated, and going through another update, people seem to remember it as a "once was" technology. [...]

I think one possible prong of a concerted new Ada marketing campaign needs to be: "Ada is still ahead of its time" [...] or some such.

Somehow we have to shake this impression that Ada is old, outdated, fat or ugly.

I have said this before, if Ada was marketed by a company, the best approach might be to rename the technology with some improvements. Then in the fine print say something along the lines of "The X language was based primarily upon the lessons learned from Ada". Then the press, might view this as an exciting new technology and get the printing presses rolling again. [...]

Warren W. Gay VE3WWG

http://home.cogeco.ca/~ve3wwg

From: Marin David Condic <nobody@noplace.com>
Date: Tue, 27 Jan 2004 13:39:59 GMT
Subject: Re: Personality Conflict was: why Ada is so unpopular ?
Newsgroups: comp.lang.ada

> In addition to the responses given, on the thread "why Ada is so unpopular", we must remember that it never

Ada might have done better with two simple overriding rules: A) Know who your customer is. B) Figure out how to make that customer unbelievably happy.

Ada bungled the job with embedded developers early on because it didn't pay attention to the needs of the guys in the trenches who were doing the job and had the ability to say to any management: "If you make me use Ada, I can't get the job done" (They may not have the power to select the language, but they sure have the veto power over a language! ;-) In the early days it was too big, too slow, too buggy and too expensive - not to mention it didn't provide features that the garden variety embedded developer considered essential to getting the job done.

As languages go, Ada probably had more money thrown at it than any other language in history. It should have used some of that money on "Market Research".

O.K. So the pooch got screwed on that one. How to fix it now? Start with (A) above - figure out what market you want to address. Then go to (B) - Ask people in that market what they want out of a language. Maybe offer them some suggestions for possible new and wonderful capabilities, and find out what they say. Figure out what they need to get the most possible leverage from the language and give it to them.

That's not exactly rocket science, eh? :-)

From: Randy Brukardt <randy@rrsoftware.com>
Date: Mon, 26 Jan 2004 20:26:14 -0600
Subject: Re: Personality Conflict was: why Ada is so unpopular ?
Newsgroups: comp.lang.ada

Perhaps. Economics says that you can't give the same service to free customers as you do to ones that pay money, or pretty soon you don't have any paying customers.

Are any of these other languages maintained by commercial companies without significant other products?

Ada (the language) is built around the notion of multiple vendors competing for your business with some assurance that the language that those vendors support is similar in important ways. If your current vendor is not supporting you well enough, try a different vendor. [...]

I realize that changing compilers isn't always possible.

From: Ludovic Brenta <ludovic.brenta@insalien.org>
Date: 21 Jan 2004 15:31:00 +0100
Subject: Re: why ada is so unpopular ?
Newsgroups: comp.lang.ada

>> I'd like to know how Ada is popular ? And i which countries. I'm asking because I live in Poland and here I couldn't find any firm that use it.

> Well, I wouldn't say that Ada is unpopular. There are other factors to take into consideration:
> 1) Management don't know about Ada.
> 2) Management tend to want the programmers to use languages that are the current fad, i.e. C/C++.

In my view their attitude is more cynical than that. [...] They don't mind that their language of choice has an adverse effect on the quality of software, because they're interested in selling bug fixes, upgrades and maintenance. They also don't mind that disposable programmers will produce disposable software. They're in fact quite happy about it. [...] I hope there are still companies that try to produce quality software.

> 3) I had to learn Ada at uni and I had no idea about before then. I actually love the language, It has so many features not found anywhere else that are (IMO) necessary for development.

Yes. Furthermore, I have found that people who learn Ada often change their attitude regarding software development. They no longer want to develop junk, disposable software; instead they want to develop quality software that lasts. [...]

> 4) Programmers learn what is required of them.
> 5) The DoD (supposedly) dropped all support for Ada and this then looks (to the outsider) that the language is dead.
> I think that if enough programmers get to know Ada, I think that better programming standards will emerge, but it's up to those who know it and those who can tell others about it to spread the word and make sure that others start to use it.

I would like to see more free software developed in Ada. The free software world does not try to produce disposable software, and therefore would benefit from a language that helps improve quality. Perhaps, that way, Ada will become a little bit more mainstream.

Ada and Supercomputers

From: Warren W. Gay VE3WWG <ve3wwg@cogeco.ca>

Date: Mon, 24 Nov 2003 17:05:52 -0500
Subject: New Language Design: (Cray, Sun prep radical software models for petaflops systems)
Newsgroups: comp.lang.ada

Full article at http://www.eet.com/story/OEG20031119S0013

> Getting high-end computer users to rally around a new language will be a tough job given past failures with languages such as Ada, HPF and Prolog, said Zima. Therefore, he said, the Cray team will also show ways of extending Fortran and C++ to deliver the same functionality the new language could offer.

I find it interesting that they classified "Ada" as a failure, and are looking at ways to extend "Fortran" (of all things) and C++. [...]

Warren W. Gay VE3WWG

http://home.cogeco.ca/~ve3wwg

From: Marin David Condic <nobody@noplace.com>
Date: Tue, 25 Nov 2003 13:05:46 GMT
Subject: Re: New Language Design: (Cray, Sun prep radical software models for petaflops systems)
Newsgroups: comp.lang.ada

Depending on what one wants to use as a measure of "Success" - one might judge Ada to be a failure. Ada was supposed to be designed for embedded software development and was supposed to be used for mission critical DoD applications. By the first goal, Ada is a miserable failure when "other" ranks higher than "Ada" in just about every survey of embedded developers when asked "What language do you use?" As for the "Mission Critical DoD" apps? The developers had to be drug into Ada kicking and screaming and as soon as The Mandate was lifted they began to abandon it in droves. So by those measures, Ada is a failure. [...]

From: Ludovic Brenta <ludovic.brenta@insalien.org>
Date: 24 Nov 2003 23:40:21 +0100
Subject: Re: New Language Design: (Cray, Sun prep radical software models for petaflops systems)
Newsgroups: comp.lang.ada

Ada would be a perfect fit. Ada already has tasking built-in for intra-node communications, and it has a Distributed Systems Annex for inter-node communications. I think that creating an optimised Ada run-time environment for massively parallel supercomputers is both feasible and highly desirable.

Furthermore, I think Cray's input into the Ada revision process would be extremely valuable; perhaps some extensions to the standard would help get the most out of NUMA architectures. [...]

GNAT on Crays anyone?

On the Success of SPARK

From: Vinzent 'Gadget' Hoefler <ada.rocks@jlfencey.com>
Date: Fri, 12 Dec 2003 08:51:19 -0800
Subject: Re: SPARK gets another fan
Newsgroups: comp.lang.ada

> This is from the latest "Embedded Muse 90", an e-newsletter put out by Jack Ganssle (he gives permission for non-commercial redistribution):

Yeah. SPARK seems to reach some audience. Yesterday Rod mentioned that the current edition of John's "High Integrity Software" sold better than the first one. Whatever this means when expressed in numbers – it can't be too bad. ☺

Ada Support

From: David Starner <dvdeug@email.ro>
Date: Tue, 27 Jan 2004 01:06:54 GMT
Subject: Re: Personality Conflict was: why Ada is so unpopular ?
Newsgroups: comp.lang.ada

> And certainly, complaining about bugs in a "free" product is counter-productive. You get what you pay for, and if you want bugs fixed in a timely manner, you need to pay (someone) for that service. Otherwise, you are hoping that some paying customer runs into the same problem - and there is no guarantee of that.

If I have a bug in GCC, it generally gets fixed in a timely manner. If it's important enough, a patch may get back-ported from the mainline to the release branch by Debian developers (which is purely a volunteer position.) To me, submitting a bug to a developer is a courtesy; it's usually easy to work around the bug then write up a good bug report, but writing up a bug report means a better program for everyone. [...]

If I can't get the same service from my Ada compiler as I do from my C compiler, that is a valid reason to change languages.

Developing Safety-Critical Applications

From: snarflemike@yahoo.com (Mike Silva)
Date: 22 Dec 2003 21:06:08 -0800
Subject: Re: Certified C compilers for safety-critical embedded systems
Newsgroups: comp.arch.embedded,comp.lang.ada

>>> As for using C, it is a simple language that can be and is used safely by many people.

>> I think a more interesting question is: given a particular quality of programming talent and fixed amounts of time and money, how will software written in C fare against software written in "better" (as determined by safety-critical industry consensus) languages? I think the evidence is overwhelming that it will fare quite badly, meaning it will cost more and/or take more time and/or and have more residual errors.

> Sounds interesting. Can you provide references to such evidence, obtained under the stated conditions?

I think the Ada and SPARK communities can, which is why I've added comp.lang.ada to this thread. For example, here's reference to a 100:1 residual error reduction between C and SPARK, and a 10:1 reduction between C and Ada, with all code having been previously certified to DO178B level A:

http://www.sparkada.com/ under: downloads/Mar2002Amey.pdf

Some more interesting reading (note that MISRA acknowledges that there are better languages than C for safety-critical work):

http://www.sparkada.com/ under: downloads/misracatsil4reader.pdf

This document has a table of language recommendations (search for "Language Recommendations (IEC 1508)"). C is only recommended for SIL1, while it is not recommended for SIL3 and SIL4:

https://www.cis.strath.ac.uk/ under: teaching/ug/classes/52.422/programming.languages.doc

From: David Emery <demery@cox.net>
Date: Fri, 26 Dec 2003 19:42:46 -0500
Subject: Re: Certified C compilers for safety-critical embedded systems
Newsgroups: comp.arch.embedded,comp.lang.ada

Several people have asserted the argument that the large number of users of a given tool implies quality in that tool.

I don't buy it. By this logic, MS Windows should be absolutely flawproof, and the infamous Pentium Divide bug should never have happened.

No amount of usage can make up for design flaws. Popularity is much less about technical qualities than it is about marketing. C, Windows, x86 chips have substantial market share, which is not the same as saying that they have substantial quality by design.

From: Chris Hills <chris@phaedsys.org>
Date: Sat, 27 Dec 2003 16:15:27 +0000
Organization: Phaedrus Systems
Subject: Re: Certified C compilers for safety-critical embedded systems
Newsgroups: comp.arch.embedded,comp.lang.ada

> Several people have asserted the argument […].

We are in comp.arch.embedded..... Ms Desktop tools are not relevant. Besides AFAIK their license excludes any safety critical use.

> No amount of usage […].

Correct for the desktop tools but among *Embedded* tools it does especially where the aim is to go for the high reliability market.

From: Stephen Leake <stephen_leake@acm.org>
Date: 26 Dec 2003 20:58:12 -0500
Subject: Re: Certified C compilers for safety-critical embedded systems
Newsgroups: comp.lang.ada

> That is the problem. A non-validated Ada compiler would be no more value than a good C compiler. [...]

The version of Ada most likely to run on 8 bit machines would have no tasking, no exceptions, and possibly no floating or fixed point, or dynamic dispatching. That would still be a far better language than C! Packages, generics, strong typing in general, aggregates, representation clauses; none of these make demands on the run-time environment, but all are very powerful programming language features.

> Actually a good C compiler eg the Keil C51 that has been extensively used in safety related projects by a large number of people would be better simply because of the empirical field usage compared to a non-validated Ada compiler with a small user base..

Well, if by "better" you solely mean "more trusted", or possibly "more thoroughly tested", I would agree. But I'd still use the Ada compiler, and write thorough unit tests. I don't trust _any_ compiler to not have bugs when running _my_ code.

From: Peter Amey <peter.amey@praxis-cs.co.uk>
Date: Sun, 28 Dec 2003 10:15:43 +0000
Subject: Re: Certified C compilers for safety-critical embedded systems
Newsgroups: comp.arch.embedded,comp.lang.ada

I would much rather concentrate on technical issues here (for example, why deep static analysis of _any_ general-purpose language is impossible; or, why systems of integrities in the better-than 10e-6 failures per hour class _require_ deep static analysis); [...]

> AFAIK they did not make their SPADE C results available to the MISRA-C working group who for the last 3 years have been working on MISRA-C2.

Wrong. We did. Unfortunately the rather stern view we took of what was needed to make C fully-analyseable (basically, a Pascal subset in C syntax) was not seen as being compatible with the apparent aim of the committee: as much C as possible with the minimal restrictions needed to plug the biggest holes. [...]

We don't rubbish C. We rubbish magic where logic is to be preferred. We have well-articulated reasons for saying that C is not well suited for constructing high-integrity systems. The proponents of C for this purpose never seem to present their reasons. All we ever hear are: "there are lots of C programmers around"; "we only ever employ the best people and they don't make those kinds of mistakes"; and, the falsehood, "C, with suitable support tools, is as good as anything else".

We don't take this view because we have alternative tools, we have alternative tools because we take this view! [...]

> So whilst straight Ada *is* better than vanilla C. No one would debate that! Spark Ada is no better than C with a subset, coding standard and using static analysis.... IE much the same constraints as SPARK Ada has over Ada...

Fundamental misconception. SPARK is wholly unambiguous and therefore analyseable in a formal and mathematical sense. An un-annotated subset of C _cannot_ have this property. Analysis of such a language can, therefore, only result in the _detection_ of certain kinds of error, it cannot _prove_ that they have been eliminated.

> I know of projects using C in Railway, space, aero and medical projects.

Use, even widespread use, does not imply suitability. [...]

From: Robert I. Eachus <rieachus@comcast.net>
Date: Sat, 27 Dec 2003 16:33:47 -0500
Subject: Re: Certified C compilers for safety-critical embedded systems
Newsgroups: comp.arch.embedded,comp.lang.ada

> That seems obvious. It's possible to write a C program with *no* residual errors. It may be easier to write a SPARK program with no residual errors, but there's no law that says C programs have to have more errors.

But is it possible to write a C program and an Ada program which implement the same algorithm with no residual errors?

My experience indicates that it is not. This is not saying that you can't build an Ada to C translator, or a C to Ada translator, although the first has been done several times, and the second would be much harder. The problem is that if I have a C program "in hand" and go to translate it into Ada, there are usually hundreds of questions that I end up asking. Some of them may be answered in the specification for the C program--assuming that such a specification exists.

The net result is that even if I start out to write identical programs in C and Ada, the Ada program ends up much more tightly specified. Whether these additional specifications are meaningful in terms of the intent of the program, or are just "accidental" over-specification due to the fact that Ada asks the question and it is easier to answer it than dodge it, the Ada and C programs implement two different specifications. (Or to be formal, the set of specifications that the C program satisfies will normally be a superset of those satisfied by the "equivalent" Ada program.)

Does this make the C program better? I don't think so. I am much more concerned that set of specifications satisfied includes the "real" specification, which often includes requirements omitted from the original written specification. Ada does a much better job of identifying these implicit requirements and getting them formalized.

[...] Formal proofs of correctness are worthless if the requirements are not complete. I like a language that asks the questions that the specification forgot.

From: Scott Moore <samiam@moorecad.com>
Date: Thu, 08 Jan 2004 10:13:40 GMT
Subject: Re: Certified C compilers for safety-critical embedded systems
Newsgroups: comp.arch.embedded,comp.lang.ada

"safety critical C" is an oxymoron if I ever heard one.

Ada vs. Borland Builder

From: Szymon Guz <alpha@skynetSMIECI.VONorg.NOJUSZpl>
Date: Fri, 30 Jan 2004 00:50:17 +0100
Organization: tp.internet - http://www.tpi.pl/
Subject: Ada & Builder
Newsgroups: comp.lang.ada

Lately I started a rather huge project and now I'd like to implement this. That's the main problem. I don't know what to choose.. Ada or Borland Builder. First of all I'd like to make the program work under both Linux and windows but that's not a problem. The real problem is that I need to make some windows in the program. If I implement it in Ada then I'll have to use for example GtkAda but Gtk doesn't look nice, what's more for example glade is so unstable that it can crash while working with that. I'd rather use some stable application for developing software. If I choose Builder I'll have to write it in C++ and that is what I want to avoid too. In work I use Builder and Kylix and I must say that using it is quite easy and nice but debugging takes ages so I'd like to use Ada. Well, what do you think, what should I do ?

From: Stephen Leake <stephen_leake@acm.org>
Date: 29 Jan 2004 20:29:30 -0500
Subject: Re: Ada & Builder
Newsgroups: comp.lang.ada

Use Ada and GtkAda, but not Glade. It's quite easy, once you get used to it.

GtkAda has ways of implementing custom look & feel, so you can make it look as "nice" as you want.

From: James E. Hopper <hopperj@macconnect.com>
Date: Fri, 30 Jan 2004 04:21:20 GMT

Organization: Road Runner High Speed Online http://www.rr.com
Subject: Re: Ada & Builder
Newsgroups: comp.lang.ada

Yes I have been using Glade for Gtk for a while, but in my latest project i got tired of it and decided to just use the GtkAda example code to help me do it by hand. it seems to work fine.

By the way GtkAda lets you run on Mac OS x (which is what I use) and not just Linux and Windows.

From: Preben Randhol <randhol+valid_for_reply_from_news@pvv.org>
Date: Fri, 30 Jan 2004 08:21:58
Organization: PVV
Subject: Re: Ada & Builder
Newsgroups: comp.lang.ada

This is not true at all. Firstly use GtkAda 2 (not Gtkada 1). Secondly if you are developing for both windows and Linux why don't you also develop the program *in* Linux? You will see that it is much easier and better than using Windows. Thirdly it is hogwash to say that GtkAda doesn't look nice. Look at the screenshots here and explain to me what isn't nice: http://gtk-wimp.sourceforge.net/screenshots/ and if you don't like this theme you can always choose a different from here: http://art.gnome.org/themes/gtk2/index.php or make your own theme. I'd say use Ada and GtkAda. [...]

From: Alexandre E. Kopilovitch <aek@VB1162.spb.edu>
Date: Fri, 30 Jan 2004 18:46:52
Subject: Re: Ada & Builder
Newsgroups: comp.lang.ada

Perhaps you should make a prototype with Kylix, and then decide whether to continue with Kylix for you product or switch to Ada, using the experience, which you gained with your Kylix-based prototype for design of Ada/GtkAda-based product.

Alexander Kopilovitch, aek@vib.usr.pu.ru

Ada Art

From: Dan Eilers <dan@irvine.com>
Date: Wed, 10 Dec 2003 11:41:12 -0800
Subject: Happy Birthday Ada
Newsgroups: comp.lang.ada

Augusta Ada Byron was born on December 10, 1815.

MIL-STD-1815 Ada was born on December 10, 1980.

p.s. some fun Ada art intended for viewing with mono-spaced font:

```
package       i
           is
         type xo
        is  range
       (%**%(1,2:0:     *
      8:1:))..""*"(3
,    (7:3:+0)/9#1#-1)
         ;end;

procedure    p
          is
       type rec
      (d: natural)
     is null record;
    protected type pt
  (e:integer) is private
 f:rec(e) ;end; protected
body pt is end; begin null
         ;end;

        generic
      package abc
     is    generic
   with      package
  de               is
 new               abc
   (      <>       )
   ;       procedure
   l                ;
 end                ;
```

From: Ludovic Brenta <ludovic.brenta@insalien.org>
Date: 10 Dec 2003 22:30:49 +0100
Subject: Re: Happy Birthday Ada
Newsgroups: comp.lang.ada

Cool, the first ever obfuscated Ada program! [...]

The History of the Ada Language

From: Jerry Petrey <jdpetrey@raytheon.com>
Date: Wed, 28 Jan 2004 15:55:51 -0700
Organization: Raytheon Company
Subject: Re: the history of Ada
Newsgroups: comp.lang.ada

> where can I find some information about the history of Ada ? I mean the programming language, not Ada Byron King.

Here are just a few sources:

http://www.cs.fit.edu/~ryan/ada/ada-hist.html

http://www.adahome.com/History/

http://www.learnada.com/history.htm

http://archive.adaic.com/pol-hist/

From: Preben Randhol <randhol+valid_for_reply_from_news@pvv.org>
Date: Wed, 28 Jan 2004 22:57:53
Organization: PVV
Subject: Re: the history of Ada
Newsgroups: comp.lang.ada

Some links

http://archive.adaic.com/ under: docs/flyers/history.html

http://unicoi.kennesaw.edu/ under: ase/ase02_01/docs/pol_hist/history/histada.txt

http://www.learnada.com/history.htm

Packed Data

From: David C. Hoos <david.c.hoos.sr@ada95.com>
Date: Fri, 13 Feb 2004 15:06:53 -0600
To: GNAT Discussion List <gnatlist@lyris.seas.gwu.edu>
Subject: Re: unchecked conversion and pragma pack

>>> Is there a way to determine which bytes that are added by the compiler when applying an unchecked conversion from a record to an array of bytes?

>> Try using -gnatR to dump the representation for all records and various other types. It's quite handy for determining this sort of thing. Gnat also has very complete representation specification support, so you probably can make a record layout that matches your legacy system. See the GNAT Reference Manual.

> Hello Chris! My mistake, I was not clear enough. What I'm after is to determine which bytes the compiler added for padding _at runtime_. I want to figure out which bytes I should skip, in my code. In my example, I had a Byte_Array of size 22, which of 2 bytes were padding bytes. I want to pass all but these 2 bytes into another byte_array of size 20, for later treatment. This array would then correspond what the first array would look like without padding bytes. So I guess I'm wondering if there is an algorithm to use, stating 'a record of 2 integers and a string(10) will always be padded with 2 bytes after the second integer' or something like, 'All byte(22) you find are padding bytes'

>> You ought to be able to use shared memory to pass packed or unpacked records between programs made with the same compiler. You can use S'Size to check subtype size in the code.

> Yes I can, and I do, but I'm as you say restricted to the same machine, compiled with the same version of the compiler.

>> Different compilers, alignment requirements and possibly byte-order issues will bite you in network IPC.

> That is why I'd like to use soap.

>> This is one of the reasons so many internet protocols use plain ASCII text. Encoding all of your IPC into XML

ought to be safe, but don't expect high message rates. There are network safe encoding like the XDR library, but I am not aware of Ada bindings or implementations of them.

> I'm thinking of using the soap implementation in Ada Web Server (AWS)

Since Ada95 (and specifically gnat) supports distributed programming across heterogeneous machines (Annex E), a solution to your problem already exists.

The Glade version of the System.Stream_Attributes package (file s-stratt.adb) contains an endianness-independent implementation. If you replace the standard gnat version with the version which comes with glade, and recompile the Gnat runtime, you can then write out your data structures on any platform, and read them on any platform.

Multitasking on Multiprocessors

From: Anders Gidenstam <anders@gidenstam.org>
Date: Wed, 12 Nov 2003 18:24:16
Subject: RE: x86, 586,686 SMP etc.
To: GNAT Discussion List <gnatlist@lyris.seas.gwu.edu>

> Has anybody got a success story with getting gnat to compile code containing tasks that map onto Linux threads, thus managing to use 2 or 4 processors on an SMP board, e.g. dual Xeons or Athlons? I'd just like to know that it is possible.

I have successfully run multithreaded Ada programs compiled with GNAT 3.14p and 3.15p on (among others) a 6 processor SunFire 880 running Solaris and on a 2-processor SPARCstation 10 running Linux (kernel 2.2.19).

I can't see why there should be any problem on SMP x86 machines. (Except that the pthreads library on Linux is changing now and that might require adjustments in the runtime, so a bleeding edge Linux kernel and an old compiler might not work well together.)

Also be aware that the memory consistency provided by a modern SMP box might not be what you expect, so memory writes performed on one processor may not become visible to the other in the order they were made. (This is certainly true for Suns but I don't know what x86 boxes do.)

However, this is usually only significant if you implement your own low-level synchronization routines (e.g. non-blocking synchronization). If you use the normal high-level features like protected objects etc there should be no problems.

From: Simon Wright <simon@pushface.org>
Date: Wed, 12 Nov 2003 21:48:46 GMT
Subject: Re: x86, 586,686 SMP etc.
To: GNAT Discussion List <gnatlist@lyris.seas.gwu.edu>

> Has anybody got a success story with getting GNAT to compile code containing tasks that map onto Linux threads, thus managing to use 2 or 4 processors on an SMP board, e.g. dual Xeons or Athlons?

This machine is a dual Celeron (Abit mobo), Mandrake 8.2, kernel 2.4.18-6mdksmp: GNAT tasking has used both processors with no trouble, certainly with 3.14p, 3.15p, 3.16a1 (not sure whether I've tried 5.01a yet).

Instructions to the ARG for Preparation of the Amendment to ISO/IEC 8652

ISO/IEC JTC1/SC22/WG9, Document N412, 10 October 2002

The ARG is instructed to prepare a working draft of an amendment to ISO/IEC 8652. The main purpose of the Amendment is to address identified problems in Ada that are interfering with Ada's usage or adoption, especially in its major applications areas (such as high-reliability, long-lived real-time and/or embedded applications and very large complex systems). The resulting language changes may range from relatively minor, to more substantial.

Examples of worthwhile changes are:

- inclusion of the Ravenscar profile;
- inclusion of a solution to the problem of mutually dependent types across packages.

The ARG is requested to pay particular attention to the following two categories of improvements:

(A) Improvements that will maintain or improve Ada's advantages, especially in those user domains where safety and criticality are prime concerns;

(B) Improvements that will remedy shortcomings in Ada.

Improvements in the real-time features are an example of (A) and should be considered a high priority.

Improvements in the high-integrity features are an example of (A) and should be considered a high priority. Features that increase static error detection are an example of (A) and should be considered a priority, but less important than the two listed above.

Improvements in the facilities for interfacing to other languages are an example of (A) and should be considered.

Improvements in the object-oriented features—specifically, adding a Java-like interfaces feature and improved interfacing to other OO languages—are an example of (B) and should be considered.

In selecting features for inclusion in the amendment, the ARG should consider the following factors:

- Implementability (vendors concerns). Can the proposed feature be implemented at reasonable cost?
- Need (users concerns). Does the proposed feature fulfill an actual user need?
- Language stability (users concerns). 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 is permitted to compromise backwards compatibility when the impact on users is judged to be acceptable.

The use of secondary standards should be minimized; secondary standards should be proposed only when they would include material so important as to require standardization but so voluminous as to preclude inclusion in the Ada language standard. In particular, material similar to the current ISO/IEC 13813, Generic Packages of Real and Complex Vector and Matrix Type Declarations and Basic Operations for Ada, should be incorporated into the language standard.

WG9 targets the following schedule for the development of the amendment:

- Dec 2002: Presentation at SIGAda, providing for discussion groups and feedback.
- Jun 2003: Similar presentation at Ada-Europe
- Sep 2003: Receipt of the final AIs from groups other than WG9 or delegated bodies
- Sep 2003: Presentation at IRTAW
- Autumn 2003: Presentation at SIGAda
- Dec 2003: Receipt of the final AIs from WG9 or delegated bodies
- Jun 2004: WG9 approval of the scope of amendment (perhaps by approving AIs, perhaps by reviewing draft amendment)
- Informal circulation of draft, receipt of comments and preparation of final text
- Spring 2005: Completion of proposed text of amendment to be contributed to WG9
- Mid 2005: WG9 email ballot
- 3Q 2005: SC22 FPDAM ballot
- Late 2005: JTC1 FDAM ballot

Comments on: "Instructions to the ARG for Preparation of the Amendment to ISO/IEC 8652"

James W. Moore

The Mitre Corporation, James.W.Moore@ieee.org, ISO/IEC JTC1/SC22/WG9 Convener, Document N423, 20 December 2002

Background

- The ARG (Ada Rapporteur Group) performs Ada language maintenance.
- They make recommendations for formal standardization to WG9 for national body voting.
- The ARG has been assigned the responsibility to draft the language amendment.
- In October 2002, WG9 prepared instructions to the ARG governing this work: "N412 Instructions to the Ada Rapporteur Group from SC22/WG9 for Preparation of the Amendment to ISO/IEC 8652, 10 October 2002"
- This presentation reproduces those instructions and provides my comments.

Purpose

"The ARG is instructed to prepare a working draft of an amendment to ISO/IEC 8652. The main purpose of the Amendment is to address identified problems in Ada that are interfering with Ada's usage or adoption, especially in its major applications areas (such as high-reliability, long-lived realtime and/or embedded applications and very large complex systems). The resulting language changes may range from relatively minor, to more substantial."

- The purpose of amendment is to address identified problems. WG9 rejected wording calling for language update and support of new paradigms.
- The phrase "usage or adoption" suggests appeal to both current and prospective users.
- Ada's "major application areas" are identified.
- Substantial language changes are permitted. This wording steers a middle course between requiring substantial change and prohibiting substantial change. This presentation quotes the complete text of the instructions. In some cases, I offer my comments on the intent or significance of the instructions.

Two Specific Improvements

"Examples of worthwhile changes are:

– inclusion of the Ravenscar profile;

– inclusion of a solution to the problem of mutually dependent types across packages."

WG9 makes two specific requests of the Amendment:

– Ravenscar Profile

– Solving problem of mutually dependent types

Two Categories of Improvement

"The ARG is requested to pay particular attention to the following two categories of improvements:

(A) Improvements that will maintain or improve Ada's advantages, especially in those user domains where safety and criticality are prime concerns;

(B) Improvements that will remedy shortcomings in Ada."

Amendment should build on Ada's advantages, particularly for safety and criticality.

Amendment should remedy shortcomings. WG9 removed the words "with respect to other languages" suggesting that we should not focus on feature-by-feature match-up with other languages.

Suggested Prioritization

"Improvements in the real-time features are an example of (A) and should be considered a high priority. Improvements in the high-integrity features are an example of (A) and should be considered a high priority. Features that increase static error detection are an example of (A) and should be considered a priority, but less important than the two listed above. Improvements in the facilities for interfacing to other languages are an example of (A) and should be considered. Improvements in the object-oriented features—specifically, adding a Java-like interfaces feature and improved interfacing to other OO languages—are an example of (B) and should be considered."

(A) Build on Ada's advantages, particularly for safety and criticality

– Real-time features

– High-integrity features

– Static error detection

– Interfacing to other languages

(B) Remedy shortcomings

– Object-oriented features—specifically, adding a Java-like interfaces feature and improved interfacing to other OO languages

The instructions create three priority levels:

– High Priority
  • Real-time features
  • High-integrity features
– A priority but less important
  • Increase static error detection
– Should be considered
  • Interfacing to other languages
  • Object-oriented features—specifically, adding a Java-like interfaces feature and improved interfacing to other OO languages.

This list is notable, not only for the prioritization, but also for what is missing. WG9 considered adding "design by contract features" to the list but decided not to add it. No other categories of features were considered.

Considerations in Selection

"In selecting features for inclusion in the amendment, the ARG should consider the following factors:

– Implementability (vendors concerns). Can the proposed feature be implemented at reasonable cost?
– Need (users concerns). Does the proposed feature fulfill an actual user need?
– Language stability (users concerns). 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?"

"Uniqueness and innovation" was considered as a criterion, but was not included.

Backwards Compatibility

"In order to produce a technically superior result, it is permitted to compromise backwards compatibility when the impact on users is judged to be acceptable."

- Compromise of compatibility may be considered.
- It was difficult to reach agreement on wording here. I interpret this instruction as saying that the Amendment is permitted to be less strict than the Ada 95 revision in maintaining backward compatibility.
- The voting on this section was close, suggesting that "acceptable impact" may be closely judged.

Secondary Standards

"The use of secondary standards should be minimized; secondary standards should be proposed only when they would include material so important as to require standardization but so voluminous as to preclude inclusion in the Ada language standard. In particular, material similar to the current ISO/IEC 13813, Generic Packages of Real and Complex Vector and Matrix Type Declarations and Basic Operations for Ada, should be incorporated into the language standard."

- Minimize secondary standards.
- A rationale for use of secondary standards is provided.
- Move function of ISO/IEC 13813 into the language standard.

Schedule

"WG9 targets the following schedule for the development of the amendment:

- Dec 2002: Presentation at SIGAda, providing for discussion groups and feedback.
- Jun 2003: Similar presentation at Ada-Europe
- Sep 2003: Receipt of the final AIs from groups other than WG9 or delegated bodies
- Sep 2003: Presentation at IRTAW
- Autumn 2003: Presentation at SIGAda
- Dec 2003: Receipt of the final AIs from WG9 or delegated bodies
- "Jun 2004: WG9 approval of the scope of amendment (perhaps by approving AIs, perhaps by reviewing draft amendment)
- Informal circulation of draft, receipt of comments and preparation of final text
- Spring 2005: Completion of proposed text of amendment to be contributed to WG9
- Mid 2005: WG9 email ballot
- 3Q 2005: SC22 FPDAM ballot
- Late 2005: JTC1 FDAM ballot."

Results

- Most notable result is the repeated emphasis on safety and criticality as Ada's niche.
- Despite spirited discussion, WG9 approved the instructions by a unanimous vote of all nations who cast a ballot (six of them

Proposal for Defining Scope of Amendment to ISO/IEC 8652:1995

Pascal Leroy

IBM France, pascal.leroy@fr.ibm.com, ISO/IEC JTC1/SC22/WG9 ARG Rapporteur, Document N437

Introduction

Excerpts from document N412 appear below in **bold font**. My additional comments appear in regular font. I have added a few bulleted subcategories in cases where N412 only gave broad advice and a few examples.

Because we are arguing about scope, not AIs, I only give AI numbers, not version numbers or dates.

(Note by the editor: *The author has requested to advise the reader that the contents of this document as for discussion only and do not represent the position of the ARG.*)

The AIs that are not marked with a symbol have been approved by WG9, or approved by the ARG and will go to the next WG9 meetings.

The AIs that are still in the works are marked with one of the following symbols after their number:

† This AI still needs a bit more work before it can go to WG9, but its technical content is well-defined and believed to be sound (15 AIs).

‡ This AI still needs substantial work before it can go to WG9, and its technical content is still in a state of flux. The reason why it's still alive is that the ARG sees sufficient value in the ideas being proposed, and therefore wants to study them some more before making a final decision. Note that there is no firm consensus on these AIs yet, as many people want to see the AI mature before forming an opinion (7 AIs).

Examples of worthwhile changes are:

- **inclusion of the Ravenscar profile;**

AI95-00249 Ravenscar profile for high-integrity systems
AI95-00265 Partition Elaboration Policy for High-Integrity Systems
AI95-00305 New pragma and additional restriction

AI95-00249 and AI95-00305 together define the language features corresponding to what is known in the vernacular as the "Ravenscar profile". While not strictly part of Ravenscar, AI95-00265 was motivated by practical usage of the Ravenscar profile.

- **inclusion of a solution to the problem of mutually dependent types across packages.**

AI95-00217 Limited With Clauses
AI95-00230 Generalized use of anonymous access types
AI95-00326 Incomplete types

Both AI95-00217 and AI95-00326 are required to solve the problem of mutually dependent types across packages. However, in isolation, they would lead to proliferation of access types and conversions between these types. AI95-00230 addresses this second problem.

The ARG is requested to pay particular attention to the following two categories of improvements:

(A) Improvements that will maintain or improve Ada's advantages, especially in those user domains where safety and criticality are prime concerns;

- **Improvements in the real-time features are an example of (A) and should be considered a high priority.**

AI95-00297† Timing events
AI95-00307† Execution-Time Clocks
AI95-00321 Definition of dispatching policies
AI95-00327† Dynamic ceiling priorities
AI95-00353 New Restrictions identifier No_Synchronous_Control
AI95-00354† Group Execution-Time Budgets
AI95-00355‡ Priority Specific Dispatching including Round Robin
AI95-00356‡ Support for Preemption Level Locking Policy
AI95-00357‡ Support for Deadlines and Earliest Deadline First Scheduling

All these AIs come from the IRTAW, so they are supposed to reflect the needs of the run-time community.

- **Improvements in the high-integrity features are an example of (A) and should be considered a high priority.**

AI95-00266† Task termination procedure
AI95-00347 Title of Annex H

Note that the Ravenscar profile mentioned above is actually a capability that relates to the high-integrity usage of Ada.

- **Features that increase static error detection are an example of (A) and should be considered a priority, but less important than the two listed above.**

AI95-00218 Accidental overloading when overriding

AI95-00231 Access-to-constant parameters and null-excluding access subtypes
AI95-00262 Access to private units in the private part
AI95-00287 Limited Aggregates Allowed
AI95-00310 Ignore abstract nondispatching subprograms during overloading
AI95-00318‡ Returning [limited] objects without copying
AI95-00363† Eliminating access subtype problems

AI95-00218 addresses a problem that can lead to extremely severe errors in systems using OOP, and that is addressed by some other OOP languages (Eiffel, C#).

AI95-00363 eliminates a number of problems with access types which could lead to extremely severe errors.

AI95-00287 and AI95-00318 are intended to make limited types more usable: currently limited types have so many restrictions that they are hardly used at all. If users could use limited type more often, they would benefit from the associated static error detection (in particular to avoid unwanted sharing).

AI95-231 makes it possible to specify more precisely the properties of access types. AI95-262 gives more control on the visibility of entities and makes private units more usable. In both cases, additional static error detection can be obtained by using the new features.

AI95-00310 makes it possible to "undefine" operations, thereby avoiding references to (inherited) operations that don't make sense for an entity.

- **Improvements in the facilities for interfacing to other languages are an example of (A) and should be considered.**

AI95-00216 Unchecked unions -- variant records with no run-time discriminant
AI95-00248 Directory Operations
AI95-00315† Full support for IEC 559:1989
AI95-00351† Time operations
AI95-00370† Environment variables

AI95-00216 is the only AI that actually pertains to interfacing to another language (C). However the ARG felt that there was a need to be able to interface to other *computing environments* as well. The other AIs listed here all address this issue.

The following AIs add new predefined units which increase the capabilities of the Ada programming environment, and increase the portability of programs:

AI95-00296 Vector and matrix operations
AI95-00302† Container library

(B) Improvements that will remedy shortcomings in Ada.

- **Improvements in the object-oriented features—specifically, adding a Java-like interfaces feature and improved interfacing to other OO languages—are an example of (B) and should be considered.**

AI95-00251 Abstract Interfaces to provide multiple inheritance
AI95-00252 Object.Operation notation
AI95-00345‡ Protected and task interfaces
AI95-00348 Null procedures

Support for Java-like interfaces is provided by AI95-00251, AI95-00345 and AI95-00348. AI95-00252 adds support for a prefix notation which is common in other languages OOP and sometimes more convenient than the traditional Ada notation.

The following AIs enhance the portability of Ada programs:

AI95-00224 pragma Unsuppress
AI95-00257 Restrictions for implementation-defined entities
AI95-00260 How to control the tag representation in a stream
AI95-00270 Stream item size control
AI95-00286 Assert pragma
AI95-00368† Restrictions for obsolescent features

The following AIs improve composability of the elements of the language, making it easier to build libraries of components and reuse them:

AI95-00254 Anonymous access to subprogram types
AI95-00317 Partial Parameter Lists for Formal Packages
AI95-00344‡ Allow nested type extensions
AI95-00359‡ Deferring Freezing of a Generic Instantiation

The following AIs lift somewhat arbitrary restrictions, or add new capabilities that improve the usability and readability of the language:

AI95-00301 Operations on language-defined string types
AI95-00328 Preinstantiations of Complex_IO
AI95-00340 Mod attribute
AI95-00361 Raise with message
AI95-00362† Some predefined packages should be recategorized
AI95-00366† More liberal rule for Pure units

The following AIs provide mechanisms for improving the efficiency of user programs or of implementations:

AI95-00267 Fast float-to-integer conversions
AI95-00273 Use of PCS should not be normative
AI95-00329 pragma No_Return -- procedures that never return

The following AIs improve the compatibility between the language as revised by the Amendment and Ada 83 and Ada 95:

AI95-00284 Nonreserved keywords
AI95-00364† Fixed-point multiply/divide

The following AI improves the support of internationalization and localization:

AI95-00285† Support for 16-bit and 32-bit.

Fixing Software Before It Breaks: Using Static Analysis to Help Solve the Software Quality Quagmire

S Tucker Taft

SofCheck, Inc. 11 Cypress Drive, Burlington, MA 01803 USA.; Tel:+1 781 750 8068; email: stt@sofcheck.com

Abstract

There is growing realization that something must be done to improve the quality and security of software, but the costs associated with exhaustive testing approaches are economically prohibitive for all but the most critical systems. But progress is being made to develop scalable and accurate static analysis technologies to help find a way out of the current software quality quagmire. Static analysis has the potential to automate the process of line-by-line, full path inspection of source code for defects, while also providing component-by-component characterization of the software system in terms of inputs, outputs, and effects, thereby fostering better understanding of legacy software and enabling more informed software evolution and reuse strategies. In the real-time arena, static analysis has been used to detect race conditions and possible deadlocks, and help with schedulability determination. When coupled with this more general capability for automated defect detection and component characterization, new opportunities are created. In particular, by implicitly adding to the source code metering variables to count quantities such as stack depth, loop counts, dynamic storage allocation, lock frequency and duration, etc., the same static analysis technology can summarize these additional kinds of important real-time characteristics of the code.

Keywords: Static Analysis, Software Quality.

1 Introduction to Static Analysis

There is growing realization that something must be done to improve the quality and security of software, but the costs associated with traditional exhaustive testing approaches are economically prohibitive for all but the most critical systems. However, an alternative approach, based on static analysis, has begun to emerge in the past few years. Although the research underlying static analysis began 20 to 30 years ago [5] only recently has sufficient computing power been available on the developer's desktop to support truly practical automated testing tools using static analysis.

The software engineering community has been talking about proving programs correct for many years, but only relatively small examples of full program correctness proofs have ever been accomplished [15]. Even automated proof systems have not been able to handle truly large, complex systems, due to the nearly intractable challenges associated with large formal correctness proofs. But an even bigger impediment to formal correctness proofs has been the difficulty of formally specifying the intended result of a large software system to the level of detail required for formal proof [3].

More recently, an alternative to formal correctness proofs has emerged which can nevertheless provide many of the same advantages, such as full coverage of all possible input conditions and all possible program control paths. This alternative is based not so much on proving programs correct, as proving that they do not perform certain clearly incorrect or meaningless operations [7]. For example, although it may be difficult to specify exactly what a large system should do, it is generally agreed a system should not index beyond the bounds of an array, dereference a null pointer, read uninitialized memory, or update shared data without proper synchronization.

A tool that can prove that a given software system performs none of some set of well-defined *incorrect* operations, while taking all possible inputs and all possible control paths into account, could dramatically enhance the testing process. If such a tool could also identify the exact line within a program where the *incorrect* action is performed, it could also dramatically enhance the debugging process. Finally, if the tool could identify exactly which possible input values might cause the incorrect operations to be performed, and which values can be processed without any such incorrect operations occurring, it could provide significant help in determining in what context a given program or software component could be appropriately reused.

All of the above capabilities are now becoming possible through the new brand of static analysis tools [2]. One way to think of this new brand of testing tool is as a *super* type checker. The type checker in a compiler for a language with a static type model is effectively operating as an automated test tool, identifying places within the program that violate certain safety rules. In languages like Ada and Java, the *static* type checking is augmented by *dynamic* run-time checks as necessary to identify violations of safety rules that are not easily checked at compile-time. A static analysis software *checker* is simply pushing the static checking process further, with more complex safety rules,

or with more language rules checked statically that are normally not checked until run-time. This is accomplished by using more sophisticated algorithms than those used in most compiler type checkers, in particular using analysis algorithms that enable the checker to perform a flow-sensitive prediction of the possible values of run-time variables.

Currently there are a number of distinct algorithmic approaches to doing this kind of sophisticated flow-sensitive analysis. Formal proof techniques are used in some tools, though at the current state of the art, formal proof tools generally can only be semi-automated, requiring some amount of human intervention to suggest loop invariants or other intermediate assertions needed to allow the proof to complete [13]. A second approach that has been garnering a fair amount of interest is derived from hardware verification efforts, namely *model checking*. Model checking is a technique for checking whether a finite state model of a system violates one or more desired invariants. The model checking approach generally requires the software to be recast into a representation with a smaller number of states, such as a *Boolean* program representation [1]. A third approach is based on what has been called Abstract Interpretation [6]. This is basically a formalization and generalization of the kinds of iterative control and data flow analysis algorithms familiar from compiler optimizers. The results of Abstract Interpretation depend heavily on the kinds of approximations which must be done to get the iterative flow analysis algorithms to converge [5].

2 Challenges for Static Analysis Tools

For all of these static analysis approaches, there are fundamentally four challenges: scalability, precision, modularity, and understandability. Scalability, in the sense of being able to analyze large, complex software systems, in a reasonable amount of time, is probably the critical factor now in terms of whether such tools can be used in a real production environment. This remains a challenge for all of the approaches. Careful engineering is clearly required here, and different tools draw on different bodies of research into efficient algorithm and data structure design. For example, in model checking, the so-called Binary Decision Diagrams (BDDs) have emerged as a favorite structure for handling large models efficiently [17].

Precision is also a critical challenge for static analysis tools. By analogy to a medical diagnostic test, there are the problems of false negatives and false positives. False negatives are when a real problem goes undetected. False positives are when non-problems are identified as being problems (also called *false alarms*). Having either too many false negatives or too many false positives can make a diagnostic test have little value.

For the static analysis of programs, accomplishing perfect precision is essentially equivalent to solving the halting problem, and is not something anyone is expecting any time soon. On the other hand, for any particular program or type of program, it is possible to characterize a tool in terms of the likelihood of false negatives or false positives per million lines of code analyzed. In the case of false negatives, we may be able to claim that this number is zero, at least for a certain category of errors (such as use of uninitialized variables). That is, all possible uses of uninitialized variables are identified. Alternatively, it is possible to claim that the number of false positives is zero, for a certain category of errors. For example, a tool might never identify a line of code as using an uninitialized variable, unless it can prove that it will do so. The hard part is claiming both zero false negatives and zero false positives.

Some tools try to minimize both false negatives and false positives, without any guarantees either way, thereby maximizing the number of messages that correspond to real bugs, but still possibly missing some [12]. Other tools, particularly those aimed at the high integrity marketplace, are designed to detect all bugs of a certain category, but may also end up with more false positives (*crying wolf*) [2]. The relatively simple checkers that are sometimes included with compilers tend to try to minimize false positives, preferring to be silent if there is a chance the code is correct (*pollyanna/rose-colored glasses).*

Modularity is a third challenge for static analysis tools. Programs are built out of components, and it is preferable if components or groups of components can be analyzed even before the entire program is complete. Often groups of components are built as reusable libraries, so there is also a desire to check the correctness of the components, even in the absence of particular *client* programs. Finally, changes are generally made to a small number of components at a time. It is desirable that an incremental change need only require an incremental analysis, rather than a complete reanalysis of the entire software system.

Unfortunately, a number of the classic algorithms used in static analysis tend to be oriented to dealing with an entire system at a time, rather than on a component-by-component basis. Model checking in particular tends to be oriented toward checking an entire finite state representation at one time [16]. Similarly, some of the global *points-to* databases used to represent potential aliasing between variables used in a program are not well suited to incremental analysis [14].

The fourth challenge is the understandability of tool output. A testing tool is of little use if the error or warning messages it provides are inscrutable to the average programmer. Unfortunately, it is not always easy to explain a problem, even if a tool is quite confident that the problem exists [9]. Modularity is related to this, because a tool based on an approach that tends to work globally rather than component-by-component may report errors in a way that only a programmer who is very familiar with the entire global structure of the program could understand.

Another understandability issue can arise if the program has been transformed to some kind of canonical representation as part of the analysis. This transformation may need to be at least partially reversed to produce

programmer-meaningful messages. In some ways this is similar to the problem faced by interactive run-time debuggers, which need to translate the binary state of the machine back into a programmer-meaningful description of the state.

3 Static Analysis for Real-Time

In the real-time arena, static analysis has been used in the past to detect race conditions and possible deadlocks [11], and to help with schedulability determination [8]. The tools to do these kinds of analyses have tended to be special purpose tools, devoted to one particular concern. However, as more general purpose static analysis tools become available, which can automatically detect other kinds of errors and provide automatic characterizations of components, new opportunities are created.

Most of the new breed of static analysis tools are oriented toward checking for errors, or validating certain assertions. Model checking, in particular, is well adapted to validating assertions relevant to a multi-threaded environment [4], due to model checking's heritage in hardware validation (which is inherently highly parallel). Similarly, theorem proving can be used to verify desirable properties of a real-time system, such as no race conditions, no deadlocks, etc [11].

In addition to checking-oriented analysis, real-time systems could also benefit from analyses that can characterize the properties of a system numerically, rather than just validating the truth or falsehood of an assertion. For example, it would be useful to know the maximum number of times a given loop might iterate, or the maximum depth of subprogram call nesting, or the total amount of storage allocated dynamically during system initialization. Unfortunately, most of the new breed of static analysis tools are not oriented toward producing these kinds of numerical results.

By extending the basic Abstract Interpretation approach to static analysis, to include characterization of the range of possible outputs of each component, some of these more numerically-oriented properties can be included in the results produced by the static analysis tool. In particular, our company has been developing some new techniques for efficiently and precisely determining the set of possible values of every object modified by a given software component, whether directly, or indirectly via other components. The net result is that we can summarize the effects of each software component quite completely, in a way that could be particularly useful to real-time system development.

The summary can be made more useful by having the tool (or some pre-processing step) automatically add implicit *metering* variables into the code. The idea is to add to each subprogram an extra **out** (or **access**) record parameter which holds various counts reflecting the resource utilization of the subprogram. For example, this *resource-utilization record* could include a count of the number of statements executed (or perhaps an estimate of the cycle counts) by the subprogram (directly or indirectly), the maximum stack depth, the total amount of dynamically allocated storage, the number of locks or unlocks, etc. The tool can then utilize its normal value and range propagation mechanisms to propagate an upper bound on these counts associated with each subprogram, in some cases as a function of the inputs.

4 Conclusion

Static analysis has the potential to automate the process of line-by-line, full path inspection of source code for defects, while also providing component-by-component characterization of the software system in terms of inputs, outputs, and effects. This information in turn can foster better understanding of legacy software and enable more informed software evolution and reuse strategies.

The benefits of static analysis can also be carried over to the real-time arena, but in this case, it becomes even more important that the tool has the ability not simply to check for certain desired properties, but also to derive certain important numerical characteristics of the code, such as maximum nesting depth, or maximum statement count, as a function of the inputs. Our belief is that this kind of static analysis tool can be a big help in the process of turning software development and validation, in particular for real-time systems, from a black art into a true science.

References

[1] Thomas Ball and Sriram K. Rajamani. (2000), *Bebop: A Symbolic Model Checker for Boolean Programs*, in *Proc. of the 7th SPIN Workshop* (SPIN 2000), Stanford Univ., CA, pp. 113-130.

[2] B. Blanchet *et al* (2003), *A static analyzer for large safety-critical software*, in *Proc. Of the ACM SIGPLAN 2003 conference on Programming language design and implementation* (PLDI 2003), San Diego, CA, pp. 196-207.

[3] Manfred Broy (1996), *Experiences with Software Specification and Verification Using LP, The Larch Proof Assistant,* in *Formal Methods in System Design: An International Journal*, 8(3), Kluwer Academic Publishers, May 1996, pp. 221-272.

[4] T. Bultan, R. Garber, W. Pugh (1999), *Model-checking concurrent systems with unbounded integer variables: symbolic representations, approximations, and experimental results,* in *ACM Transactions on Programming Languages and Systems,* 21(3), July 1999, pp. 747-780.

[5] P. Cousot (1981), *Semantic foundations of program analysis,* in *Program Flow Analysis: Theory and Applications*, S.S. Muchnick and N.D. Jones, eds., chapter 10, Prentice-Hall, Inc., pp. 303-342.

[6] P. Cousot (1996), *Abstract Interpretation,* in *ACM Computing Surveys*, 28(2), June 1996, pp. 324-328.

[7] Economist Magazine (2003), *Building a better bug trap,* in *The Economist*, June 19, 2003 issue.

[8] V. C. Gerogiannis and M. A. Tsoukarellas (1995), *SAT – A schedulability analysis tool for real-time*

applications, in *Proc. Of 7th Euromicro Workshop on Real-Time Systems* (EUROMICRO-RTS'95), Odense, Denmark, March 1995, pp. 155-163.

[9] Alex Groce and Willem Visser (2003), *What Went Wrong: Explaining Counterexamples,* in *10th Intl. SPIN Workshop on Model Checking of Software,* LNCS 2648, Springer Verlag, May 2003, pp. 121-135.

[10] G. M. Karam and R. J. Buhr (1990), *Starvation and critical race analyzers for Ada,* in *IEEE Trans. Software Eng.*, 16(8), August 1990, pp. 829-843.

[11] G. M. Karam and R. J. Buhr (1991), *Temporal logic-based deadlock analysis for Ada,* in *IEEE Trans. Software Eng.,* 17(10), October 1991, pp. 1109-1125.

[12] V. Benjamn Livshits and Monica S. Lam (2003), *Tracking Pointers with Path and Context Sensitivity for Bug Detection in C Programs,* in *Proc. Of 9th European software engineering conference* (ESEC/FSE'03), Helsinki, Finland, Sep. 2003, pp. 317-326.

[13] Monroe Newborn and Monty Newborn (2001), *Automated Theorem Proving: Theory and Practice,* Springer Verlag, 2001.

[14] Marc Shaprio and Susan Horwitz (1997), *Fast and accurate flow-insensitive points-to analysis,* in *Proc. Of 24th ACM SIGPLAN-SIGACT symposium on Principles of Programming Languages* (POPL '97), Paris France, Jan. 1997, pp. 1-14.

[15] M. G. Staskauskas (1996), *An experience in the formal verification of industrial software,* in *Communications of the ACM,* 39(12es), December 1996, Article 256.

[16] M. V. Vardi (1995), *On the complexity of modular model checking,* in *Proc. Of the 10th Annual IEEE Symposium on Logic in Computer Science,* San Diego, CA, June 1995, pp. 101-111.

[17] B. Yang (1998), *A Performance Study of BDD-Based Model Checking,* in *Proc. Of the 2nd Intl. Conf. On Formal Methods in Computer-Aided Design,* (FMCAT'98), Palo Alto, CA, Nov. 1998, pp. 255-289.

Ada-Europe 2003 Sponsors