ADA
USER
JOURNAL

Volume 22
Number 2
June 2001

Contents

Editorial Policy for *Ada User Journal*

Publication

Ada User Journal – The Journal for the international Ada Community – is published jointly by Ada Language UK Ltd and 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 Language UK Ltd and Ada-Europe an unlimited licence 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Unless the authors state explicitly otherwise, submission of an article is taken to imply that it represents original, unpublished work, not under consideration for publication elsewhere.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 Language UK Ltd, Ada-Europe or their 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. Electronic submission is preferred – typed manuscripts will only be accepted by the Editor 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 at:

www.adauk.org.uk

There is no limitation on the length of papers, though a paper longer than 10,000 words would be regarded as exceptional.

Editorial

This is the first opportunity that the Ada User Journal has had to thank John Barnes for the nine years spent as President of Ada-Europe – John retired at the recent Ada-Europe General Assembly. I am sure that everyone would agree that John's presidency has been valuable for the community. Also, we wish Erhard Plödereder well as the new President. A full report of the Ada-Europe General Assembly is contained later in the Journal.

The technical article in this issue contains a summary of the 2000 Ada market report, commissioned by Ada Language UK Ltd. This makes interesting reading, particularly in that Ada remains well established in its niche critical systems market. The article concludes with the observation that developing safe critical systems remains an expensive exercise, with Ada regarded as the only viable candidate language. There is no evidence that alternative languages would enable the system to be developed faster or cheaper, or the resultant system to have the same level of safety.

A bumper news section has many interesting items. Those who are actively involved in language choice for projects should read the Ada success stories in the "Ada Inside" part of the news. Also the "Ada in Context" news section contains an excellent contribution by Mark Lundquist within the "Advantages & Disadvantages of Ada Compared to Other Languages" thread. This provides a good, fair and honest comparison of Ada with other languages.

Finally, I draw your attention to four calls for participation detailed in the Forthcoming Events section:

- ACM SIGAda Annual International Conference – SIGAda 2001
- Symposium on Reliable Object-Oriented Programming (SROOP) 2001
- 7th International Conference on Reliable Software Technologies – Ada-Europe 2002
- 11th International Real-Time Ada Workshop 2002 (IRTAW 11)

Although many readers will have just returned from the successful Ada-Europe conference, it is time to start thinking about contributions to other Ada conferences.

Neil Audsley
York
June 2001
Email: Neil.Audsley@cs.york.ac.uk

News

Dirk Craeynest (ed)

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

Contents

Ada-related Events

Ada-Europe'2001 Conference Pictures

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

The 6th International Conference on Reliable Software Technologies - Ada-Europe'2001 - was held in Leuven, Belgium, during the week of May 14-18, 2001. Pictures taken at the conference and the social events are available on the conference website at http://www.ada-europe.org/conference2001.html

Dirk Craeynest, Ada-Europe'2001 Program Co-Chair

Ada and Education

Comp.lang.ada Newsgroup Sees Increased Interest

From: "Ehud Lamm" <mslamm@mscc.huji.ac.il>
Date: Fri, 9 Feb 2001 23:21:06 +0200
Organization: The Hebrew University of Jerusalem
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

> If the profs who teach Ada would refer their students here [= comp.lang.ada (CLA)] and to some of the more useful web sites like Adapower, this would help because they would get exposed to these sorts of uses of Ada. We can always do our part by politely helping out the students when we can.

I do this all the time. I send them to find reusable stuff on AdaPower (some of it my contributios, which I for this very purpose post on AdaPower and not on our local site). I also tell them about interesting comp.lang.ada threads. Those that have genuine interests in learning, try to use these resources. Alas, they are always the minority.

Some recent threads here started by my students. One was the "Visual Ada" thread. What's interesting is that the semester just ended, so the student is asking about an Ada IDE after the course ended. Does this mean he likes to use Ada for other things? I don't know.

From: "Jeffrey D. Cherry" <jdcherry@utech.net>
Date: Fri, 09 Feb 2001 13:25:09 -0800
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

As a part-time instructor, I teach Ada (and other languages) at the local community college for both CS1 and CS2 courses. My syllabus for each of the Ada courses strongly recommends that students check out Ada resources on the Internet, starting with AdaPower and CLA. The one thing I restrict students from doing is asking for help on their homework in CLA. That's my job. The other computer science instructor has a similar policy and even assigns homework requiring students to summarize a recent Internet article, discussion thread, etc.

Although students loathe to do any more reading than the minimum, occasionally, there is the motivated student that asks a question about some discussion on CLA. I've also found certain CLA discussions to be quite interesting and posed the original query to my class in order to generate a discussion of "real-world" problems. I have demonstrated GtkAda applications to show students that GUI applications can be built with Ada. I have also shown how a Java application can be built using Ada rather than Java (using JGNAT of course). Although my Ada students were happy to hear that they don't have to learn Java to create Java apps, my Java students were rather disappointed.

I suspect that other instructors, throughout the world, have similar practices that encourage students to explore the resources of the Internet and participate in discussion groups. If you fail to notice a large contingent of student participation on CLA, it's not necessarily due to the lack of encouragement by their instructors.

Personally, I believe that it is more important to teach students good software engineering principles and practices than to sing the praises of any one particular programming language. I try to instill in my students that a programming language is merely a tool used to express their design in a form that a stupid machine can understand. At the end of each of my CS1 courses, I always devote a lecture hour to persuading students to learn another programming language, and then another, and another, ... I do this because a good engineer will learn about all the available tools and then apply proper engineering criteria to select the right tool for the job at hand. Ada, by design, is one of the best tools for expressing a software design in the vast majority of real-world applications.

Perhaps the increased interest in Ada is due to all the diligent teachers who have taught their students well. Perhaps those students have now graduated and are choosing Ada after performing a tradeoff analysis with other programming languages. Perhaps these well educated graduates are dismissing the marketing group's recommendation for a programming language because it's based on personal bias, advertising hype, perceived popularity, or the misperception that a certain programming language will somehow guarantee an increased market share. Perhaps these graduates are negating management pressure to use one language by showing that it is more cost effective to select a programming language based on technical merits of the language in light of the specific problem and accounting for all phases of the resulting product's useful life. [...]

Jeffrey D. Cherry, Senior IV&V Analyst, Logicon Operations and Services, Logicon Inc., a Northrop Grumman company

From: Stephen Leake <stephen.a.leake.1@gsfc.nasa.gov>
Date: 12 Feb 2001 12:43:40 -0500
Organization: NASA Goddard Space Flight Center
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

As an Ada enthusiast, but more importantly as a software engineer who would like to work with well trained software engineers, I thank you for your efforts. Keep up the good work; the world will be a better place for it!

Stephen Leake, NASA Goddard Space Flight Center.

Students Prefer Ada Over Java

From: John McCormick <mccormic@cs.uni.edu>
Date: Fri, 2 Mar 2001 12:37:35 -0600
Subject: Students prefer Ada over Java
To: team-ada@acm.org

Currently at the University of Northern Iowa, students learn Ada in their first and second programming courses. The third course they take is "Object-Oriented Programming with Java". And the fourth course is Algorithms. In the algorithms course, students may use any language they want. Both the OO/Java course and Algorithms course are taught by anti-Ada faculty. This semester I learned through a student taking the Algorithms course that no student is using Java. A few are using C or C++ , one is using PERL , and the rest are using Ada. I find fact that none are using Java particularly enlightening as Java is the last language with which they had significant programming work.

Just a lone but intriguing data point.

John W. McCormick, mccormick@cs.uni.edu, john.mccormick@acm.org Computer Science Department, University of Northern Iowa, Cedar Falls, IA 50614-0507, voice (319) 273-2618, fax (319) 273-7123 http://www.cs.uni.edu/~mccormic/

From: Ben Brosgol <brosgol@world.std.com>
Date: Mon, 5 Mar 2001 15:15:37 -0500
Subject: Re: Students prefer Ada over Java
To: team-ada@acm.org

Interesting data, John- thanks for sending the information.

But it looks like the message could also have been titled as "Students prefer Ada over C and C++". In an Algorithms course (which presumably deals with data structures also :-) many of Java's deficiencies would become apparent fairly quickly: no enumeration types, all composite objects go on the heap, no generics, weakly-typed scalars, ... What is interesting is that students preferred Ada over the more commercially-popular C and C++. A university where students make choices based on fitness to purpose versus applicability in the immediate job market? Is there something in the water in northern Iowa that induces such behavior? :-)

Ben Brosgol, Ada Core Technologies, 79 Tobey Road, Belmont, MA 02478, USA, +1-617-489-4027 (voice), +1-617-489-4009 (FAX), brosgol@gnat.com

From: John McCormick <mccormic@cs.uni.edu>
Date: Mon, 5 Mar 2001 15:52:27 -0600
Subject: Re: Students prefer Ada over Java
To: team-ada@acm.org

The only reason I didn't is that the students do not have any formal exposure to C or C++. They had courses in Ada and Java before the algorithms course.

From: Geoff Bull <gbull@acenet.com.au>
Date: Thu, 8 Mar 2001 11:32:33 +1100
Subject: Re: Students prefer Ada over Java
To: team-ada@acm.org

[...] The reason that students prefer Ada over Java for an algorithms class is almost certainly Ada's much superior abstraction capabilities. And not being forced to used objects even when you don't want them.

Ada95 Tutorials With Sample Code

From: "Marin David Condic" <marin.condic.auntie.spam@pacemicro.com>
Date: Tue, 6 Mar 2001 10:03:07 -0500
Subject: Re: Ada95 tutorials with sample code
Newsgroups: comp.lang.ada

My web page appears to be back in working order. You might want to look at some of the example code that appears on my Ada Programming page. (Start at: http://www.mcondic.com/)

I have a bunch of small Ada programs that I used in conjunction with an internal training course I taught. They make more sense within the context of the course, but they may serve well to provide some small, easily manipulated programs that assist in learning the language. Look for the "Gnat Examples" code on my Ada Programming page.

Hope this is useful...

Marin David Condic, Senior Software Engineer, Pace Micro Technology Americas, www.pacemicro.com, e-Mail: marin.condic@pacemicro.com, Web: http://www.mcondic.com/

C++ and Java in Academia - a Quote

From: "Michael Kölling" <mik@monash.edu.au>
Sent: Saturday, March 10, 2001 7:57 PM
Subject: Re: [Bluej-discuss] Java in the curriculum
To: <bluej-discuss@bluej.org>

[This message was quoted by Alan Brain <aebrain@dynamite.com.au> on the Team-Ada mailing list. -- dc]

> A bit of unpleasant news: my department is switching its first-year programming sequence for majors [from Java] to C++, effective next fall. I made all the obvious arguments, including that switching to C++ in 2001 was like buying stock in buggy-whips shortly after the Model T hit the market. [...]

Wow - what can I say... I have had to teach C++ to first years before, and I think I'd rather resign than do it again. My thoughts are with you in these trying times...

An Embedded Programming Course in a Bag?

From: "Marin David Condic" <marin.condic.auntie.spam@pacemicro.com>
Date: Wed, 21 Feb 2001 10:18:31 -0500
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

I could imagine Ada being popular in electrical engineering departments if there were a convenient and inexpensive (maybe free?) Ada environment for playing around with embedded computing. It would have to work "off the shelf" with readily available hardware so that some prof could build a class/lab around it & students could afford to play with it on their own. I am thinking of Dr. McCormick's model railroad class or the Lego robot discussed here a while ago. If either of these was packaged as "An embedded programming course in a bag" so that a prof could just pick it up and start teaching it, this might go a long way toward encouraging Ada as an educational tool as well as a practical tool for building real-world systems.

(Does anyone smell commercial possibilities here? :-)

[And from another message:]

It occurs to me that much of the embedded programming experience could be simulated in software. Obviously, you wouldn't get the "Real World" experience of dealing with actual physical entities, but, for example, actuators could be displayed on a screen and made to move much as they would in the physical world. The software interface to such simulated sensors and actuators wouldn't be quite the same thing as having to deal with actual ports, memory addresses, etc., but it might be made close enough to be a useful experience. Providing such a simulation in Ada would certainly be a lot easier to achieve than finding an appropriate embedded target & compiler port.

Question: Given that a simulation like this would lack certain important aspects of the embedded, realtime programming experience (having to somehow work with a cross-compilation environment, dealing with linkage issues, memory mapping, physical reality, etc.) might it still be useful as a teaching tool? I think a simulation in conjunction with hardware would be useful, but I'm wondering about finding a way around the problem of compiler and hardware availability?

Marin David Condic, Senior Software Engineer, Pace Micro Technology Americas, www.pacemicro.com, e-Mail: marin.condic@pacemicro.com, Web: http://www.mcondic.com/

From: Jerry Petrey <"jdpetrey <NOSPAM>"@west.raytheon.com>
Date: Wed, 21 Feb 2001 15:56:00 -0700
Organization: Raytheon Company
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

[On the idea of simulating the "embedded programming experience" in software:]

Marin, from my experience, I have found that such an approach can be very useful. Years ago, I used to teach a course in embedded programming using Forth. My class project was modeled after one from Forth, Inc. (who I had worked for in the past). They used a small traffic light with road sensors and such so that you could move a toy car over them and activate the sensors. The students would then write the Forth code to implement the assigned behavior. In my class, I needed to travel to the location and I didn't want to take this kind of hardware with me so I build a PC simulator for the traffic light and sensors and then each student would run this on his PC and write the code to control it much like he would with the real hardware. When he would touch the road on the PC screen with the mouse it would register as a car passing and his code would then set the light sequence according to the required behavior (and display them on the screen). It seemed to be quite effective. Each student had his own little self contained environment in his PC to play around in and experiment with different algorithms without the need to connect to any special hardware. Of course, at some point they need to get the real hardware experience but this is a good way to start.

Jerry Petrey, Senior Principal Systems Engineer, Raytheon Missile Systems

From: Peter Amey <pna@praxis-cs.co.uk>
Date: Thu, 22 Feb 2001 10:43:42 +0000
Organization: Praxis Critical Systems
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

We have done something like this for the SPARK course. We have a visual basic :-(on-screen emulation of a hardware device and students can drive it from their SPARK code using interface packages we provide. The link between SPARK and VB is done with David Botton's excellent COM stuff.

From: "Marin David Condic" <marin.condic.auntie.spam@pacemicro.com>
Date: Thu, 22 Feb 2001 09:27:33 -0500
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

That sounds interesting. I'm wondering what sort of students you present this to and what are the learning objectives? Do you think they learn much about the embedded aspects, or do they learn more about realtime programming?

My concern is that such a simulation would be useful for developing the high level algorithms for embedded, realtime controls, but probably won't do a good job of teaching the low level aspects of embedded programming. At the high level, embedded programming looks much like any other kind of programming - albeit, within a specialized problem domain. What I'd like to find is a good, inexpensive way of teaching the low level aspects - things like accessing different kinds of memory, interfacing to I/O devices, utilizing hardware interrupts, etc. as well as the higher level concepts of device control. You just don't get much of a feel for real embedded programming unless you've had to spend time fighting with a linker to get things located at specific places, or struggling to get bootstrap code to load your software across a comm link, or get into the "Broken Software/Broken Hardware" debate.

I think it would be helpful to Ada to have a good embedded/realtime off-the-shelf course (book, software, hardware...) available, but I've just not encountered the components that would make this possible at a practical cost. Software simulation might make an interesting project, but I'm not sure that it would illustrate enough of the important parts of the embedded world.

Now possibly, if one were to bundle a compiler with a simulation of an actual SBC with a popular processor, then you might have something there. However, that starts becoming more work than simply designing a board and retargeting a compiler... :-)

From: Peter Amey <pna@praxis-cs.co.uk>
Date: Tue, 27 Feb 2001 11:28:31 +0000
Organization: Praxis Critical Systems
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

> I'm wondering what sort of students you present this to and what are the learning objectives? Do you think they learn much about the embedded aspects, or do they learn more about realtime programming?

Our aim is to teach the design and static analysis aspects of SPARK. The low-level, real-time aspects of the project are not particularly important. The aim of the emulator is more to give students satisfaction: they write all this stuff, the SPARK Examiner tells them it is good but they didn't used to get to see it working. The emulator just gives that extra satisfaction. (Incidently, the first time the "model answer" had ever been compiled or run was when we tested the emulator - up to then it had only been analysed - it worked perfectly first time). [...]

Peter Amey, Product Manager, Praxis Critical Systems Ltd, 20, Manvers Street, Bath, BA1 1PX, Tel: +44 (0)1225 466991, Fax: +44 (0)1225 469006, http://www.praxis-cs.co.uk/

From: Cesar Rabak <csrabak@uol.com.br>
Date: Fri, 23 Feb 2001 01:58:30 -0300
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

[On an "inexpensive Ada environment for playing around with embedded computing": -- dc]

I don't know how is the situation in other parts of the world for this kind of products, but in this country (Brazil), usually the kits for this type of training are based in 8 bit microcontrollers. This IMHO will lead to the need of some kind of subset of Ada language, which ultimately may be counterproductive to original objective (spreading Ada IIRC).

From: "Marin David Condic" <marin.condic.auntie.spam@pacemicro.com>
Date: Fri, 23 Feb 2001 10:15:45 -0500
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

[On most "training kits" being based on 8 bit microcontrollers: -- dc]

Well the world has become a lot bigger than 8 bit microcontrollers. I am currently working with a box that has a MIPS processor and almost the whole system on a single chip. So a 32-bit processor able to control some physical/electrical devices from a single board at an inexpensive price is not at all out of the question. The problem is: Which One? If you are familiar with embedded systems, I'm sure you know that there just aren't thousands of Ada ports out there for popular boards/development kits. You need the compiler, plus a good, powerful linker & cross-target debugger along with probably some available libraries, bootstrap code, descent documentation of everything, etc. Saying "Well GNAT has a port to chip X available somewhere on the net..." is interesting, but if you don't have all the pieces pulled together into a nicely integrated package that works reliably, it wouldn't make a good student environment. (It's hard enough for the pros to figure out how to get this sort of thing to work - how much harder would it be for the neophytes? :-)

I'm always interested in hearing ideas on this topic if you have any.

From: Cesar Rabak <csrabak@uol.com.br>
Date: Sat, 24 Feb 2001 18:40:11 -0300
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

[...] As already posted elsewhere in this thread, I second the position that perhaps porting is not the principal problem, but rather to assemble all the pieces, including a targeted tutorial (with exercises), etc. [...]

> I'm always interested in hearing ideas on this topic if you have any.

I think a way to reduce the 'initial' cost of a such project would be to detect the 20% of board/kits which have the 80% of the "market" and have the ports (with all the provisos above mentioned) funded. [...]

From: Marin David Condic <mcondic.auntie.spam@acm.org>
Date: Sun, 25 Feb 2001 10:10:32 -0500
Organization: MindSpring Enterprises
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

[On "32-bit processors able to control some physical/electrical devices from a single board at an inexpensive price": -- dc]

> This is an interesting question! It had to be inexpensive with abundant (and perhaps free) documentation available, and if possible a chip which people would feel it is worthwhile to expend time on it.

I've seen plenty of development kits wherein some company will sell you a developmental version of their SBC, the cross compiler, (related tools), cable to your PC & documentation for in the neighborhood of $500. (Of course, this is with C as the programming language.) That ought to be within the budget of the serious student if it was used for more than one class. It would at least be within the budget of the school's computer lab to have 3 or 4 available for student's to do their lab work on. A company selling the kit might give away several sets of their documentation or make them available at an inexpensive price, so I don't see much standing in the way of students getting the material they need.

The real problem is having a similar environment with an Ada compiler. No technical reason why it couldn't exist at a similar price - just apparently not an economic powerhouse or someone would have likely done it by now. [...]

From: "Marin David Condic" <marin.condic.auntie.spam@pacemicro.com>
Date: Mon, 26 Feb 2001 09:51:40 -0500
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

[On development kits of $500 being too expensive: -- dc]

Well, if you think about it, this may not be an issue at all. Consider the possibility that the school's computer lab has a handful of the development boards & the base compiler is a GNAT variant. A student with a PC could hack the code at home on his PC & make sure it compiles - perhaps even testing parts of it without the SBC, then take it to the lab for full-up testing. No big expenditure there except for the need to have a PC. [...]

To some extent, the "falling in love with Ada" may not be important. *If* there was a really spiffy kit out there that pretty much provided an embedded programming class in a bag, profs would be *really tempted* to use whatever was available, even if they weren't thrilled with the language or SBC architecture. Mostly, this is because it eliminates so much work for them.

[...] You'd need a good textbook with plenty of examples & homework problems based on the SBC you chose. I don't think that is impossible, but it certainly is non-trivial. [...]

Well, as I observed above - even if Ada is not perceived as popular, simply having the kit available would start creating the demand. Profs have way too many other things to do with their time and don't necessarily want to design an intro-level embedded programming course. Or maybe they'd like to have one, but there is a perceived lack of materials. Being the guys who had an off-the-shelf, shrink-wrapped course would be a good position to be in. They'd want it no matter what the details are just because it lets them provide an educational experience at minimal cost & time. [...]

From: "Tarjei T. Jensen" <tarjei.jensen@kvaerner.com>
Date: Thu, 22 Feb 2001 12:56:07 +0100
Organization: Kvaerner Group IT
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

[On commercial possibilities for "an embedded programming course in a bag": -- dc]

It is clearly commercial possibilities if you can find a decent and cheap PC/104 card or a PC motherboard/bios to work with. Then it should be a matter of documenting how to use gnat and/or RTEMS to get results. You may have to write a few drivers for the network card or graphics card. You would have a great teaching tool and an easy kit to commercialize.

From: "Marin David Condic" <marin.condic.auntie.spam@pacemicro.com>
Date: Fri, 23 Feb 2001 10:17:01 -0500
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

I'm taking a look at the PC/104 card info from: http://www.pc104.org/ It looks interesting, but as far as I can tell, you'd probably have some retargeting issues for GNAT no matter what you did. And of course, embedded programming is more than finding GNAT and RTEMS for some target hardware - you really need a lot of additional software for it to be useful. From the hardware side, you'd need to have some basic electrical things like A/Ds, D/As, (F/Ds, maybe?) discretes & ports that would be useful for student projects and representative of real-world development.

I'm going to examine the PC/104 thing a bit more thoroughly. If you think of any other possibilities for an SBC to which GNAT might target with minimal fuss, let me know...

From: "Tarjei T. Jensen" <tarjei.jensen@kvaerner.com>
Date: Fri, 23 Feb 2001 18:22:23 +0100
Organization: Kvaerner Group IT
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

Check with oarcorp. They have something about a GNAT/RTEMS combo on their front page. That might be worthwhile to examine.

[See "GNAT 3.13p Linux RPMs for RTEMS" in AUJ 21.4. -- dc]

I suspect that anything quick and easy for a student/hobbyist to do is hard to find information on. [...] I think the PR value of such a "kit" would be incalcuable because everybody could see that Ada is easy. It would be ridiculously easy to check it out for yourself.

From: "Marin David Condic" <marin.condic.auntie.spam@pacemicro.com>
Date: Fri, 23 Feb 2001 15:40:15 -0500
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

I have some small familiarity with GNAT/RTEMS - which is to say I basically know what it does, but with no knowledge of how it actually does it. Basically, this wouldn't be a bad way to go. [Overview of required tools deleted. -- dc]

Now the problem as I see it is this: Nobody has all these pieces pulled together all in one place using Ada (at a low price, at least), but it *does* exist (mostly) for C and maybe C++. You can go to any number of vendors who will sell you an SBC development kit that will plug into your PC with all the appropriate software at the PC end, etc. You can be up and programming the card with C in short order and maybe the only thing you're really missing is the college level text. Pulling together all this stuff in Ada is certainly feasable, but it would be a non-trivial amount of work.

Since great minds think alike, I'll agree with you that the PR value for Ada would be high because it would demonstrate how easy Ada is relative to C in this arena. I'd go one step further in saying that *if* the kit were to exist, a *lot* of EE profs would be tempted to structure a course around it because it would eliminate a ton of work for them - hence even more PR value. Throw on top of it that every EE student who's first embedded experience is Ada would likely go on to industry with a favorable impression of Ada and start pushing for its adoption. And of course, if the card itself were fairly generally useful, you've got a commercial market for it as well.

My only problem with this idea is that my full-time occupation is not the development of such kits and as a speculative, part-time venture I just don't think I've got the time to do it. (Not in any reasonable timeframe!) Maybe a vendor or professor or idle-rich-kid (or several of them) might get interested and start pulling the pieces together.

From: James Rogers <jimmaureenrogers@worldnet.att.net>
Date: Fri, 23 Feb 2001 19:49:19 GMT
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

Another possibility, although not exactly free, is the PC/104 solution provided by Aonix and PharLap.

PharLap offers a very nice PC/104 RTOS implementing a subset of the Win32 API. Aonix bundles this solution with an Ada compiler that runs on a PC. The Aonix Ada compiler can target either the PC or the PC/104 board, allowing simple unit testing of many packages on the PC, and the remaining testing on the PC/104 board.

The PharLap operating system comes with a useful collection of capabilities including LAN networking (ftp, telnet, http, TCP/IP, etc.)

There could be additional packages created for this solution to address devices not on the PC/104 hardware stack, such as RS232 ports, etc. [...]

Jim Rogers, Colorado Springs, Colorado USA

From: "Marin David Condic" <marin.condic.auntie.spam@pacemicro.com>
Date: Fri, 23 Feb 2001 15:47:23 -0500
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

Now that might just be the basis of an embedded programming course. As I said elsewhere, I think it is a bit like cheating to have some version of an RTOS like Windows or Unix on the card - but realistically speaking, that is what a lot of students will see when they get out of school. (I'd like to see them understand what the LynxOS or VxWorks guys have to do to get *their* code to run!) I suppose if you add some specialized device cards to the stack, then you're providing them with the opportunity to gain the low-level access experience on at least some fronts.

I'll check out the site & take a look at Aonix again to see how much of the solution may already be there. (And at what price!!! $$$$) Maybe a solution is within grasp?

From: "Randy Brukardt" <randy@rrsoftware.com>
Date: Fri, 23 Feb 2001 15:08:00 -0600
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

You can use the PharLap kit with the Windows version of Janus/Ada, as well. While we don't actually sell a bundled version, we will provide support for the combination.

Randy Brukardt, R.R. Software, Inc.

From: "Hans-Olof Danielsson" <Hans-Olof.Danielsson@swipnet.se>
Date: Fri, 23 Feb 2001 22:21:26 +0100
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

Another option is the Lego Mindstorm robotics kit.

The robotics contains a Hitachi H8 microcontroller which is supported by the latest GCC version. Recently there was a posting on the *crossgcc* list regarding building a C cross compiler for H8 with GNU/Linux as host , so there is at least a GCC-based C cross compilor.

When GNAT moves to GCC common source tree (according to earlier posting on this cla-list, that is planed for GCC 3.1) it shouldn´t be to difficult to build a GNAT cross compiler for H8. It might be equaly easy (or difficult) with GCC 2.81 (if it supports H8) used for GNAT 3.13p.

The next step would then be to build a binding to the robotic ROM image, making it possible to control the robot from Ada.

Hans-Olof Danielsson, Danitek AB, Dragspelsv. 20, S-732 32 Arboga, Sweden, Tel: int +46 589 140 38, Email: Hans-Olof.Danielsson@swipnet.se, Web: www.node98.com/danitek, Member of Node98, www.node98.com

From: Jerry Petrey <"jdpetrey <NOSPAM>"@west.raytheon.com>
Date: Fri, 23 Feb 2001 15:26:05 -0700
Organization: Raytheon Company
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

There is a GNAT Ada interface for the Mindstorms kit available at: http://www.usafa.af.mil/dfcs/adamindstorms.htm

It is not complete but a nice start.

[See also "'Fun with Ada' Lab at AdaPower" in AUJ 21.2 (July 2000). -- dc]

From: Rush Kester <Rush.Kester@jhuapl.edu>
Date: Mon, 05 Mar 2001 14:00:59 -0500
Organization: AdaSoft at Johns Hopkins Univ/ Applied Physics Lab
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

There are two Ada interfaces to the Lego Mindstorms robotics kit already available. One is Ada to NQC developed by Barry Fagin at the US Air Force Academy, see http://www.usafa.af.mil/dfcs/adamindstorms.htm The other is a binding Lego's SPIRIT.OCX active X control developed by David Botton, see http://www.adapower.com/gnatcom/mindstorm.zip

Rush Kester, Software Systems Engineer AdaSoft at Johns Hopkins Applied Physics Lab.

From: dirk@cs.kuleuven.ac.be (Dirk Craeynest)
Date: 27 Feb 2001 00:49:58 +0100
Organization: Ada-Belgium, c/o Dept.Computer Science, K.U.Leuven
Subject: Model railroad package (was: Re: Increased Interest In Ada?)
Newsgroups: comp.lang.ada
Summary: John McCormick is working on it

> I could imagine Ada being popular in electrical engineering departments if there were a convenient and inexpensive (maybe free?) Ada environment for playing around with embedded computing. [...] I am thinking of Dr. McCormick's model railroad class [...] If [...] was packaged as "An embedded programming course in a bag" so that a prof could just pick it up and start teaching it, this might go a long way toward encouraging Ada as an educational tool as well as a practical tool for building real-world systems. (Does anyone smell commercial possibilities here? :-)

Some time ago, John McCormick told me he was looking into exactly that: commercialising the hardware and software for his model railroad class in order for others to be able to buy it "off the shelf" and use it for similar classes at other universities.

I contacted him to check what progress has been made. John's reply is reposted here for your information, with his approval.

[John McCormick wrote: -- dc]

[...] I am indeed working on just such a project. Ultimately it would include:

1. Printed circuit boards to make it easy to connect a model railroad to one or more computers. The connections are made through D/A, A/D, and digital I/O on the computer end which should make it possible to use just about any computer

2. An Ada environment for developing software.

3. A sample program that can be easily adapted to any railroad geometry.

4. An undergraduate level introductory textbook on real-time embedded systems based on the train layout. Of course, it will use Ada as the development language. This piece of the project is still in the dreaming stage.

The hardware development is being supported by two local corporations: Rockwell-Collins (avionics) and Maytag (household appliances). My target date for completing the hardware is the end of this summer.

Feel free to pass this information on to anyone interested.

John W. McCormick, mccormick@cs.uni.edu, john.mccormick@acm.org Computer

Science Department, University of Northern Iowa, Cedar Falls, IA 50614-0507, voice (319) 273-2618, fax (319) 273-7123 http://www.cs.uni.edu/~mccormic/

Ada-related Resources

Revised Ada Reference Manual (ARM) and Ada Issues (AI)

From: Jeffrey Carter <jrcarter@acm.org>
Date: Mon, 12 Mar 2001 04:09:54 GMT
Subject: Re: Reference manual
Newsgroups: comp.lang.ada

The place to start with Ada questions is www.adapower.com And you'll find a version of the ARM there.

A revised ARM incorporating accepted Ada Issues (AIs) is available at http://www.ada-auth.org/~acats/

From: Tucker Taft <stt@averstar.com>
Date: Tue, 13 Mar 2001 13:38:03 -0500
Organization: AverStar (formerly Intermetrics) Burlington, MA USA
Subject: Re: XML/HTML DOM in Ada
Newsgroups: comp.lang.ada

[...] An overall (html-ized) index of AIs is now available (thanks, Randy!) at: http://www.ada-auth.org/AI-SUMMARY.html

Tucker Taft, stt@avercom.net, http://www.averstar.com/~stt/, Chief Technology Officer, AverCom Corporation (A Titan Company), Burlington, MA, USA (AverCom was formerly the Commercial Division of AverStar)

From: "Randy Brukardt" <randy@rrsoftware.com>
Date: Tue, 13 Mar 2001 13:09:01 -0600
Subject: Re: XML/HTML DOM in Ada
Newsgroups: comp.lang.ada

[In reply to a reference to the Ada Issues (AI) on the www.adaic.org website: -- dc]

That's the wrong URL. The AIs on AdaIC.org are not maintained, and are very stale. (The ARA is starting a housecleaning on AdaIC.org, so perhaps this will change.)

The current set of AIs is found at: www.ada-auth.org/ais.html

New feature: The summary is now available in HTML with links to the actual AIs, so you don't even have to wade through a text file.

Randy Brukardt, ARG Editor

More Ways to Access Comp.lang.ada Newsgroup

From: "Gary L. Howell" <howellg@wes.army.mil>
Date: Fri, 13 Apr 2001 13:38:41 -0500
Subject: Re: Ada recommendation
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

[See also "Comp.lang.ada mailing list" in AUJ 21.4 (January 2001). -- dc]

Yet another way... I read c.l.a on the Web via Google (aka DejaNews). http://groups.google.com/groups?hl=en&lr=&safe=off&group=comp.lang.ada

Ada-related Tools

PragmAda Reusable Components

From: Jeffrey Carter <jrcarter@acm.org>
Date: Wed, 16 May 2001 04:01:20 GMT
Subject: Announce: PragmAda Reusable Components Release
Newsgroups: comp.lang.ada

A new release of the PragmARCs is now available at

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

Hopefully the mirror at www.adapower.com will be updated soon.

[See also AUJ 22.1 (March 2001). -- dc]

This release includes the long-awaited (almost) universal date-formatting functions in PragmARC.Date_Handler. If you want to format dates or times using Standard.Character and cannot achieve the desired format using these functions, we'd like to hear about it.

This release also includes minor improvements to a number of components. See readme.txt for complete information.

Jeffrey R. Carter, PragmAda Software Engineering

Booch Components

From: Simon Wright <simon@pushface.org>
Date: Wed, 16 May 2001 05:11:17 +0100
Subject: Booch Components 20010513
To: team-ada@acm.org

This release has been uploaded to

http://www.pushface.org/components/bc/

(still an alias for the old http://www.pogner.demon.co.uk/components/bc/) and is mirrored at http://www.adapower.net/booch/ .

[See also AUJ 22.1 (March 2001). -- dc]

Features:

Added BC.Containers.Quicksort, BC.Containers.Shellsort generics (implemented as child packages, on the theory that you won't want the clutter in BC.Containers if you don't need it).

Removed BC.Containers.Copy, BC.Containers.Filter generics (use BC.Copy, BC.Filter).

Bounded forms have an Available function, returning the number of free entries. The same function is now available for Dynamic and Unbounded forms (in which case it returns Natural'Last; just don't believe it!

Added Synchronized forms for Dynamic and Unbounded Maps.

Added BC.Containers.Maps.Hash_Statistics to report how well the hashing algorithm is doing. Easy enough to do the same for Bags, Sets ...

ArcFour - Cryptography

From: "Julian Morrison" <julian@extropy.demon.co.uk>
Date: Mon, 26 Feb 2001 03:27:26 +0000
Subject: Arcfour in Ada
Newsgroups: comp.lang.ada,sci.crypt

http://download.sourceforge.net/fling/arcfour-ada-1.0.0.tar.gz

This code has been created for use with the Fling project (http://fling.sourceforge.net/).

This is ArcFour (Alleged RC4), CipherSaber variant, capable of CipherSaber-1 and CipherSaber-2. It is coded in Ada, and is dependent on AUnit and Formatted_Output (available via the AdaPower site). It's probably pretty GNAT-dependent too, since I've had no need to compile it anywhere else. If you want fixes, send patches and/or bug reports via Fling's SourceForge patch tracker.

This code has been placed in the public domain by its author.

Release notes: first full release, all unit tests pass, but it may be implementation dependant.

Dynamically Growing Arrays

From: Stephen Leake <stephen.a.leake.1@gsfc.nasa.gov>
Date: 07 Mar 2001 16:33:56 -0500
Organization: NASA Goddard Space Flight Center
Subject: Re: Got bitten by GNAT.Table
Newsgroups: comp.lang.ada

[In response to a problem with the GNAT.Table package: -- dc]

GNAT.Table is not implemented in "pure" Ada, so it has some restrictions. Be sure to read the spec very carefully. [...]

Or, use sal.poly.unbounded_arrays, from http://users.erols.com/leakstan/Stephe/Ada/sal.html. That's a pure Ada solution.

Serial Communication

From: Stephen Leake <stephen.a.leake.1@gsfc.nasa.gov>
Date: 07 Mar 2001 16:38:30 -0500

Organization: NASA Goddard Space Flight Center
Subject: Re: Serial communications PC - RS232 with Ada ?
Newsgroups: comp.lang.ada

> I want to use the COM: ports (COM1, COM2 ...) of the PC to communicate within an Ada program (send and receive characters). Where can I found exemples or sources of code?

http://users.erols.com/leakstan/Stephe/, see the "com ports" package.

Ada Game Development

From: Erik Sigra <sigra@home.se>
Date: Thu, 8 Mar 2001 21:57:37 +0100
Subject: Re: Ada game development
Newsgroups: comp.lang.ada

> anyone know of a good site about Ada game development? thanks.

Here is some stuff (all free software and/or open source):

* António.M.F.Vargas is developing SDL bindings. See <http://www.adapower.net/~avargas/>.

* Michael Hope, Sandeep Dutta and Daniel Rammelt are developing the Gameboy Development Kit. See the sourceforge page at <http://gbdk.sourceforge.net>.

* Bobby D. Bryant is developing a game called Yasfync. See the homepage at <http://www.cs.utexas.edu/users/bdbryant/yasfync/index.html> and the announcement at <http://arch.freeciv.org/freeciv-dev-200101/msg00022.html>.

* Wolfgang Jeltsch and Kyrill Meyer are planning a project called Adventurer's Support Kit. See the sourceforge page at <http://sourceforge.net/projects/ask/>.

[From another reply: -- dc]

And don't forget OpenGl
www.opengl.org

Calendar and Date Operations

From: "Marin David Condic" <marin.condic.auntie.spam@pacemicro.com>
Date: Tue, 13 Mar 2001 09:24:37 -0500
Subject: Re: calander package
Newsgroups: comp.lang.ada

[In reply to a request for date formatting operations: -- dc]

What you'll have to do is take the individual pieces of the date/time and convert them to appropriate strings. There are no subprograms defined in Calendar to automatically give you the date/time as string values.

If you want some help with this, check out my web page: http://www.mcondic.com/ and look for the "Ada Programming" subpage. There's a collection of code there under the title "Utilities" in which you'll find a package that formats dates/times into various styles.

Marin David Condic, Senior Software Engineer, Pace Micro Technology Americas, www.pacemicro.com, e-Mail: marin.condic@pacemicro.com, Web: http://www.mcondic.com/

From: Ted Dennison <dennison@telepath.com>
Date: Tue, 13 Mar 2001 18:52:43 GMT
Subject: Re: calander package
Newsgroups: comp.lang.ada

> [...] Considering that many languages come with existing string-formatted dates/times &/or subprograms [...] Ada's Calendar package probably should have had this in it somewhere, but its no biggie to roll-your-own. [...]

Well, with that caveat, I'll mention that the 2.0 release of the SETI@Home service just happens to have a routine in it that does exactly that (for date/time-stamping log entries). It'd probably be just as easy for most folks to write their own, though.

T.E.D. homepage
http://www.telepath.com/dennison/Ted/TED.html

From: "Randy Brukardt" <randy@rrsoftware.com>
Date: Tue, 13 Mar 2001 15:47:54 -0600
Subject: Re: calander package
Newsgroups: comp.lang.ada

> I've built my own to generate the date/time in a whole variety of formats because on any number of occasions you might need it different ways. [...]

I've done this dozens of times, too. So, when the problem came up with the Smplsrvr, we went ahead and added a package of operations to Claw. The Claw.Time package really doesn't depend on Claw itself (nor on Windows), so it could be used separately. It's in the Claw Introductory version, which is available for free non-commerical use at www.rrsoftware.com.

I was thinking about proposing something on this line to the ARG for standardization, but it unclear that the ARG is interested in (semi-)standardizing these sorts of packages. (It seems to run about 50-50 within the membership as to whether time should be spent on it. The objections mostly are that there are more important things to work on.)

From: Vincent Marciante <marciant@li.net>
Date: Tue, 13 Mar 2001 21:04:02 -0500
Subject: Re: calander package
Newsgroups: comp.lang.ada

Randy Brukardt wrote: I was thinking about proposing something on this line to the ARG for standardization, [...]

The original Booch.Calendar_Utilities could be considered.

http://www.adapower.com/original_booch/

From: Jeffrey Carter <jrcarter@acm.org>
Date: Wed, 14 Mar 2001 02:19:55 GMT
Subject: Re: calander package
Newsgroups: comp.lang.ada

With the same caveat, I'll mention PragmARC.Date_Handler, part of the PragmAda Reusable Components. See

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

or the mirror at www.adapower.com.

From: Pascal Obry <p.obry@wanadoo.fr>
Date: 14 Mar 2001 09:33:32 +0100
Subject: Re: calander package
Newsgroups: comp.lang.ada

In case you are using GNAT have a look at GNAT.Calendar and GNAT.Calendar.Time_IO.

Pascal Obry, Team-Ada Member, 45, rue Gabriel Peri, 78114 Magny Les Hameaux, France
http://perso.wanadoo.fr/pascal.obry

From: Mario Amado Alves <maa@di.fct.unl.pt>
Date: Wed, 14 Mar 2001 11:52:57
Subject: Re: calander package
Newsgroups: comp.lang.ada

Package Datetime provides conversion between date/time legible formats and Ada Time values. It is an Ad=AAlib package, category Miscelany:

http://lexis.di.fct.unl.pt/ADaLIB/

Departmento De Informatica, FCT/UNL, 2825-114 Caparica, Portugal

From: "Marin David Condic" <marin.condic.auntie.spam@pacemicro.com>
Date: Thu, 15 Mar 2001 12:45:00 -0500
Subject: Re: calander package
Newsgroups: comp.lang.ada

You might look over the stuff in the Utilities code on my web page (search for "date_tools" in the .chop file, IIRC.) I've got code there that delivers a large number of different formats. I may even have written a test driver for it - maybe. (Senility setting in again! :-) You may find one there you like - or maybe you suggest a new one and when I get that elusive Round Tuit, I might add it...

From: Peter Hermann <ica2ph@iris16.csv.ica.uni-stuttgart.de>
Date: 22 Mar 2001 09:05:51 GMT
Organization: Comp.Center (RUS), U of Stuttgart, FRG
Subject: calendar,day&time calculations
Newsgroups: comp.lang.ada

Long time ago.... ;-) I rolled my own additional calenday package and use it since years: http://www.csv.ica.uni-stuttgart.de/homes/ph/adapilotresources/basic_tools/calenday.ads

The current discussion triggered my mind to go to brush up my calendar/date/time-related tools with new useful ideas. GNAT.calendar is of course a fine starting point for everyone but I generally prefer using fully portable independent software. In order to make my private stuff possibly eligible for public use (e.g. potential contribution for GNAT), I ask you all to send me recommendations, ideas, and even suggestions for change of spec, etc., related to date and time calculations and their string representations.

Keywords: locale adjustment, pattern, genericity, flexibility, usefulness, open software.

Gathering ideas...

Peter Hermann, D-70569 Stuttgart, Uni Computeranwendungen
http://www.csv.ica.uni-stuttgart.de/homes/ph/

From: tmoran@acm.org
Date: Tue, 08 May 2001 04:38:42 GMT
Subject: Re: Seconds since Epoch
Newsgroups: comp.lang.ada

> Is there an easy way to just get a big number of seconds since 01/01/1970 epoch

```
procedure Difference (Left, Right :
            Ada.Calendar.Time;
   Days : out Day_Count;
   Seconds : out
            Ada.Calendar.Day_Duration);
-- Returns the difference between Left
   and Right. Days is the number of days
   of difference, Seconds is the remainder
   seconds of difference. If Left < Right,
   then Days is negative. (Seconds is
   always positive).
```

is available in package claw.time in the Claw free intro version. Download it from www.rrsoftware.com

From: serious@clerk.com (Noam Kloos)
Date: Mon, 14 May 2001 12:30:39 GMT
Subject: handy date routines
Newsgroups: comp.lang.ada

I have made a few simple programs to calculate days offset from given or current date.

http://www.noam.nl/download/ada95/days2date.zip
http://www.noam.nl/download/ada95/date2date.zip

Ada Mode for Vim Text Editor

From: David Wheeler <dwheeler@ida.org>
Date: Wed, 28 Mar 2001 12:01:54 -0500
Subject: Ada mode for vim text editor.
To: team-ada@acm.org

If you use "vim" or "gvim" as your text editor for editing Ada source code, please try out my new syntax highlighting mode. vim's syntax highlighting colors different keywords differently, and the new version is (hopefully) an improvment.

You can download this new vim Ada mode from: http://www.dwheeler.com/vim

Note that on some systems, running "vi" actually invokes vim.

Unit Testing Tools

From: "Dr. Michael Paus" <paus@ib-paus.com>
Date: Thu, 10 May 2001 12:56:29 +0200
Subject: Re: AUnit
Newsgroups: comp.lang.ada

[In response to questions about the usability of AUnit for students: -- dc]

I think the use of [AUnit or JUnit (the Java counterpart to AUnit)] is quite straight forward if you have a complete example which you can then extend with further test cases. Even if you do not understand object oriented concepts like tagged types etc. you can just copy and extend an example and so I think it should be possible to teach students how to use this tool even if they do not completely understand why it works.

[See also "ACT - AUnit test framework for Ada" in AUJ 21.4 (January 2001). -- dc]

From: Ed Falis <falis@gnat.com>
Date: Fri, 11 May 2001 21:53:29 GMT
Organization: Ada Core Technologies
Subject: re: AUnit
Newsgroups: comp.lang.ada

Just a quick note regarding the usability of AUnit. GNATPro 3.14a and beyond include an extra submenu [in their integrated development environment GLIDE -- dc] for AUnit that generates all the boilerplate code for test cases, suites and harnesses. The result is that you really only have to manually deal with the specific test procedures and any context they may require.

From: Pascal Obry <p.obry@wanadoo.fr>
Date: 10 May 2001 20:23:10 +0200
Subject: Re: AUnit
Newsgroups: comp.lang.ada

To test units I've been using Test Driver Generator from Andre Spiegel. Latest version is 3.1.

ftp://ftp.informatik.uni-stuttgart.de/pub/local/

[See also AUJ 21.4 (January 2001). -- dc]

I like it and found it easy to use. There is even an Emacs mode to support the Driver Generator script :)

From: Poutanen Olavi <olavip@kaarne.cs.tut.fi>
Date: 13 May 2001 08:10:30 GMT
Organization: Testwell
Subject: Re: AUnit
Newsgroups: comp.lang.ada

If your project is at Ada 83 level you might want to look at Testwell's TBGEN as a unit testing tool for Ada 83. If interested, please have a look at http://www.testwell.fi (company) or http://www.testwell.fi/tbgdesc.html (straight to TBGEN)

Olavi Poutanen, Testwell

AdaGraph v0.6 - High-Resolution Color Graphics

From: Jerry van Dijk <jvandyk@attglobal.net>
Date: 12 Mar 2001 03:00:45 +0100
Subject: homepage back online
Newsgroups: comp.lang.ada

A quick note to say that after the unexpected demise of the server hosting my homepage, a updated version is back online at:

http://home.trouwweb.nl/Jerry

You will also find the sourcecode of AdaGraph v0.6 beta (with bitmap support) here. Let me know of any problems, additions, etc.

[See also "AdaGraph-X11 Courseware" in AUJ 21.3 (October 2000). -- dc]

Jerry van Dijk, jdijk@acm.org

OpenGL Bindings and Demos

From: Pascal Obry <p.obry@wanadoo.fr>
Date: 17 Apr 2001 09:34:17 +0200
Subject: Re: Opengl bindings for Gnat
Newsgroups: comp.lang.ada

I have a GLUT and OpenGL demos on my Web site ported to GNAT. It uses the Win32.Gl package for the OpenGL part.

http://perso.wanadoo.fr/pascal.obry/contrib.html

From: "Joachim Schröer" <joachim.schroeer@web.de>
Date: Tue, 8 May 2001 00:47:27 +0200
Subject: OpenGL - framework for Win32; Xearth in Ada. New freeware on Adapower!
Newsgroups: comp.lang.ada

OpenGL - framework for Win32; Xearth in Ada

This is the new page http://www.adapower.com/schroer. It contains some Ada reuse source as well as a linked application ready for execution.

The following zip-files are given:

- lib-src.zip - Source of an Ada library with some utilities and components. This link is wrong in http://www.adapower.com/schroer, use http://www.adapower.com/schroer/lib-src.zip for download.
- ogl-src.zip - Source of an OpenGL framework for Win32 including an application (Ada-main).

- opengl-exe.zip - The main-executable for MS-windows.
- dlls-etc.zip
- SGI-DLLs for OpenGL on MS-windows.
- bitmaps.zip - Bitmaps used by the application.

For the documentation of the Opengl-Win32 framework see http://www.adapower.com/schroer/ogl.htm .

The application shows an analog clock and/or a rotating earth similar to the famous xearth background image one can find on Unix. It uses a photograph of the complete unclouded earth for texturing.

As my spare time and my two little children allow, there will be future activities on this page:
- Adaptation of a subset of LINPACK and EISPACK numerical algorithms for Ada95 still in Ada83 style.
- Adaptation of SW for symbolic mathematics to my new lib-src standards.
- Adaptation of SW for modeling and simulation of dynamic system models (differential equations, state-space models; parts of my thesis).
- Adaptation of the above OpenGL framework for GLUT and a port to linux.

The Ada community will be informed in Comp.Lang.Ada when SW updates on this side have been conducted.

For comments please mail to Joachim Schröer (joachim.schroeer@web.de)

GtkAda 1.2.11 Release

From: charlet@paris.act-europe.fr (Arnaud Charlet)
Date: Mon, 12 Mar 2001 17:14:55 GMT
Subject: ANNOUNCE: GtkAda 1.2.11
Newsgroups: comp.lang.ada

GtkAda is an Ada95 graphical toolkit based on Gtk+ version >= 1.2.2. It allows you to develop graphical applications in Ada95 using Gtk+.

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

[See also AUJ 21.3 (October 2000). -- dc]

New features in 1.2.11 include:

- Support for GNOME is detected and handled automatically by configure
- Many additional GNOME widgets bound (but are provided as is: they have not been tested).
- A new utility is provided to ease the automatic generation of bindings for new Gtk+/GNOME widgets. See the GtkAda User's Guide for more details.
- Under Windows, non ASCII characters are now displayed instead of being truncated.
- New routines to update and copy XML trees easliy in Glib.XML
- Documentation additions
- various refreshing/displaying problems fixed in the GUI builder under Windows
- new functions Gtk.Viewport.Get_Bin_Window, Gtk.Notebook.Get_Allocation_[Height, Width, X, Y]
- gmodule is now bound in Glib.Module
- libglade is now bound in Glade and Glade.XML, and obsoletes the capability previousely provided in Gtk.Glade
- Gtk.Glade.Register_Generate allows to add easily support for new widgets in Gate
- The following widgets are now recognized automatically and no longer require the use of Gtk.Type_Conversion: Gtk_Item, Gtk_Check_Menu_Item, Gtk_Radio_Menu_Item, Gtk_Tearoff_Menu_Item, Gtk_Tree_Item.
- A new function, Gtk.Unchecked_Cast, provides an alternative to Gtk.Type_Conversion.
- The Gtkada.Canvas has been partly rewritten and now uses less memory. It also provides scrolling while moving items.
- A few bug fixes, as described in the known-problems file

GNU Visual Debugger (GVD 1.1.0)

From: charlet@paris.act-europe.fr (Arnaud Charlet)
Date: Mon, 12 Mar 2001 17:22:57 GMT
Subject: ANNOUNCE: Release 1.1.0 of the GNU Visual Debugger
Newsgroups: comp.lang.ada

We are pleased to announce the availability of GVD, the GNU Visual Debugger, a general purpose graphical debugger front-end licensed under the GNU General Public License.

Besides providing all the features of other debugger GUIs, GVD includes advanced data display and visualization capabilities. Furthermore, GVD allows the debugging of multi-process/multi-threaded applications in the same debugging session. GVD works with native as well as cross debuggers and can handle several languages in the same debugging session and the same application. Currently Ada, C and C++ are supported.

GVD can run on a host different from the machine where the debugger is running and provides friendly support for cross-debuggers (VxWorks, Lynx, etc.). For instance, you can use Linux or Windows to debug an application running on a Power PC board with a debugger running on a Sun workstation.

To build GVD we are using the GtkAda GUI technology. GVD comes with all the GtkAda benefits such as a pluggable look-and-feel, a set of very high-level widgets and the ability to have the same look-and-feel on all of your platforms.

You can download GVD (sources and binaries for GNU/Linux x86, Solaris sparc and x86, Windows NT/2000, DEC Unix/Tru64, HP-UX, UnixWare, IRIX, AiX) and get more information at http://libre.act-europe.fr

[See also AUJ 22.1 (March 2001). -- dc]

New features in GVD 1.1.0:

- Complete documentation now available.
- Under Windows, non ASCII characters are now displayed instead of being truncated.
- New command line option: --pargs to set program arguments on the command line.
- New command line option: --dargs to specify additional debugger arguments.
- Start up of big programs have been significantly improved.
- Menu "Print" in the code editor contextual menu now prints the variable inside the command window instead of the canvas.
- Support for preferences (menu Edit->Preferences), with many user-configurable options. See documentation for more details.
- Scrolling of assembly window outside of displayed range is now possible.
- The Data window can now be separated from the main window (Edit->Preferences, Data/Separate Window).
- The Memory View now handles editing in non-Hex mode, and the possibility of displaying ASCII and another format at the same time.
- The different window settings are now remembered from one session to another.
- New command line option --editor-window that gives a much better integration with external editors such as Glide/Emacs.
- A graphical list of processes is now displayed when attaching to a process.
- New item in data and source editor contextual menus: view memory at the address of a given variable.
- New item in the source editor contextual menu: print dereferenced variable.

If you are interested in participating in the GVD development, do not hesitate to contact us (mailto:gvd-devel@lists.act-europe.fr)

For the GVD team,

Arnaud Charlet <charlet@act-europe.fr>

VAD 5.4 - Visual Ada Developer

From: Leonid <dulman@attglobal.net>
Date: Sun, 11 Mar 2001 18:23:55 +0300
Subject: Announce: Visual Ada Developer (VAD) version 5.4 Leonid Dulman
Newsgroups: comp.lang.ada

VISUAL ADA DEVELOPER (VAD) version 5.4

VAD (Visual Ada Developer) is a tcl/tk oriented Ada-95 GUI builder which portable to difference platforms, such as Windows NT/9x, Unix (Linux), Mac and OS/2.

What’s new

1. From Build menu you may GENERATE script for compiling and building executable for project

2. NEW package Rcom.Draw for easy drawing in canvas lines, ovals, rectangles, texts, ... (draw sample)

3. NEW package Adagraph for more powerfull graphic operations (adagr sample)

4. New version of QuickimeTcl 3.0 is available to tcl/tk 8.3/8.4

5. Powerfull Tktable package support (you may easy build 2D tables from ADD WIDGETS page)

6. CScrollbar (more nice scrollbar ADD WIDGETS page)

7. Barchart, Graph widgets from BLT package ADD WIDGETS page

8. New disp(layout) setwidget for BWidgets (such as ScrollableFrame)

9. Help F1 for selected widget (mouse on widget gif)

[See also AUJ 21.3 (October 2000), pp.163-164, and AUJ 22.1 (March 2001) for more details. -- dc]

VAD 5.4 is available in http://members.xoom.com/ldulman/vad.htm

You may dowload sources vad54scr.zip, vadhlp.zip, adahlp.zip, vad54tcl.zip, vadsmp.zip, adastyle.zip, philosophers.zip, vad50idl.zip and binaries vad54win.zip, pkg83win.zip, tcl84win.zip (WINDOWS 9x/NT), vad54lin.zip, pkg83lin.zip, tcl84lin.zip (OpenLinux 2.x,RedHat 7.x).

Leonid Dulman (dulman@attglobal.net)

AWS - Ada Web Server Package

From: Pascal Obry <p.obry@der.edf.fr>
Date: 08 Mar 2001 09:38:42 +0100
Organization: Electricite De France - http://www.edf.fr
Subject: ANNOUNCE: New AWS version (0.9.9)
Newsgroups: comp.lang.ada,fr.comp.lang.ada

[See also "AWS - Ada Web Server package" in AUJ 21.4 (January 2001), pp.228-229. -- dc]

Dmitriy Anisimkov and I are happy to announce the availability of the AWS 0.9.9 release. [...]

AWS stand for Ada Web Server. It is not a real Web Server like Apache. It is an HTTP component to embedded in any application. AWS is fully developed in Ada with GNAT.

This new version has a lot of changes:

- Many bugs have been fixed (as always!)
- Add first version of a configuration file (aws.ini).
- Add logging facility (log format is very close to the Apache one). The log file can be configured to change daily or monthly.
- AWS support select HTML tags with multiple values (i.e. a parameter name can be given a list of values)
- Add an API to use AWS as a communication layer between programs.
- Add support for Hotplug module (dynamically bound module into a running Web server). See documentation. This uses the communication API. Support for Hotplug modules is at a beta stage.
- POSIX implementation of the OS_Lib.GMT_Clock function now returns GMT time. The GMT_Clock was correct only with the GNAT and Win32 implementation of OS_Lib.
- Client method HEAD is implemented.
- AWS.Status is lightler because all routines to set the status have been moved to another unit.
- AWS.Client now support Keep-Alive connection.
- We have simplified the makefile and build process, but this needs certainly some more improvements.
- And again, improve the documentation.

[See AWS 0.10 message below for a full announcement. -- dc]

From: Pascal Obry <p.obry@wanadoo.fr>
Date: 10 Mar 2001 22:50:53 +0100
Subject: ANNOUNCE: new AWS version (0.9.10)
Newsgroups: comp.lang.ada, fr.comp.lang.ada

Dmitriy Anisimkov and I are happy to announce the availability of the AWS 0.9.10 release. [...]

This is a maintenance release:

- hotplug cb.ads was missing from the distribution. Reported by Sune Falck.
- Can be used with latest version of Adasocket 0.1.12 and 0.1.13. With this release you must use the updated Win32 port (see link below). Reported by Sune Falck.
- In the administrative page, if the socket is not opened a dummy value was reported, not the character minus is displayed. Reported by Sune Falck.
- some minor bugs have been fixed.

[See AWS 0.10 message below for a full announcement. -- dc]

From: Pascal Obry <p.obry@wanadoo.fr>
Date: 01 May 2001 16:43:43 +0200
Subject: [ANNOUNCE] - New AWS (Ada Web Server) version (0.9.11)
Newsgroups: fr.comp.lang.ada, comp.lang.ada

Dmitriy Anisimkov and I are happy to announce the availability of the AWS 0.9.11 release. [...]

Here are the main changes:

- Fix bug in Keep-Alive connection handling in the server. If client does not ask for a non Keep-Alive connection we assume a Keep-Alive connection. This conform to RFC 2616.
- New rountine AWS.Response.URL to jump to a given Web page.
- AWS now use the new Templates_Parser API. This version should be stable now. Templates_Parser is used to display the status page.
- The main demo (runme) can now be launched as a Windows NT/2000 service. This uses Ted Dennison Ada Services library.
- It is now possible to specify the certificate to use for the SSL connection.
- Improve (slightly) the documentation.
- Do not use GIF images anymore, we use PNG images.
- As always some bugs have been fixed.

[See AWS 0.10 message below for a full announcement. -- dc]

From: Pascal Obry <p.obry@wanadoo.fr>
Date: 24 May 2001 11:37:10 +0200
Subject: [ANNOUNCE] - AWS (Ada Web Server) version 0.10
Newsgroups: comp.lang.ada,fr.comp.lang.ada

Dmitriy Anisimkov and I are very happy to announce the availability of the AWS 0.10 release. This version is close to the 1.0 version. Note that the API has been changed a lot. This is part of a redesign to have a cleaner API. We plan to change slightly the API at this stage but it should be mostly stable.

Note that this is definitly a major version.

AWS stand for Ada Web Server. It is not a real Web Server like Apache. It is an HTTP component to embedded 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.

Here are the main changes:

- API redesign.
- Templates_Parser (included) has been almost completly rewritten. It is something like 8 to 12 times faster than previous version, has lot of nice new features (like Matrix_Tag). See below for references about Templates_Parser module.
- Fix bug in session handling. The same session ID could have been allocated to differents client. Generation of Session ID is more secure. And there is some performance improvement in the way sessions ID are handled.
- New Server interface (more dynamic). HTTP has only one discriminant now, the other setting are done through the Start procedure.
- Default AWS.OS_Lib is now using the GNAT based implementation instead of the POSIX one. This should make it easier to build AWS.

- Implement HTTP/1.0 and Keep-Alive connection (Netscape browser ask this kind of connection). Should fix more server hanging problems.
- Server parameters can be handled with case sensitivity or not.
- Fix possible memory leak in status data.
- Improve again the way slots are aborted, this should fix more browser hanging problems.
- Add Peername to the status data.
- Status page (status.tmplt) use the new Matrix_Tag templates parser feature and the new filter syntax. Note that this new version of templates parser is about 8 to 12 times faster than previous version. It has been completely rewritten.
- All status page (status.tmplt) vetor tag name have now an _V suffix (was _L for historical reasons)
- Handle properly all "Cookie:" HTTP messages format.
- Add many configuration options in aws.ini.
- Improve a bit the documentation.
- Hello_World new AWS demo, the famous Hello_World a la AWS. This is certainly the smallest AWS application.
- As always some minor bugs have been fixed but are not listed here.

NOTE: Since we have switched to the .PNG file format we have found that Netscape Navigator is not able to display the PNG transparent layer properly!

At this stage we feel that AWS is ready to build small to medium Web server. AWS has been reported to work under Windows NT, Linux and FreeBSD 4.1.

With this new version you'll need at least version 0.1.11 of the Socket binding from ENST. It has been tested and works fine with version 0.1.13 too.

You can download AWS and the Sockets binding for Win32 directly from: http://perso.wanadoo.fr/pascal.obry/contrib.html

An online version of the documentation can be found at http://perso.wanadoo.fr/pascal.obry/aws.html.

The OpenSSL libraries (optional) distributed are for Windows GNAT 3.13. GNAT 3.12 users must build the libraries from sources or obtain Win32 binaries from http://vagul.tripod.com/libssl.zip.

Under UNIX you'll have to build the libraries from sources, it is quite easy to do so. This has been tested under Linux without trouble.

See documentation for build information.

Pointers:

- AWS documentation: http://perso.wanadoo.fr/pascal.obry/aws.html

- AWS sources: http://perso.wanadoo.fr/pascal.obry/contrib.html

- Templates_Parser sources: Templates_Parser module is provided with AWS distribution. Latest version of this module and the documentation can be found at:

http://perso.wanadoo.fr/pascal.obry/contrib.html
http://perso.wanadoo.fr/pascal.obry/templates_parser.html

Templates_Parser is a very useful add-on for AWS. You should have a look at it if you plan to develop a Web service. Templates_Parser permits to completly (yes 100%) separate the HTML design from the Ada code.

Some other Templates engines are WebMacro, FreeMarker, PHP, ASP, JSP and Velocity. All of them are based on explicite iterators (#foreach with a variable) where Templates_Parser is based on implicit ones (you use a more intuitive table iterator). Be sure to check the documentation. Only Velocity project will support complete separation of HTML design and code.

- Socket binding: for Win32: http://perso.wanadoo.fr/pascal.obry/contrib.html for UNIX: http://www.infres.enst.fr/ANC/

- POSIX Binding (optional) : for Win32: http://perso.wanadoo.fr/pascal.obry/contrib.html for UNIX: http://www.cs.fsu.edu/~baker/florist.html

- OpenSSL library (optional) : Sources for UNIX or Win32: http://www.openssl.org (we have used and we distribute OpenSSL version 0.9.5a with this AWS release, we have also tested AWS with OpenSSL 0.9.6a without trouble) binaries for Win32 with GNAT 3.13 (and later): Included with the main AWS distribution. binaries for Win32 with GNAT 3.12: http://vagul.tripod.com/libssl.zip See OpenSSL license (docs/openssl.license).

- Windows Services API (optional): To build runme demo as a Windows NT/2000 services you must download the services API made by Ted Dennison for his SETI@Home project. http://www.telepath.com/dennison/Ted/SETI/SETI_Service.html

Reporting bugs: you can report bugs to Dmitriy Anisimkov: anisimkov@yahoo.com,or: Pascal Obry p.obry@wanadoo.fr

It would be nice if you could also sent us a note if you are using AWS just to know if it is used at all or not :)

AWS uses:

- SETI@Home from Ted Dennison. AWS is used as a "plugable" GUI to control the services status.

[See also "SETI@Home Service's Usage of AWS" further in this AUJ issue. -- dc]

- DOCWEBSERVER from Wiljan Derks

In our department we keep our documents in a directory tree. These documents are all project related and have a certain naming convention to be able to find the right document. In the past I already wrote a program that searches though this directory and then converts the found documents into fixed html pages. With AWS I was able to get a much nicer setup. I have now a server that can do the following: - browse through the projects in explorer style. The html contains info about the document like date and title. - one can check in documents through the web interface - it shows our download page as I have send you in the example - we have now all our documentation in small pieces of html as is needed to build .chm (w2k compiled help) files. For these we use a content file, that is also stored in the document archive.

The docwebserver gives by reading all this stuff the direct view on this documentation. On the other hand I can run some tool and automatically generate the .chm files.

- OESM Server (OESM = Overall Equipment Status Monitoring) from Wiljan Derks

I am working on a project now for our factories. ITEC mainly delivers equipment for discrete semiconductor assembly. Allmost all of that equipment is now controlled by a similar Ada 95 based code with having a lot of code in common. One of the common things, is the way we log errors and state changes of our equipment.

The OESM Server is an application which copies all this information continuously to its local pc by opening the proper files on the remote equipment. That data copied is also stored in local files. The web server component of the application can then, making use of that data, give reports that show things like the amount of products produced in a certain period, error paretos of equipment, mtbf, %time in production and of course many other things.

The cool thing of course is that this information can easily be charted (I am using kavachart) and it allows simple navigation through different groups of equipments and different views on the equipment.

- WORM from Pascal Obry (see http://www.ada-france.org/ADHERENTS/101100/05-obry.pdf).

A Web server to share bookmarks, this server was using a standard CGI design. To keep session information we were using a GLADE partition. With AWS the design has beeen really simplified, there is no need for a session partition, there is no need to build all CGI as partitions too. GLADE is now used only to handle distributed objects. Indeed WORM is a multi-server system (using RACW) with a register/unregister mechanisme.

Also the server seems to be fastest, there is no more CGI to spawn.

- Internet Currency Trading System at www.actforex.com by Dmitriy Anisimkov

This is a server used to keep historical data about currency trading to build charts of currency prices. The charts viewer part is written in Java and loaded through AWS. This server can be reached on the Internet.

Thanks to all who have reported bugs and have sent us patches.

Dmitriy & Pascal.

Pascal Obry, Team-Ada Member, 45, rue Gabriel Peri, 78114 Magny Les Hameaux, France
http://perso.wanadoo.fr/pascal.obry

AWS Use - SETI@Home Service

From: Ted Dennison <dennison@telepath.com>
Date: Mon, 12 Mar 2001 20:07:01 GMT
Subject: Re: ANNOUNCE: new AWS version (0.9.10)
Newsgroups: comp.lang.ada, fr.comp.lang.ada

> It would be nice if you could also sent us a note if you are using AWS just to know if it is used at all or not :)

I'd like to mention that the upcomming release of the SETI@Home Serivce will be using AWS so that users can check the status of the service and the SETI@Home Client with a web browser. I'd estimate that the SETI@Home Service currently has somewhere over 100 users, probably with an average of around 4 installations (on various machines) per user. Hopefully this new funtionality will be attractive enough to spike that number a bit though. :-)

[See also "SETI Service 1.0 - Creating a NT service using Ada" in AUJ 21.4 (January 2001). -- dc]

From: Ted Dennison <dennison@telepath.com>
Date: Tue, 13 Mar 2001 16:02:40 GMT
Subject: Re: ANN: SETI@Home Service 2.0 Released
Newsgroups: comp.lang.ada

Ted Dennison says... I'm pleased to announce that the SETI@Home Service version 2.0 is now available for dowload [...] This latest version includes the following changes: [...] Client progress and other info can now be served to your web browser during execution. The last item ought to be of particular interest to those running wide-spread SETI farms. Curious how far from completion that machine over in building 20 is? Just point your web broswer there and see. :-)

As the more astute of you may have guessed, this was indeed accomplished with the aid of AWS. So thanks to Pascal and Dmitry for the fine package. I can report that once I figured out how to download all the parts and compile them together, using AWS was a breeze. If you want to see an Ada program dynamicly generating and serving up web pages, download the SETI@Home Service and check it out.

I'd also like to mention that I found a somewhat novel solution to the problem (mentioned here earlier this year) of getting some kind of prominent mention of Ada into the service. Each dynamicly-served web page includes an "Ada Powered" bug on the bottom, with a link to AdaPower.com. :-)

From: Pascal Obry <p.obry@wanadoo.fr>
Date: 14 Mar 2001 09:40:59 +0100
Subject: Re: ANN: SETI@Home Service 2.0 Released
Newsgroups: comp.lang.ada

Next version of AWS [0.9.11, see above -- dc] will have an example on how to launch the the famous (at least for me) runme demo as a service under Windows. This is using the very nice Windows NT Service API developed by Ted.

From: Ted Dennison <dennison@telepath.com>
Date: Wed, 02 May 2001 15:00:25 GMT
Subject: Re: [ANNOUNCE] - New AWS (Ada Web Server) version (0.9.11)
Newsgroups: comp.lang.ada

[Pascal Obry wrote:] - The main demo (runme) can now be launched as a Windows NT/2000 service. This uses Ted Dennison Ada Services library.

I'm glad to see it getting some good use. I wrote it in response to a thread here a few months back. I think its a very good example of how Ada can be used to drasticly simplify system calls that are a total pain in the keister for C and C++ developers. I still intend to make a GMGPL release of it [GMGPL = GNAT Modified GNU Public License (GPL) -- dc] sometime with some of my other bindings, but in the meantime anyone wanting to make use of it can get a copy from the SETI@Home Service sources. It is currently GPL, because I think that license works best in the context of the SETI@Home Service as a whole. However, if the licensing terms won't work for you, contact me and we'll see what arrangements can be made.

> At this stage we feel that AWS is ready to build small to medium Web server. AWS has been reported to work under Windows NT, Linux and FreeBSD 4.1.

A previous version of AWS has been in use for more than a month in the SETI@Home Service, which I think would now qualify as a "small Web Server". [...]

Much kudos go to Pascal and Dmitriy and everyone else involved for developing and maitaining such a useful package. I know what a strain on their free time it must be, but its very much appreciated.

From: Ted Dennison <dennison@telepath.com>
Date: Wed, 02 May 2001 19:38:08 GMT
Subject: Re: [ANNOUNCE] - New AWS (Ada Web Server) version (0.9.11)
Newsgroups: comp.lang.ada

[Pascal Obry wrote:] I think we will include the service API when it is released under GMGPL. Please let us now when it is ready.

I currently have way more projects than time, but keep an eye on this space.

For those unacquainted with what we are talking about, the SETI@Home Service comes with a very small set of thick Win32 bindings. No attempt was made in them to completely cover all of Win32; I just wrote what I needed for the SETI@Home Service. Right now this amounts to a nice facility for creating an NT service, support for dealing with the registry, and support for creating and controlling Win32 processes. I'm looking at possibly adding a package for reading NT/2K "performance stats" (process and task lists, CPU utilization, etc).

From: Ted Dennison <dennison@telepath.com>
Date: Tue, 22 May 2001 13:41:12 GMT
Subject: Re: ada web server setup for win2000 prof
Newsgroups: comp.lang.ada

> [...] is anyone currently using the AWS setup for windows, I would also be very interested in anyone using the linux setup,

The SETI@Home Service uses it on NT and Win2k. The sources are available at http://www.telepath.com/dennison/Ted/SETI/SETI_Service.html

From: Ted Dennison <dennison@telepath.com>
Date: Thu, 24 May 2001 13:57:53 GMT
Subject: Re: [ANNOUNCE] - AWS (Ada Web Server) version 0.10
Newsgroups: comp.lang.ada

Pascal Obry says...- SETI@Home from Ted Dennison. AWS is used as a "plugable" GUI to control the services status.

Actually, it just displays the service's status and the progress of the SETI@Home client(s). There is no control functionality through the web interface (yet). To do control, I'd also have to implement a fair amount of security. SETI@Home is very competitive, and I don't want that sneak in the next cubicle remotely turning off my clients so he can catch up to me in the listings. :-)

SDL_mixer - Binding to Multi-Channel Audio Mixer

From: Antonio F. Vargas <antonio.vargas@clix.pt>
Date: Mon, 26 Feb 2001 21:07:33 +0000
Subject: [ANNOUNCE] binding to multi-channel audio mixer
Newsgroups: comp.lang.ada

Thin Ada binding to SDL_mixer + (sample programs ported to Ada) "playmus" and "playwave".

Adapted from the original SDL_mixer README file: Due to popular demand, here is a simple multi-channel audio mixer. It supports 8 channels of 16 bit stereo audio, plus a single channel of music, mixed by the popular MikMod MOD, Timidity MIDI and SMPEG MP3 libraries.

See the spec. file SDL_mixer.ads and the examples playwave.adb and playmus.adb for documentation on the binding to this mixer library.

The mixer can currently load Microsoft WAVE files as audio samples and can load MIDI files via Timidity and the following music formats via MikMod: .MOD .S3M .IT .XM. It can also load MP3 music using the SMPEG library. [...]

The binding was tested under Linux (TM) and Windows (TM)

Please send feedback.

Antonio Ferreira Vargas,
avargas@adapower.net
http://www.adapower.net/~avargas

AdaSockets Binding

From: Fraser Wilson <fraser@whitelion.org>
Date: 26 Feb 2001 18:20:11 -0800
Subject: Re: UDP
Newsgroups: comp.lang.ada

> Do you know where I can find a package to use UDP protocol with Ada?

You can download Adasockets from http://www.rfc1149.net/devel/adasockets

pyAda - Ada 95 Binding to the Python Interpreter

From: Gerhard Häring <gerhard.nospam@bigfoot.de>
Date: Thu, 01 Mar 2001 03:54:25 +0100
Subject: [ANNOUNCE] pyAda, first public release
Newsgroups: comp.lang.python, comp.lang.ada

I am glad that I can finally announce a first version of pyAda. As the name implies, it's an Ada 95 binding to the Python interpreter. With pyAda, you can write Python extensions in Ada 95, and, much more important, you can also embed a Python interpreter in your Ada applications. The current release is far from finished, but the thin binding works already quite well for me. The thin binding is a one-to-one mapping of the Python -> C interface.

website: http://pyada.sourceforge.net
release: http://download.sourceforge.net/pyada/pyAda-2001-03-01.tar.gz docs: included in release

pyAda currently works for me under (SuSE 7.1) Linux and FreeBSD (4.2). It used to work on win32, too, but currently doesn't. That's to be fixed in the next release. Porting to other Unixen should be fairly straightforward.

pyAda currently depends on GNU make, GNAT 3.13p and Python 2.0 (and DocBook SGML for doc creation). The licence is a BSD variant.

Future plans include bugfixes, code cleanup, and development of a thick binding that renders manual reference counting obsolete. And more ports.

PS: This is my first major Ada project, so I am open to any suggestions for improvement.

From: gerhard.nospam@bigfoot.de (Gerhard Häring)
Date: Sun, 15 Apr 2001 00:32:53 +0200
Subject: [ANNOUNCE] pyAda release 2001-04-15
Newsgroups: comp.lang.ada, comp.lang.python
Summary: pyAda is an Ada binding to the Python interpreter

I'd like to announce a new release of pyAda. pyAda is an Ada 95 binding to the Python interpreter. With pyAda, you can write Python extensions in Ada 95, and, much more important, you can also embed a Python interpreter in your Ada applications. With the current release, the project has stabilized a little more:

- all compiler warnings were fixed
- bindings not needed were removed
- minor bugfixes

In addition to Linux and FreeBSD, pyAda now supports building on Windows and Solaris.

website: http://pyada.sourceforge.net
release:
http://prdownloads.sourceforge.net/pyada/pyAda-2001-04-15.tar.gz docs: docs + examples included in release

The focus of the next release will be on documentation improvements and updating to the soon-to-be-released Python 2.1.

pyAda currently depends on GNU make, GNAT 3.13p and Python 2.0 (and DocBook SGML for doc creation). The license is a BSD variant.

GNADE - GNU Ada Database Environment

From: Michael Erdmann <michael.erdmann@snafu.de>
Date: Mon, 21 May 2001 21:36:17 +0200
*Subject: ** Release of GNU Ada Database Environment ***
Newsgroups: comp.lang.ada, comp.databases.oracle.marketplace, comp.databases.ibm-db2, comp.databases.informix, comp.databases.ingres, de.comp.datenbanken.misc

GNU Ada Database Environment (GNADE) Version 1.1.1

This version 1.1.1 of the GNADE project is available at http://sourceforge.net/projects/gnade for the Linux, Solaris and WindowsNT platforms using the GNU Ada Translator (GNAT).

The GNADE project features:

- Thin bindings to the ODBC interface.
- Embedded ISO/92 SQL Translator generating code on top of the ODBC bindings.
- Native bindings to Postgres and MySQL.
- Installation support for MySQL, Postgres and Mimer Database products.
- Basic documentation in html, postscript and PDF format.

For more information refer to: http://gnade.sourceforge.net/

Ada-related Products

Accel Software Engineering - Message Translation and Validation (MTV) Builder

From: Charles Plinta <cplinta@accelse.com>
Date: Mon, 5 Mar 2001 11:08:34 -0000
Subject: Re: Technical Question
To: team-ada@acm.org

> My idea is to build an ASIS tools to check the points where invalid values enter in the program ("Unchecked_Conversion", 'read, 'input, ...) and to warn the programmer if some of the dangerous fields are used without a 'Valid check. Not so easy to do... Does-it exist?

I'd like to point you to some information in response to your comment "...to build an ASIS tool...". We may already have the foundation upon which you can build.

A few years back we had an SBIR (Small Business Innovative Research) contract with the Air Force. We developed something we called MTV Builder. MTV Builder is a PC-based tool that simplifies message specification and automatically generates Ada95 message translation and validation (MTV) software.

MTV software is a vital component in the integration of a diverse community of disparate systems and devices. Accel’s MTV Builder replaces the traditionally time consuming and error prone manual coding process. MTV Builder provides System Integration professionals with an engineered solution for constructing MTV software quickly, inexpensively, and reliably.

If you visit our web site (www.accelse.com), you can get more information on MTV Builder. There are several white papers, one specifically provide a good overview (http://www.accelse.com/mtv/MTVBld_WhitePaper.pdf) and you can even download a demo of the tool.

Accel Software Engineering, 9 Mellon Road, Export, PA 15632, USA (w) 724-733-8800, (f) 724-733-8820, http://www.accelse.com/

ACT-Europe - XML/Ada 0.5 Suite of Tools

From: Emmanuel Briot <briot@gnat.com>
Date: Thu, 26 Apr 2001 12:19:53 GMT
Subject: [ANNOUNCE] XML/Ada 0.5 released
Newsgroups: comp.lang.ada

We are happy to announce the first public release of the XML/Ada suite of tools.

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 most of the applicable tests of the official XML conformance testsuite.

This library is still considered beta, and is released under the standard GNU Public License (GPL). As usual, it is provided as is, without guarantee or support. We do not recommend using of this package in a commercial application. If you are interested in using a supported version of this library suitable for commercial applications, please contact sales@act-europe.fr.

This software is available freely on the http://libre.act-europe.fr/ web page.

Please contact the author for questions or bug reports.

Emmanuel Briot, briot@act-europe.fr

ARTiSAN Software Tools - Ada Code Generator for UML Models in Real-time Studio

From: "ARTiSAN Software Tools" <newsroom@bizwire.com>
Date: Mon, 19 Mar 2001 10:39:29 +0000
Subject: ARTiSAN Software Tools
To: Ada User Journal News <Dirk.Craeynest@offis.be>

ARTiSAN Software Tools Announces Ada Code Generator for UML Models in Real-time Studio Professional 4.0

Supports Ada 83, Ada 95, SPARK 83, and SPARK 95

Shrivenham, U.K. -- March 16, 2001 -- At the Ada UK Technology Update here today, ARTiSAN Software Tools announced an Ada code generator add-on module to Real-time Studio Professional (RtS) 4.0 that translates UML class models into Ada source code and supports Ada 83, Ada 95, SPARK 83, and SPARK 95.

"ARTiSAN is well known for the strength of its UML-based systems engineering and software engineering support for large teams developing mission critical applications," said S. Caine O’Brien, ARTiSAN president and CEO. "While these teams are replacing older design methodologies with UML, in many cases they are maintaining their commitment to the Ada language and have been asking for a solution to automate the implementation of their design models in Ada."

RtS 4.0 is uniquely designed to allow easy modification of the code generator’s mapping between design artifacts and code, providing the critical flexibility required to tune the code generation strategy to specific needs. Templates for each of the four Ada variants [Ada/SPARK 83/95 -- dc] can be used as delivered or tailored to control how specific UML constructs are mapped to Ada source. The code generator also recognizes additional properties that can be added to the model.

According to Venture Development Corp., Natick, MA, Ada is finding increasing use beyond traditional military applications and is becoming a language of choice for high reliability applications in the transportation, energy, telecommunications and medical industries. Originally written as a formal and well-defined language, the standards and robustness of Ada give it distinct advantages over the popular C++ language.

Embedded software developers have discovered that the use of Ada reduces coding errors and the time to find and fix them. The ARTiSAN Ada Code Generator will be available in June 2001.

For pricing, contact an ARTiSAN sales office; visit www.artisansw.com.

About ARTiSAN Software Tools

ARTiSAN Software Tools provides electronic product development teams with the "fastest path to the right product". The company offers products, services and a process for systems and software modeling to accelerate the development of next-generation real-time systems while ensuring that they meet requirements. ARTiSAN enables engineering teams to visualize, design and validate systems before building them, and simplifies implementation with code generation and software reuse. ARTiSAN is a practical partner offering easy-to-adopt, down-to-earth solutions.

ARTiSAN Software Tools, Inc., founded in March 1997, is privately held with headquarters in Portland, Oregon and Cheltenham, United Kingdom. The company has regional sales offices and distributors throughout the world. For more information, visit www.artisansw.com.

Contact: ARTiSAN Software Tools, Roy C. Webster, 503/245-6200 x230, royw@artisansw.com

DCS Corporation - Target Graphics Toolkit

From: Pritchett, William "Bill" <wpritchett@dcscorp.com>
Date: Tue, 13 Mar 2001 13:29:20 -0500
Subject: Re: Ada95 Graphics
To: team-ada@acm.org

DCS Corporation has a portable embedded graphics solution, Target Graphics Toolkit, which was developed entirely in Ada 95 and currently runs on VxWorks, Linux, Solaris, NT, and LynxOS. It’s basically a portable graphics library that sits on top of a variety of native graphics systems including X11, OpenGL, Win32 and DY4’s RTGS. We also have a screen builder that let’s you layout the "widgets" for the GUI and then generate either Ada95 or C++ code that then makes calls into our API. It has been tested with both GNAT and Green Hills compilers. For additional information you can contact Tim English, 703-683-8430 x485 (tenglish@dcscorp.com).

Bill Pritchett, DCS Corporation

DDC-I - VSM Package for VAX to Solaris Migration

URL: http://www.ddci.com/news/news_vol1num3.html
Subject: DDC-I Online News December 2000 Vol. 1 Issue 3

DDC-I Offers User-Friendly VAX to Solaris Migration

DDC-I’s Very Smooth Migration (VSM) package delivers a low-risk migration path for safety-critical embedded system Ada developers faced with rising VAX/VMS support costs

Phoenix, AZ September 12, 2000 DDC-I, an industry leading provider of real-time embedded system software development tools and services for safety-critical project developers since 1980, today announced the availability of a new Very Smooth Migration (VSM) package, created specifically to support safety-critical software developers decomissioning their venerable DEC VAX hardware recently discontinued by Compaq and moving to the well-established Sun Solaris environment.

The package is designed to ease the transition for organizations replacing their existing VAX-based DDC-I IDE (Integrated Development Environment) with a mature Solaris-based version. Other developers new to DDC-I that are also migrating from VAX/VMS to Sun Solaris have the opportunity to acquire one of DDC-I's robust Ada IDEs, field-proven by customers like Boeing, Ericsson, Honeywell and Mitsubishi. [...]

One of the few contemporary operating systems to rival the unparalleled stability and scalability of the legendary clustered VAX/VMS environment, Sun Solaris in tandem with the mature DDC-I Solaris-hosted IDEs provides a secure migration path that minimizes risk via porting of existing VAX-hosted VMS applications to a new Solaris development platform. [...]

According to Seideman, the VSM package will bundle either the mature and stable DACS (DDC-I Ada Compiler System) or TADS (Tartan Ada Development Systems) IDE for Sun Solaris with a list of choices that includes on-site consulting to assist with rescripting, memory and segment set up, tool adaption, related Ethernet work and UCC upgrading. [...]

DDC-I's DACS is a powerful development environment, constructed around a validated, field-proven Ada compiler. The standard DACS bundle also features an Ada recompiler, symbolic debugger, linker, run-time profiler and system, and library management tools.

The TADS-2000 products (TADS-1750A, TADS-68xxx, and TADS-i960) combine a highly optimizing compiler with modular run-times and selective linking to generate the most compact code available, in addition to performance benefits and classical optimizations specific to target architecture.

"DDC-I is offering the VSM package to its existing IDE users as a cost-effective migration path away from the aging VAX hardware, while we will continue to offer a level of support for existing DDC-I IDE users that choose remain on VAX/VMS," Seideman concludes.

DDC-I - Tartan Ada Development System (TADS) V.5.2.3

URL: http://www.ddci.com/news_vol2num2.shtml
Subject: DDC-I Online News February 2001 Vol. 2 Issue 2

Announcing Release V.5.2.3 of the Tartan Ada Development System (TADS). This actively upgraded product line now offers significant debugger improvements and enhancements. The X-window interface has been upgraded to the Motif standard, and menu customizations have been replaced with toolbar customizations. Overall, the debugger is considerably more robust than in V.5.2.

Summary of New and Modified Features: [see URL above -- dc]

PragSoft - UMLStudio Supports Ada

From: "Pat Rogers" <progers@classwide.com>
Date: Thu, 5 Apr 2001 08:16:41 -0500
Subject: Re: WithClass 2000 UML tool
Newsgroups: comp.lang.ada

[In response to a request for recommendations of UML tools: -- dc]

Have a look at UMLStudio. Inexpensive, free trial, and it has Ada support too. :-) http://www.pragsoft.com/

Rational - Apex Embedded 4.0.0B, Solaris to PowerPC Family

From: "Eddie Glenn" <cav@Rational.Com>
Subject: Rational Apex Embedded, version 4.0.0B, Solaris to PowerPC Family for Rational Exec, Tornado, and LynxOS, is available by FTP
To: <apex-announcements @Rational.Com>

[Extracted from postings on Mon, 12 Mar, and Sat, 14 Apr 2001. -- dc]

Rational is pleased to announce the FTP availability of:

[In all Rational' URLs below, substitute <DOC> by <FTP>/apex_cross/documents/unix, <FTP> by ftp://ftp.rational.com/public, and <platform> by rt, vw, and lx, for the resp. platforms mentioned above. -- dc]

PRODUCT 1: Rational Apex Embedded for Rational Exec

PRODUCT 2: Rational Apex Embedded for Tornado

PRODUCT 3: Rational Apex Embedded for LynxOS VERSION: 4.0.0B PLATFORM: Sun SPARC Solaris to PowerPC Family URL: <FTP>/apex_cross/releases/sol/power/<platform>/4.0.0/

This release is pending Generally Available (GA) status as it goes through the final steps of the manufacturing process. We anticipate that this will be complete within the next 60 days. Once this release reaches GA status, it will be available for shipping. Until then, it is being provided on Rational's FTP server for immediate access.

This product is intended to be compatible with Rational Apex native Solaris 4.0.0B (or later).
[Product 2] is compatible with Tornado II or Tornado AE from WindRiver. Rational Apex Embedded 4.0.0 does not support Tornado 1.0 nor 1.0.1.
[Product 3] is compatible with LynxOS 3.1.0a. LynxOS patch set 010 from LynuxWorks must be installed.
This version of Apex Embedded is not compatible with earlier versions of LynxOS.

To install this product you must first have installed Rational Apex Ada 95/83, version 4.0.0B (or later), SPARC Solaris.

This release requires new 4.0.0 license keys, although the names of the license keys have not changed. License keys issued for previous releases will not work with this version. However, 4.0.0 licenses will work with earlier versions of the product. Please contact Rational Support (support@rational.com) for new licenses.

Release Notes for this release are located at: <DOC>/Rexec/relnote.4.0.dir/rexec_relnote_ppc/rexec_release_noteTOC.html
<DOC>/VxWorks/relnote.4.0.dir/vxworks_relnote_ppc/vxworks_release_noteTOC.html
<DOC>/LynxOS/relnote.4.0.dir/lynx_relnote/lynx_release_noteTOC.html

Rational Apex Embedded LynxOS 4.0.0 Release Note (postscript): <DOC>/LynxOS/relnote.lynx.4.0.ps.Z

There are 3 ways to do the download. You can use the classic method of the UNIX ftp command, use a web browser, or use Rational's rinstall program to do the FTP download in a user-friendly way. <FTP>/.standard.msgs/install_instructions.html

New Features in this release:

* Compiler improvements and Ada runtime improvements which will improve the efficiency of applications built with Apex Embedded. This includes better performing applications as well as applications with smaller foot
prints.
* Integration with ClearCase
* BSP's and models are more convenient to users. Two new features were added to

support this: 'model descriptions' and 'model paths'. A Model description is a textual description of what the model/BSP is (or is used for). This description is displayed by the Apex Embedded model dialog when a user is trying to decide which model to use for a new view. Model paths allow models, created by a user, to show up in the model dialog, even if they do not reside in the Rational Apex installed directory.
* Support for Ada runtime source code releases - support has been added to facilitate the use of runtime source code releases (which are sold separately). This allows users to install an Apex runtime source code release, recompile it, and rebuild all runtime archive libraries. This feature is intended for customers of safety-critical applications who need access to runtime source code.
* General usability improvements.
* Redesigned GUI for managing targets, running applications, and downloading/debugging them on the target.
* Improved integration with TestMate Cross.

New Features for Product 1 in this release:

* Rational Exec microkernel configuration utility. This allows users to configure the Rational Exec microkernel to eliminate services which are not required by the application which reduces the memory needed for the microkernel.
* BSP support for DY-4's DVME-178 & DVME-179 boards
* Local_io support - users can now configure how text_io is physically implemented on the target. Previously, text_io was only implemented via calls to TDM or through the Agilent Technologies' emulation probe. Now users can write device drivers that will be used to implement text_io.
* M.A.R.K. runtime for safety-critical applications is now delivered as part of the Apex Embedded product (instead of being delivered on a separate CD). M.A.R.K. is a separately licensed product. If you have already purchased M.A.R.K., please contact Rational to obtain the license key that unlocks the 4.0.0 version of M.A.R.K.
* Improvements in the integration with the Agilent Technologies' emulation probe, including support for both the E5900A and E5900B versions.

Eddie Glenn, Apex Embedded Product Manager, Rational Software

Rational - Apex Embedded for Rational Exec 3.2.1b

From: "Eddie Glenn" <cav@Rational.Com>
Date: Tue, 17 Apr 2001 16:42:08 -0700
Subject: Rational Apex Embedded for Rational Exec 3.2.1b, Sun SPARC Solaris to RH-32 is available by FTP
To: "Apex Announcements" <apex-announcements@Rational.Com>

PRODUCT: Rational Apex Embedded for Rational Exec VERSION: 3.2.1b PLATFORM: Sun SPARC Solaris to RH-32 URL: <FTP>/apex_cross/releases/sol/rh32/rt/3.2.1b RELEASE NOTE: <DOC>/Rexec/relnote.3.2.dir/rexec_relnote_rh32/rexec_release_noteTOC.html

This is an update release. Unless you have been instructed by a Rational account representative to install this version of the product, you should install the generally available (GA) version. Follow this link for Rational Apex Embedded for Rational Exec download and installation instructions. [...] <FTP>/.standard.msgs/install_instructions.html

Release Notes and Install Guide are located here:
<DOC>/Rexec/relnote.3.2.dir/rexec_relnote/rexec_release_noteTOC.html
<DOC>/Rexec/relnote.3.2.dir/rexec_relnote_rh32/rexec_release_noteTOC.html
<DOC>/Rexec/relnote.rexec.rh32.3.2.ps.Z

Rational - Ada Products for Compaq Tru64

From: "Greg Bek" <gab@Rational.Com>
Subject: Rational Apex 4.0.0c, Ada Analyzer 4.0.1, AXI 4.1.9 for Compaq Tru64 is available by FTP
To: "Apex Announcements" <apex-announcements@Rational.Com>

[Extracted from postings on Wed, 9 May 2001. -- dc]

PRODUCT: Rational Apex VERSION: 4.0.0c URL: <FTP>/apex/releases/axp/apex.4.0.0c

PRODUCT: Ada Analyzer VERSION: 4.0.1 URL: <FTP>/ada_analyzer/releases/axp/ada_analyzer.4.0.1

PRODUCT: AXI (Ada/X Interface) VERSION: 4.1.9 URL: <FTP>/axi/releases/axp/axi.4.1.9

PLATFORM: Compaq Tru64

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

[The Apex] release requires new license keys as the version in the license has been incremented to 4.000. Please contact Rational Technical Support for obtaining license upgrades.

Release Notes and Install Guide are located here: <FTP>/apex/documents/unix/release_note.4.0.0c.dir/release_noteTOC.html
<FTP>/apex/documents/unix/release_note.4.0.0c.ps.Z
<FTP>/apex/documents/unix/install_guide.4.0.0c.dir/igTOC.html
<FTP>/apex/documents/unix/install_guide.4.0.0c.ps.Z
<FTP>/ada_analyzer/documents/unix/release_note.4.0.1.dir/AA_Release_Note.4.0.html
<FTP>/ada_analyzer/documents/unix/release_note.4.0.1.ps.Z
<FTP>/axi/documents/unix/axi_relnotes.4.1.9.axp.dir/relnotesTOC.html
<FTP>/axi/documents/unix/axi_relnotes.4.1.9.axp.ps.Z

Ada and Linux

New Linux packages on www.gnuada.org

From: Jürgen Pfeifer <juergen.pfeifer@gmx.net>
Date: Mon, 30 Apr 2001 10:30:31 +0200
Subject: RPM update
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

I've updated the RPM collection for Linux on www.gnuada.org/rpms.html

- JGNAT (minor update) - GtkAda (new version) - GVD (new version) - Tash (new version) - Rapid (recompile against new Tash version) - Booch componentes (new version) - XmlAda (new package)

From: Jürgen Pfeifer <juergen.pfeifer@gmx.net>
Date: Sat, 5 May 2001 19:42:25 +0200
Subject: RPM update
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

I've updated the RPM collection on http://www.gnuada.org/alt.html

There are two new packages:
- adasockets. Binding to BSD sockets
- aws. This is an Ada WEB Server component

Two packages are updated:
- adaasl. This is just a rename of the former "asl" package. SuSE already has a package with this name (a macro assembler). So I renamed to avoid naming conflicts. Please uninstall the previous packages with the command "rpm -e asl asl-runtime" and install the renamed ones.
- rapid. This is only a fix in the build scripts. You should download the new version only if you build from the source RPMs yourself.

From: Jürgen Pfeifer <juergen.pfeifer@gmx.net>
Date: Mon, 14 May 2001 16:56:14 +0200
Subject: New RPM releases
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

Yesterday I uploaded some updates and new packages to www.gnuada.org

- Release 6 of the GNAT packages. These have minor corrections only, mainly for people using a C-Shell as their login shell.

- Release 7 of JGNAT. This just repackages the documentation. The info files are now copied to the ALT info directory.

- Release 1 of Version 1.1.0 of GNADE. This is a new package. GNADE stands for GNu Ada Database Environment. It contains a thin ODBC binding and an embedded SQL compiler for Ada, following closely the ISO92 SQL Standard.

From: Jürgen Pfeifer <juergen.pfeifer@gmx.net>
Date: Mon, 28 May 2001 01:55:34 +0200
Subject: RPM updates
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

I've uploaded two updated packages to http://www.gnuada.org/rpms313p.html

- Version 1.1.2 of GNADE, the GNU Ada Database Environment http://www.gnuada.org/rpms313p.html#GNADE

- Version 0.10 of AWS, the Ada Web Server component. http://www.gnuada.org/rpms313p.html#AWS

Ada and Microsoft

Dynamic Windows Graphics - Hydrogen Orbitals

From: Tom Moran <tmoran@acm.org>
Date: Fri, 20 Apr 2001 01:14:51 GMT
Subject: Ann: dynamic Windows graphics (hydrogen orbitals)
To: team-ada@acm.org

Been wondering since high school what those s-p-d-f orbitals beyond 1s and 2p really look like? Look for "Orbitals" on www.adapower.com and you'll find a demo of dynamic Windows graphics, Claw style, that lets you see higher electron probability density orbitals of the Hydrogen atom. There is a Windows executable as well as source which can be compiled and linked with the introductory version of Claw using Gnat, Janus, ObjectAda, or Rational compilers. There is also a very simple demo, Tube, which is pretty (especially if you have blue-red 3D glasses handy), and which exposes more simply the underlying reusable cloud display object.

[The direct link to the demo is at: http://www.adapower.com/os/orbitals.html -- dc]

References to Publications

Benjamin Woolley - The Bride of Science: Romance, Reason, and Byron's Daughter

From: "W. Wesley Groleau x4923" <wwgrol@ftw.rsc.raytheon.com>
Date: Fri, 2 Mar 2001 09:42:58 -0500
Subject: Re: Another book on Lady Ada?
To: team-ada@acm.org

[Referring to a book review on www.newyorker.com : -- dc]

> Another myth debunked?

If you refer to the "myth" that Ada was the first programmer, Scientific American pointed out a few years ago that Babbage wrote programs before Ada did. His were much simpler than hers, however.

[See also "Articles on Ada, the person" in AUJ 21.4 (January 2001). -- dc]

And how do you define programming? Is it only "programming" if branching is possible? Or was it "programming" to create a control card for Jacquard's loom? Still, that book sounds interesting....

From: "Ann S. Brandon" <abrandon@sover.net>
Date: Fri, 2 Mar 2001 13:41:11 -0500
Subject: Re: Another book on Lady Ada?
To: team-ada@acm.org

Benjamin Woolley's "The Bride of Science: Romance, Reason, and Byron's Daughter" certainly dispels the myth that Ada was a mathematician, but I am not sure she should therefore be equated with Lisa-Marie Presley.

I think Woolley sets up a straw woman and knocks her down. Those who have called Ada of Lovelace the first computer programmer have always been wrong. Either Joseph-Marie Jacquard or Babbage can claim that title.

Ada's historical claim is as the first published computer programmer. After its May 1999 profile on Ada, I wrote a letter to "Scientific American" taking the authors to task for simply calling her a "prophetess". Though the magazine did not publish my letter, they did write "first published computer programmer" in the caption under her portrait that appeared in the letters column, so they must have agreed with me.

Ada was also the first to foretell the computer's future activities in drawing and creating music. Yet Woolley and The New Yorker's reviewer, Jim Holt, instead emphasize that she failed to foretell artificial intelligence. She also didn't foretell how miserably computers predict the weather, but neither did all the fathers and mothers of the modern computer age. Babbage's asking Ada to write the paper, according to Woolley, is comparable to "nominating Lisa-Marie Presley to annotate a study of quantum computation". This is a tabloid headline, not research. Ada's reputation was one of being educated in math, whereas Lisa-Marie is educated in buying clothes. That Babbage did most of the math for Ada in the paper is not unheard of in scientific circles. After all, Lise Meitner did all the math for Otto Hahn's Nobel Prize in Chemistry. Should we compare him to Elvis?

Ada Essentials: Overview, Examples and Glossary

From: bscrawford@aol.com (BSCrawford)
Date: 19 Mar 2001 15:05:48 GMT
Subject: LearnAda web site changes
Newsgroups: comp.lang.ada

Announcing three recent changes at www.LearnAda.com

1. Ada in a Nutshell A section of the site entitled "Ada in a Nutshell" consists of seven pages excerpted from the HTML version of "Ada Essentials: Overview, Examples and Glossary". Two of these pages provide a high-level, graphical overview of the entire language: 4 kinds of modules (program units) and 3 ways of connecting modules (withing relationships, nesting relationships, and parent/child relationships). I would like to invite comments on this section.

2. Book Reviews A new page at the site contains two reviews of "Ada Essentials: Overview, Examples and Glossary" written by professional software engineers. One is a brief review written by Jerry Petrey of Ratheon. The other, more extensive, review is by Marin D. Condic of Quadrus Corp.

3. Change of Source for PDF Version The PDF or "eBook" version of "Ada Essentials: Overview, Examples and Glossary" can no longer be obtained from FatBrain.com. It can now be ordered directly from the www.LearnAda.com site. FatBrain was acquired by Barnes and Noble, which has changed their "business model". They are now interested only in eBooks that have best-seller potential. (They have not yet realized that Ada texts have this potential. :-)

Bard S. Crawford, Author of "Ada Essentials: Overview, Examples and Glossary," a compact volume available in three forms: printed book, pdf file, and a collection of browser-based web pages. See http://www.learnada.com

Stage Harbor Software, 9 Patriots Drive, Lexington, MA, 02420 USA, bard@learnada.com, 781-862-3613

CrossTalk Article - Ada in the 21st Century

From: "Ken Garlington" <Ken.Garlington@computer.org>
Date: Tue, 20 Mar 2001 01:24:01 GMT
Subject: CrossTalk article on Ada
Newsgroups: comp.lang.ada

See "Ada in the 21st Century" (Benjamin M. Brosgol, Ada Core Technologies) in the March 2001 issue of CrossTalk magazine. A web version is available at

http://www.stsc.hill.af.mil/CrossTalk/2001/mar/mar01ind.asp

The Letters to the Editor section also contains several observations regarding the previous issue's article "Is Ada Dead or Alive Within the Weapons System World?" (Reifer et. al.). Contributors are: Dennis Ludwig (Electronic Engineer, Warner Roberts-ALC), Tom Moran (Decision Aids), Robert C. Leif, Ph.D. (Newport Instruments, Ada_Med Division).

Ada Book Review at CanonicalTomes.org

From: "Steven G." <superant-steve@superant.com>
Date: Tue, 20 Mar 2001 05:11:55 GMT
Subject: Ada book review at CanonicalTomes.org
Newsgroups: comp.lang.ada

I have entered Barnes at http://www.canonicaltomes.org/ as THE Ada book to read. The category is: Ada under Programming Languages under Computers & Technology. People should vote for it or suggest other titles.

Ada in General Online Documents

From: Craig Spannring <Craig.Spannring@aedinc.net>
Date: Mon, 26 Mar 2001 14:47:34 -0700
Subject: Ada in the Winsock FAQ
To: team-ada@acm.org

The Winsock FAQ now has a mention of Ada. It's a very minor mention, but I guess minor is better than nothing.

http://www.cyberport.com/~tangent/programming/winsock/resources/samples.html

From: Jacob Sparre Andersen <sparre@nbi.dk>
Date: Tue, 27 Mar 2001 12:45:39 +0200
Subject: Re: Ada in the Winsock FAQ
To: team-ada@acm.org

Making sure people also see Ada mentioned when they are looking for information will hopefully repel Ada's image as a "dead" language.

Which reminds me...

<plug> Ada is among the covered language in the Danish introductory book to Linux programming, "Linux - Friheden til at programmere", which can be found at http://www.sslug.dk/linuxbog/#program </plug>

Ada Mentioned in Daemon News Ezine

From: Robin P Reagan <robin-blackhole@reagans.org>
Date: Thu, 10 May 2001 20:06:46 -0600
Subject: Ada mentioned in Daemonnews.
To: team-ada@acm.org

Ada got a small but favorable mention in the following article on the Daemon News Ezine. http://ezine.daemonnews.org/200105/dadvocate.html

[And from another message: -- dc]

don't know if anyone else has seen this, but there's an interesting article in this months Deamon News about safe computing where Ada get's some nice attention. [...]

[A small extract follows, you should read the whole (copyrighted) article online: -- dc]

"The Dæmons Advocate: Old Reliable" by Wes Peters.

Recently a friend sent me an email about Microsoft beginning to create "fault tolerant" software. I quipped back "Why would they want to do that, when they've spent two decades developing fault tolerant customers?" [...]

A lot of pioneering work in the area of safe languages was done by a group commissioned by the US Air Force in the late 70s and early 80s. Their ultimate product was a specification for a new computer language called Ada. Depending on your age, you may or may not have heard much about Ada as a language. It is another language that deserves some study. The research material and the commentary on the safety features of the language are some of the most concise treatises on the concept of safe computing languages you will find. For historical background on the development of Ada, consult the Ada Home History of Ada page. [http://www.adahome.com/History/ -- dc]

David A. Wheeler of the Institute for Defense Analyses has written a study that compares Ada95, Java, C, and C++ to the original Steelman draft specification. If you are curious about the relative merits of each of these languages, this study is an interesting read. [http://www.adahome.com/History/Steelman/steeltab.htm -- dc]

From: Wes Groleau <wwgrol@ftw.rsc.raytheon.com>
Date: Fri, 11 May 2001 09:36:06 -0500
Organization: Raytheon Company
Subject: For Wes Peters
To: team-ada@acm.org

Excellent balanced article. There are those that claim Ada will solve ALL the world's problems. This, I will admit, is inaccurate, and possibly even harmful. But even worse is all the unfounded hype about Java. [...]

In regard to your question about "27 new features" note that they already had to change the language itself once to eliminate a security hole. They also had to change the JVM to eliminate a hole that could be exploited by using a j-Code assembler rather than a Java compiler. Both of these changes occurred after JDK 1.0 was released.

Java deserves credit for getting hundreds of thousands of programmers out of the dark ages of C & C++ into the dim light of common sense. However, Java's so-called designers do not deserve rave reviews for eliminating from C only the worst features, i.e, the features that (in spite of their experience with C) were threatening the schedule of their current (at the time) project.

Because of their experience, they had few problems with other undesirable C features, and so they left them in to continue to confound beginners. Moreover, if they had done their homework, they may not have omitted as "unsafe" features that Ada and other languages had already shown how to do "safely".

Java, in other words, rather than being carefully engineered as an effective general purpose tool, was a hasty hack to rescue a project in trouble. That it was not carefully engineered is shown by some of its numerous changes since first release. Not the ones that were valuable enhancements, but the ones that were mandatory to plug unforeseen holes in Java's over-hyped "security".

Java just happened to catch the attention of some clever marketers who recognized that a lot of other projects are in the same kind of trouble. These marketers are not only clever, they are a bit dishonest in promoting the Java language itself as if inseparable from JVM portability (which has already been exploited by compilers and interpreters of Ada, Eiffel, Cobol, Smalltalk, and at least twenty other languages) and Java's admittedly extensive (though sometimes poorly engineered) suite of re-usable APIs (which can be called from any of the languages just hinted at, and via JNI from most other compilers).

Again, Java is a welcome improvement over the two previous dominant languages, but I fear most programmers will stagnate in Java (and unportable variations thereof) just as they did for decades in C.

Ada Inside

USA - Sikorsky's Control Display Unit (CDU) Redesign

URL: http://www.ddci.com/news_vol2num2.shtml
Subject: DDC-I Online News February 2001 Vol. 2 Issue 2

DACS Helps Sikorsky Standardize "Hawk Eyes"

Sikorsky's Control Display Unit redesign for the Black Hawk offers a clear view of DDC-I's important supporting role in the reduction of embedded system development costs

[...] Stockreef highlights a current project, involving a major upgrade of the cockpit visual display systems, originally designed in 1973, as the clearest indicator of where Sikorsky is heading in terms of reducing overall development costs via standardization.

"We got to a point as we were adding functionality to the old system where we said, these boxes are too heavy, they cost too much, and were running out of processor horsepower," he says.

With an eye to the bottom line, Stockreefs team put together a business plan proposing a completely new breed of display and data input device called a CDU (Control Display Unit) for Black Hawk, alongside an entirely new MFD (Multi-Function Display). The numbers clearly indicated that for the cost reduction on the next program alone they would be able to fund all the non-recurring engineering costs.

"Essentially, the CDU is a data input device with a little glass window and alphanumeric keypads the pilot uses to enter data into the system. With a sign-off from the top brass, we chucked all of the old system boxes and brought in completely new hardware being built by Rockwell-Collins," he says.

Because the proposed CDU was built around an 80486-based processor, which Sikorsky used in other programs, his team decided the most practical and cost-effective approach was to reuse available legacy code and port it to the new processor using their existing DDC-I Ada Compiler System (DACS) as the software development environment.

The results have been as solid as predicted, reducing the total number of system boxes in the helicopter from twelve to six, generating a major real estate and weight savings. The programs success is also spurring increased movement toward system standardization across most of the Sikorsky product line.

"For the present, the new display system is going into multiple platforms in our international product line, and we're evaluating all of our helicopters. While Comanche probably won't see it, all of the international helicopters are being discussed, possibly even the S-92 and the MH-53, but there is no firm plan right now. For the moment, the main effort is on the international Naval Hawk and Black Hawk," says Stockreef. [...]

Stockreef asserts that the diverse needs of a growing list of international customers only increases the need to standardize parts and processes wherever possible. For the CDU, development partner Rockwell-Collins is writing the primary flight display software, while Stockreefs team serves as the overall project integrator and lead software developer.

Using the DACS compiler specifically for the 80486 in the CDU, they are also using vendor tools to write embedded code for a pair of TI 320C31 DSPs in a dual processor configuration, one handling flight management and the other mission management. While C++ is used, most of the embedded system code is in Ada, which he sees as a large advantage when it comes to porting the code to a different processor.

"They're re-hosting some of the software we have, including DACS for the CDU, and we're doing a work-split with them. We'll say, you do this function, and they send us a drop every two or three weeks. We check it, do the build, and do the integration testing here," he says.

[...] Meanwhile, they are undoubtedly observing the experience of the Sikorsky team participating in the innovative "round table" development process for the Comanche helicopter. DDC-I is also a member of the Comanche group, contributing a customized version of the DACS environment aptly named the Comanche ACS, which has been redesigned to provide first class integrated project management tools and utilities for the multiple vendor teams. Managed by Boeing, the Comanche software program is one of the most extensive development projects for safety-critical real-time embedded systems ever undertaken. [...]

Speaking about DDC-I's role and the benefits of DACS environment used for the CDU program, Stockreef says, "The biggest cost benefit is that we're very familiar with it. Its robust, and has been used on several programs so we've got plenty of hands-on experience. We've been able to reuse a lot of legacy code." [...]

Australia - Satellite Avionics of the FEDSAT Project

From: Alan and Carmel Brain <aebrain@dynamite.com.au>
Date: Fri, 16 Mar 2001 10:26:23 +1100
Subject: Re: [OT]Satellites for Dummies
To: team-ada@acm.org

[...] for the Satellite Avionics of the FEDSAT project, Australia's first homegrown Satellite in 30 years (Hurray for us!). Naturally it's being done on the proverbial shoestring. [...]

Thank God it's being done in Ada-95, so many errors normally found in Software (exceeding table size etc) won't be in. [...]

[From a later message: -- dc]

> At least you have the luxury of starting from scratch, and therefore choosing the best language for the job. I have to cope with all the old guard, who really want to think in terms of analog amplifiers rather than software. Our "newest" idea is to use C++!

The language was a given from the start. I'm coming in after the previous contractors ran into difficulty. Too many Computer Scientists, not enough Software Engineers. But that, as they say, is another story.

As for C++... Oh dear.

[And finally: -- dc]

> I echo [...]'s sentiments (envy? ;) for being in an organization that is using Ada. Is that an uphill battle, a straightforward analysis & decision, or is it to the point where Ada is just assumed to be the best language for the problem at hand?

The last. It's avionics. It's real-time. Failure isn't an option. So it's Ada-95.

USA - Intelligent Vehicles: New Big Niche for Ada?

From: "Thomas A. Panfil" <t.panfil@gte.net>
Date: Tue, 10 Apr 2001 23:33:12 -0400
Subject: Intelligent Vehicles -- New Big Niche for Ada?
To: team-ada@acm.org

The Washington DC/Northern VA Chapter of the IEEE/Microwave Theory and Techniques Society, has a theme of "Microwaves and RF for the Intelligent Vehicle" for this academic year's presentations. [...]

For more info, see URL: http://www.ewh.ieee.org/r2/mtt-wnva/

[...] From what I've seen on the related web sites, much has been done in the areas of sensors and system concepts, but little is said about software implementation plans. It looks like this could be a huge new niche for Ada. Check out the links. Any comments?

Tom Panfil, Treasurer, Baltimore SIGAda, http://www.jhuapl.edu/sigada/

From: Michael Feldman <mfeldman@seas.gwu.edu>
Date: Thu, 12 Apr 2001 11:06:58 -0400

Subject: Re: Intelligent Vehicles -- New Big Niche for Ada?
To: team-ada@acm.org

[On to the question "Isn't there a robotic vehicle company in Colorado that uses Ada?": -- dc]

> Yes, there is. James Rogers from Omnitech Robots gave a presentation at ASEET '99.

The Ada project list has an item for this, in the military section. (Their applications have been mostly military, though there are obvious civilian possibilities too.) There's a link from there to the Omnitech website.

http://www.seas.gwu.edu/~mfeldman/ada-project-summary.html

[See also "USA / Omnitech - Robotic Control Systems" in AUJ 22.1 (March 2001). -- dc]

From: "Brakewood, Marcus" <x12479@exmail.usma.army.mil>
Date: Fri, 13 Apr 2001 01:04:40 -0400
Subject: Re: Intelligent Vehicles -- New Big Niche for Ada?
To: team-ada@acm.org

[...] Well, a group of us at the USMA are still trying to finish our intelligent vehicle using Ada. We have:

1) A dual pentium III industrial computer.

2) 2 pulnix video cameras and a sick optics laser as sensors.

3) A GALIL motion control card with encoders.

4) A frame with all the above hardware mounted (including the motors, batteries, and electrical bus).

Note that interfaces were made to our hardware using Ada.

The wheels of the vehicle have moved (using Ada) and 2D representations of lines (similar to traffic lines) have been created from camera pictures (using Ada). We are currently fixing a few interface bugs before going outside to test our "Intelligent" algorithm. This algorithm is pretty much just an implementation of Dykstra's algorithm with a weighted graph (a 2D depiction of the world), with weights based on obstacles detected with the sensors. The algorithm found the best path on paper, but....

An external website should be coming within the next few months.

Hopefully we will be able to finish this year and you can probably expect intelligent vehicles to continue to be developed at USMA using Ada for years to come (we are the first group to attempt it here).

Cadet Marc Brakewood

USA - Defense Related Programs

From: "Marc A. Criley" <marccriley@earthlink.net>
Date: Thu, 12 Apr 2001 11:32:02 -0400
Subject: Re: Ada recommendation
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

[About http://www.seas.gwu.edu/~mfeldman/ada-project-summary.html -- dc]

> I saw the C-17 project. I want to verify that it is in Ada. Is there anywhere I can go to verify it.

When I was job hunting last year, an Ada-oriented recruiter asked me if I was interested in working for AAI in South Carolina, since they were doing the C-17 flight simulator in Ada, and he'd been retained by them to find developers. [....]

Marc A. Criley, Senior Staff Engineer, Quadrus Corporation, www.quadruscorp.com

From: Michael Feldman <mfeldman@seas.gwu.edu>
Date: Fri, 13 Apr 2001 16:45:51 -0400 (EDT)
Subject: Re: Ada recommendation
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

Unfortunately you'll have to take my word for this one. As in so many of these cases, I receive tips from people close to the projects, most of whom say explicitly that I'm not to use their name. Suffice it to say I received the following, in January 2000, from one of my sources, who works for one of the large (VERY large) DoD contractors. He is a credible source but must remain nameless.

(I hate this cloak-and-dagger "anonymous source" stuff, but without playing that game we would never have a list as complete as this one.)

[begin quote]

Since you are now listing Defense related programs, please add the following from ... (sorry, but company rules restrict me from providing any details):

AWACS - NATO Midterm Upgrade; B-1B avionics (converted from JOVIAL); WedgeTail - a 737 airborne early warning system for Australia; Nimrod - upgrade for the RAF; X-32 - Joint Strike Fighter prototype (first flight this Spring); International Space Station - command & control & data handling systems; Comanche and Apache helicopters F-15, F-18, C-17

[end quote]

For a good guess at the contractor, ask yourself who builds that 737.:-)

UK / USA / France - Financial Applications

From: Dr Adrian Wrigley <amtw@linuxchip.demon.co.uk>
Date: Thu, 01 Mar 2001 00:17:52 -0800
Subject: Re: Problems with large records (GNAT) [continued]
Newsgroups: comp.lang.ada

[Description deleted of a problem 'mmap'ing a very large Oracle_T type, modeling stock prices for trading days. -- dc]

I then want to store Oracle_T representing 3000 stocks over seven years in a single file for efficient access by various utility programs.

If I increase the number of stocks to 5000, things break and significant changes are necessary because the 256M limit is exceeded. Sometimes, the wrong answers are produced, which is particularly worrying.

From: tmoran@acm.org
Date: Thu, 01 Mar 2001 07:00:26 GMT
Subject: Re: Problems with large records (GNAT) [continued]
Newsgroups: comp.lang.ada

[...] sometimes it's better to improve the algorithm than the hardware.

I worked on a commercial system (Technical Tools Co.) with stock (and commodity) historical data of this type, using DPMI DOS and Windows 3.1 (and Ada 9X). For historical data, stock prices can be 16 bit fixed point with delta of 1/8, rather than 32 bit floats (excluding Berkshire-Hathaway). Even with prices in pennys nowadays, 24 bits should be quite enough for a stock price. Similarly, a 24 bit Volume (16 million shares of one stock traded in one day) should be normally be adequate, perhaps with an exception list for anything that doesn't fit. A sixteen bit fixed point value, with suitable delta, should be fine for holding the split correction, or 24 bits if you really want to allow for even the most bizarre changes. I don't know what kind of processing you are doing, but usually one processes a small number of complete time series, or the complete market for just a few days, so only a few rows or columns of the complete matrix need be in RAM at any one time.

From: Dr Adrian Wrigley <amtw@linuxchip.demon.co.uk>
Date: Thu, 01 Mar 2001 13:52:28 -0800
Subject: Re: Problems with large records (GNAT) [continued]
Newsgroups: comp.lang.ada

[...] I decided that 16 bits was inadequate. Even with prices in the range $0.05 to $500, you need 20 bits to accommodate a delta representing 1% at the bottom end. Companies that have had a lot of splits and dividends in their history have very small prices back in the '70s. Perhaps a

16 bit logarithm of the share price would be OK. (and even speed up volatility calculations!)

With volume, I think that really needs to better than 32 bit range. Once you start to calculate weekly or monthly volumes, quite a number of companies exceed 2**32 shares. (and in some countries, they even trade fractional shares routinely). Maybe you've seen the WWW sites of historic data that show Intel's monthly share volume as things like "-1518500200 shares". I mentioned this problem to Yahoo nearly a year ago, but they haven't fixed it.

When it comes down to it, it is a matter of confidence and simplicity. Fixed point for this wide ranging data doesn't give me the confidence I want from a (mission critical) financial application. I hadn't thought of using 24 bit values, and I think they would not be worthwhile here given the issues involved.

> [...] usually one processes a small number of complete time series, or the complete market for just a few days, so only a few rows or columns of the complete matrix need be in RAM at any one time.

That's why I want a very fast data access method... I want to scan all the stocks over all the times. Sometimes I access the data sparsely as well. With mmap, the data from one invocation to another remain in RAM, and can be completely scanned in only a few seconds. Maybe someday there will be a standard persistent object store package in the Ada standard. Loading data from files into RAM tends to be amazingly slow, when the file and the in-memory representation are both as big as the physical memory - and my machine has no free memory slots :(

From: tmoran@acm.org
Date: Thu, 01 Mar 2001 19:32:05 GMT
Subject: Re: Problems with large records (GNAT) [continued]
Newsgroups: comp.lang.ada

> Fixed point for this wide ranging data doesn't give me the confidence I want from a (mission critical) financial application.

Of course we were in a "minimize storage" mode for speed and because customers downloaded the data over 14,400 baud modems. So internal computation could use large, or even floating point, variables, but stored data was as compact as possible. For instance, the original prices were stored in 16 bits scaled, but were converted, then multiplied by a "split factor", for computation. Similarly for aggregated volumes. Many customers did very trivial arithmetic, but would be unhappy over rounding errors, so float (especially for commodities with non power-of-two scale factors) was undesirable for the raw data.

From: Laurent Guerby <guerby@acm.org>
Date: 01 Mar 2001 20:38:55 +0100
Subject: Re: Problems with large records (GNAT) [continued]
Newsgroups: comp.lang.ada

[...] Nice to see we're not alone using Ada for critical applications in the financial world ;-). [...]

Anyway, here I would probably want a few things: - control the layout of the data to match computational access, if you process your serie along time or stock. - lazy loading (ie not mmap'ing the whole thing at once).

To do so, I wouldn't put all the things in the record, I would use access types hidden behind proper abstraction. [...] Note that if you compute historical volatilities or correlations, you will probably be memory-bound performance wise (algorithm complexity proportional to data size, with a small enough computation factor so that memory access becomes dominant).

Indirect Information on Ada Usage

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

From: "Andrew Kirk" <andrew@tekwork.com>
Date: Thu, 08 Mar 2001 22:52:15 GMT
Subject: C++ Embedded Software Engineer - Phoenix, AZ
Newsgroups: az.jobs,comp.lang.ada

Position: Embedded Software Engineer
Term: Permanent

Excellent Full-time Opportunity in Phoenix, AZ

[...] This position involves development of programming tools such as compilers, debuggers, and test tools. You will be working with programming languages such as Ada, C/C++ and Jovial. Test methods for safety critical systems & operating systems/kernels. Intel and RISC architectures. Candidates must be team players with strong academic backgrounds in computer science.

Since 1980, our client has been developing programming tools for embedded, real-time, safety-critical applications. Current development consists of compilers, real-time operating systems, linkers, debuggers, graphical user interfaces, test tools etc.

From: "Andre Pitre, CPUSOFT" <andre@cpusoft.com>
Date: Wed, 28 Mar 2001 21:02:47 GMT
Subject: need help with mil-std-1553
Newsgroups: comp.lang.ada

I need a software engineer who can help me with a project in Montreal. This person should have a good experience in designing mil-std-1553 bus controller. [...]

From: kepps@technisource.com
Date: Thu, 29 Mar 2001 15:17:11 -0600
Subject: Survivor(Ada)!!!!!!
To: team-ada@acm.org

[...] A Systems Engineer with an Ada background, who is very diverse, and able to operate in a multi tasking environment. This person should be able to write Ada code on a Unix platform. The individual should also have some experience with one of the Rational Suite Tools. A security clearance is a plus, or they should have a clean background where they can be cleared! [...]

From: "James A. Krzyzanowski" <James_A_Krzyzanowski@raytheon.com>
Date: Wed, 04 Apr 2001 16:58:51 -0500
Organization: Raytheon Company
Subject: Re: Why do so many companies use Apex?
Newsgroups: comp.lang.ada

[...] We staff about 100 software engineers for over 1.3 million SLOC. We call C code. We interface with Java. [...]

James A. Krzyzanowski, Staff Software Engineer, AFATDS, Raytheon Systems Company, Fort Wayne, IN 46808

From: Francisco Javier Loma Daza <Francisco.Loma@isotrol.com>
Date: 17 Apr 2001 10:53:33 +0200
Subject: Re: Opengl bindings for Gnat
Newsgroups: comp.lang.ada

[...] I am working on a radiosity program. It gives the solution using OpenGL. I use gnat, gtkada and gtkada-opengl, but there are no unique OpenGL bindings, so I have to write an intermediate package with two bodies, one for Linux OpenGL, and other for Win32.OpenGL. [...]

From: "Hutchison, Neil A" <neil.hutchison@baesystems.com>
Date: Fri, 20 Apr 2001 08:39:38 -0700
Subject: Ada Software Engineers/BAE Systems
To: team-ada@acm.org

[...] I have opportunities for Software Design Engineers to develop realtime avionic controls in either Ada, C, or C++. Our realtime operating system is an ARINC 653 compliant operating systems being developed to DOD-178B level A. Ideally, we would like someone with some embedded experience in the Motorola 680x0 or PowerPC family, or the Intel Pentium family.

We have opportunities in Johnson City, NY and Santa Monica, CA.

BAE SYSTEMS, a transnational systems, defense and aerospace company, is located in nine countries with over 100,000 employees! The Controls business in Santa Monica has a growing workforce of about 800 employees and delivers state-of-the-art avionics to commercial and defense customers. Johnson City, NY has over 1700 employees. Our Controls products fly on the Boeing 757 and 767 plus the B-2 Stealth bomber. [...]

From: "Brian J. Thibodeau" <bthibodeau@itnet.judge.com>
Date: Thu, 26 Apr 2001 11:43:53 -0400
Subject: Ada Software Engineering Opportunity / Binghamton, NY
To: team-ada@acm.org

I have the following contract opportunity immediately available in NY.

Description: Requirements for a senior (5 years or more) software engineer with the following experience. 1. Current software development on embedded systems in a structured environment. 2. A minimum of 5 years experience with development in the Ada language.

Project involves designing and coding modifications, peer reviews and unit testing for real time embedded avionic engine control programs. This Full Authority Engine Control software will be developed using Ada on a Unix platform.

Duration: 6 Months +

From: kepps@technisource.com
Date: Fri, 27 Apr 2001 22:48:41 -0500
Subject: Survivor Part 2 (Ada edition)!!!!!!
To: team-ada@acm.org

[...]

Mission 1: (1st position) I need a software engineer with a BS Degree in Engineering, Physics or Math. Must have a secret clearance. Experience in developing military applications w/ Ada, at least 5 years of experience. Experience with the Rational tools(Apex, Rose). Corba experience is also a large plus. OOD and OOA experience is required.

Mission 2: (2nd position) I need an engineer experienced in developing SW military applications in Ada95, at least 5 years of experience. Experience with tactical communication protocols (eg., TADIL-J), and/or radar knowledge. This position would be to develop BM/C3I communications.

These missions will take place in Huntsville, Alabama. [...]

From: "Qualicum.com Limited" <Gerry@qualicum.com>
Date: Mon, 30 Apr 2001 18:09:30 GMT
Subject: Senior Ada Programmer (Contract)
Newsgroups: comp.lang.ada

As a key member of the software development team, you will develop our Client's products and build innovative, high-performance display systems for our customers. You are a highly motivated software professional who enjoys working in a challenging, fast-paced environment.

In addition to being a strong Ada programmer who values a well defined software engineering process, you have experience in several of the following areas: User Interface Management Systems (such as InterMAPhics, DataViews, or SL-GMS, VAPS), User Interface Design and Development, X Windows and Motif Programming, GUI builders (such as UIM/X), UNIX, Windows NT, C and C++, Java, Object Oriented Analysis and Design, Geographical Information Systems, Air Traffic Control Systems, Military Command & Control Systems.

Location: Ottawa, ON, Canada

From: dominique.canevet [mailto: dominique.canevet@thales-is.com]
Sent: Friday, May 11, 2001 5:47 AM
Subject: Questions about Ada language
To: team-ada@acm.org

[...] Thales Information System, Toulon (France). We will have to port an application written in Ada 83 language from SUN OS to Solaris environment. [...]

From: "Beard, Frank" <beardf@spawar.navy.mil>
Date: Fri, 11 May 2001 18:06:28 -0400
Subject: Re: Questions about Ada language
To: team-ada@acm.org

We ported a system of ~180K SLOC of Ada 83 on HP-UX BLS to Ada 95 on Windows NT. We did not have to change any of the pure Ada code. The are very few things that differ from Ada 83 to Ada 95. And the compiler is probably the best way to determine any problems there. The only part we had to worry about was the OS calls. Since, we were using POSIX, we took the POSIX spec and wrote the bodies to call Windows APIs, since no POSIX binding existed at the time. [...]

From: "Martin Dowie" <martin.dowie @nospam.baesystems.com>
Date: Mon, 21 May 2001 17:24:14 +0100
Subject: Re: Computer Weekly survey: Ada does well again
Newsgroups: comp.lang.ada

> Would we be talking about a computer manufacturer like IBM or Compaq? And if so, what would they be doing with Ada?

I met someone [...] from Philips Semiconductor at Ada-Europe'2001, who use it in machine control in chip fabrication plants [...]

From: geoff.webb@xwave.com (Geoff Webb)
Date: 22 May 2001 14:00:25 -0700
Subject: Ada Development Opportunity in Toronto Ontario Canada
Newsgroups: comp.lang.ada

Position title: Real Time Embedded Developer(s). Permanent. [...]

Project Description: xwave is currently engaged in a number of Real Time Development projects spanning the GTA. As a member of the xwave Operational IT team you will have the opportunity to work on mission critical aerospace control system development as well as related technologies.

Responsibilities: You will be involved in multiple phases of software development, including: embedded real time application development, object oriented analysis and design methodology, development of software standards within the software development lifecycle, validation and verification testing, exposure to Project Management using the CMM standard, hardware and software integration and testing.

Qualifications: Mandatory Skills: Ada and Assembler programming, Vxworks or other RTOS, University degree or equivalent background.

Assets (would be nice to have): [...] Experience developing and maintaining large mission critical systems. [...]

From: George Esper <jobs@esper.ca>
Date: Tue, 22 May 2001 19:47:04 -0400
Subject: Ottawa - Ada Developers / Team Leads
Newsgroups: comp.lang.ada

We have immediate openings in Ottawa, Canada for several intermediate and senior Ada developers and team leads. Experience with Rational Apex and air traffic management systems is a definite asset.

Positive publicity for Ada in the UK

From: John Robinson <john @JohnRobinsonAndAssociates.com>
Date: Thu, 29 Mar 2001 12:21:04 +0100
Organization: John Robinson & Associates
Subject: Positive publicity for Ada in the UK
Newsgroups: comp.lang.ada

Just thought I would point out that Computer Weekly in the UK have just published a very favourable article on Ada as a "hot skill":

Computer Weekly, 29th March, pp108

I struggle to remember the mainstream computer press printing anything other than ill informed and inaccurate statements about Ada in recent times. It is a relief to see someone publishing hard facts at last.

From: "Tarjei T. Jensen" <tarjei.jensen@kvaerner.com>
Date: Fri, 30 Mar 2001 13:23:19 +0200
Organization: Kvaerner Group IT

Subject: Re: Positive publicity for Ada in the UK
Newsgroups: comp.lang.ada

If you search http://www.computerweekly.co.uk for ada, you will find it mentioned. And I think it is doing well in that publication. Particularly when they discuss language skills and jobs. Ada seems to be popular in engineering companies.

From: "Martin Dowie" <martin.dowie@nospam.baesystems.com>
Date: Tue, 3 Apr 2001 09:35:05 +0100
Subject: Re: Positive publicity for Ada in the UK
Newsgroups: comp.lang.ada

The article is here

http://www.computerweekly.co.uk/cwarchive/xtra/20010329/cwcontainer.asp?name=S1.HTML&SubSection=2&ct=careers

reassuringly in the 'News/Careers' section :-)

Computer Weekly Survey: Ada Does Well Again

From: Ian Clifton <ian@melvin.dp.ox.ac.uk>
Date: 18 May 2001 17:14:55 +0100
Organization: Oxford University, England
Subject: Computer Weekly survey: Ada does well again
Newsgroups: comp.lang.ada

This has come up before in comp.lang.ada, but it's nice to see that Ada has done well again in the regular Computer Weekly "Skills most in demand" survey, based on UK job advertisements. Ada is up 34% to position 15, the only programming language in the list of top 25 skills to increase in demand. C++, Java, Visual Basic and C still occupy higher positions, but all fell by between 29% and 40%.

Ian Clifton, Dyson Perrins Laboratory, Oxford University, OX1 3QY, UK, ian.clifton@linacre.ox.ac.uk

From: "Tarjei T. Jensen" <tarjei.jensen@kvaerner.com>
Date: Fri, 18 May 2001 23:06:43 +0200
Organization: Kvaerner Group IT
Subject: Re: Computer Weekly survey: Ada does well again
Newsgroups: comp.lang.ada

The article is named "Falling down" and is dated 16-May-2001.

I found it by searching for "skill survey ada" on the advanced search page. The start date was 01 april 2001.

From: "Tarjei T. Jensen" <tarjei.jensen@kvaerner.com>
Date: Sat, 19 May 2001 11:06:39 +0200
Organization: Kvaerner Group IT
Subject: Re: Computer Weekly survey: Ada does well again
Newsgroups: comp.lang.ada

The article can be found at the bottom of http://www.cw360.com/article&rd=&i=&ard=101919

From: "Marin David Condic" <marin.condic@pacemicro.com>
Date: Mon, 21 May 2001 09:54:19 -0400
Subject: Re: Computer Weekly survey: Ada does well again
Newsgroups: comp.lang.ada

The relevant quote is:

> The other top 25 skill to show growth relative to the first quarter of 2000 is Ada, which is enjoying the greatest popularity it has ever known. It featured in 530 ads this time, placing it 15th.

That's an interesting number - I wish we had information about this for the US - not just the UK. I'd suspect that Ada is doing better in the UK than in the US - but that's gut reaction, not data. But of course if Ada usage expands anywhere, that is a good thing because it is likely to spill over into other markets eventually.

And then:

> None of the Ada requirement came from the public sector. Software houses accounted for well over a third of the Ada opportunities, while one in 10 of these jobs was offered by a computer supplier - in this sector league table Ada lies third, behind only C and C++.

Its nice to know Ada is doing so well outside of government circles. Maybe it is an example of "Military Intelligence"? As the military abandons Ada to start trend-following, the private sector starts "discovering" Ada. [...]

> Among other industries, Ada is most in demand in the engineering sector, which again accounts for one in 10 of the jobs on offer. Here also Ada is in third place, behind C and C++ again. Engineering companies are also keen on Pascal and account for fully a half of all demand for this language.

That makes sense. In engineering applications, the emphasis is not so heavily skewed towards time-to-market where the language with the spiffiest development kit & accesories to leverage make the language a winner. In the engineering realm, one is more apt to be concentrating on a central core of code that is complex, has to work properly, and will stick around for a good long time. GUI building, databases, etc. (while they are used and are important) do not become the focal point, so Ada's relative lack of development accessories doesn't hurt it so much. Its benefits get to shine.

I think that suggests where to focus the Ada marketing effort.

Marin David Condic, Senior Software Engineer, Pace Micro Technology Americas, www.pacemicro.com, e-Mail: marin.condic@pacemicro.com, Web: http://www.mcondic.com/

From: Mark Antony <ammonia_hurts@yahoo.co.uk>
Date: Mon, 21 May 2001 21:16:55 +0100
Subject: Re: Computer Weekly survey: Ada does well again
Newsgroups: comp.lang.ada

From an industrial engineer's view, outside of the computer programming industry, I too have noticed the increase in demand for Ada skilled programmers in the UK. Most of it being military applications though, particularly because of the Eurofighter project.

I am seriously considering Ada when looking to work in the programming industry. (I'll probably be too late for the gold rush, but what the heck!)

From: Philip Anderson <phil.anderson@amsjv.com>
Date: Tue, 22 May 2001 09:41:12 +0100
Organization: Alenia Marconi Systems ISD
Subject: Re: Computer Weekly survey: Ada does well again
Newsgroups: comp.lang.ada

> As the military abandons Ada to start trend-following, the private sector starts "discovering" Ada.

No, there are still plenty of Defence projects written in Ada, but the software is written within the private sector not by Government employees.

The trend for years has been for the MoD to do less of its own work and subcontract everything out to Prime Contractors who in turn will subcontract to other companies including Software Houses - now systems tend to be PC-based, companies which used to supply their own computers with a system now just supply software.

Philip Anderson, Alenia Marconi Systems, Cwmbrân, Cymru/Wales

Ada in Context

Facts vs. Hype

From: Jean-Pierre Rosen <rosen.adalog@wanadoo.fr>
Date: Fri, 9 Feb 2001 09:25:18 +0100
Organization: Adalog
Subject: Re: The Good and Bad News about Java
To: team-ada@acm.org

> Engineering executives are supposed to make decisions on facts -- not hype.

I know you wrote "are supposed to", but in practice...

Yeah, that's the problem. If we could achieve that technical decisions be based on technical evaluation of available technologies, Ada would perhaps not win *all the times*, but it would certainly win often enough to make us all rich...

Safety Checks Missing in Hand-written C

From: Tucker Taft <stt@averstar.com>
Date: Thu, 15 Mar 2001 11:11:00 -0500
Organization: AverStar (formerly Intermetrics) Burlington, MA USA
Subject: Re: Increased Interest In Ada?
Newsgroups: comp.lang.ada

[...] Actually, one of the interesting things working with the version of our Ada 95 technology that uses C as an intermediate is how it illustrates exactly what you are giving up by going to C. All of the consistency checks performed by Ada at compile-time, plus the additional checks which are performed at run-time when they can't be proved safe by the compiler, are generally all lost when you write in C "by hand ". By looking at the generated C you immediately see all the run-time checks that remain, and it makes me cringe to think that people writing by hand in C don't have any of those safety checks performed. And that is after our optimizer has already eliminated many of the provably-safe checks.

For what it is worth, the "glued together" compiler is actually quite easy to use, and it "feels" like a regular Ada compiler, except that if you want to check up on what the compiler is doing, you can save and look at the generated C code rather than having to look at the generated machine code. By default, the intermediate C code is deleted, so all that is left is a ".obj"/".o" just like a "regular" compiler.

Tucker Taft, stt@avercom.net, http://www.averstar.com/~stt/, Chief Technology Officer, AverCom Corporation (A Titan Company), Burlington, MA USA (AverCom was formerly the Commercial Division of AverStar)

What You Can't Do With Ada...

From: Ted Dennison <dennison@telepath.com>
Date: Mon, 26 Mar 2001 14:56:18 GMT
Subject: Re: Binary Data File - PLEASE HELP
Newsgroups: comp.lang.ada

[A specific question "How could I do this in Ada?" got the response "Ada simply cannot do that, sorry." -- dc]

BZZZT! Wrong. There are no less than 3 ways to do this (perhaps more). You can: [details irrelevant here -- dc]

About the only thing you can do with C that you can't do with Ada is have a yearly obfuscated code contest with thousands of worthy submissions. :-)

Advantages & Disadvantages of Ada Compared to Other Languages

From: "Mark Lundquist" <mark@rational.com>
Date: Thu, 05 Apr 2001 21:06:39 GMT
Subject: Re: Learning Ada (newbie)
Newsgroups: comp.lang.ada

> What are the advantages & disadvantages of Ada compared to C or C++ and other languages?

OK! I'd like to take a whack at that question...

"Other languages" of course is pretty broad... :-) so most of my comparisons will be with the "C-class" languages, C/C++/Java (and there are others on this group who know Java a lot better than I do, so they can add to what I have to say). Someone like Brian Rogoff :-) can handle comparing Ada with functional languages like OCAML. You probably already understand the issues involved w/ scripting languages vs. compiled languages, so I'll leave that alone.

I'll give a summary first, then go into more detail in a bit.

The bottom line is: "Done sooner, fewer bugs, less pain". And over the lifecycle of a long-lived project, I think that well-written Ada code is more maintainable than well-written code in many other languages (of course it is possible to write crap in any language, and there isn't much point in comparing the crap :-). This benefit accrues primarily to two factors.

The first factor is the expressive power of Ada, which translates directly into better understandability of code written in Ada. I think a programmer who really knows both Ada C++ (for instance) can communicate "intent" better in Ada.

The second factor has to do with early detection of programming errors. Let me try to explain this.. :-) When you make a programming mistake, the result is going to fall into one of four categories:

1) The compiler will reject your code. Or,

2) Your code will compile OK, but when you try to link your program you will get errors (undefined symbol references). Or,

3) Everything compiles and links OK, but when you run your program it blows up with an unhandled exception. Or,

4) No exception is raised, your program just goes beserk. This result can range from subtly or occasionally incorrect behavior or results, to fatal errors (e.g."segmentation fault"), to system hangs, to destructive crashes.

Compile-time errors are generally the easiest thing to figure out. If I make a mistake that results in one of these errors, it could be that I just made a simple mistake that violated one of the language definition rules. In that case, I just figure out what rule I broke and fix the code. Other times, the violation of a language rule points to some underlying logic error or design error. Now I have to step back and do some redesign, but then I'm glad that at least my error was one that could be caught at compile time instead of later, because later means more head-scratching and farting around to figure out what the problem was.

A robust, programmer-friendly language would be consciously designed to "shift" the manifestation of errors along the scale toward the "compile-time" category and away from "unbounded run-time" category. This is just what Ada is designed to do.

The designers of Ada tried to eliminate as much nonsense as possible at compile time. For instance, an Ada "function" is analogous to a non-void-returning function in C (the analog of a void-returning function is an Ada "procedure"). Now, if you write an Ada function with no "return" statement, the compiler will reject it because this is not legal Ada. But in C, it's perfectly legal for a non-void-returning function not to have a return statement. The result of this at run-time is that the caller simply takes as the return value whatever happens to be in the return-value register. Don't miss the fact that this behavior is in fact the *meaning* of that formulation in C. Now how likely is it that the programmer intended this meaning? Fat chance... How likely is it that he/she just forgot to write the return statement, or deleted it inadvertently? Pretty likely. Now consider that the resulting error may not appear until well after the product has been released to the user community.

That's just one example out of many. Another is Ada's "case" statement, compared to the fall-through semantics of C's "switch" statement. And it's well-known that in C, a simple typo of "=" in place of "==" (or vice-versa) can escalate right up to an unbounded run-time error. From the syntax level all the way up, Ada was consistently designed to catch these kinds of errors at compile time, and it does this without imposing burdens on the programmer.

Link-time errors are more of a pain than compile-time errors. The compiler has all kinds of information that the linker can't see, so a compiler is able to give error messages with a lot more specifics about what went wrong. All a linker can say is "I couldn't resolve symbol X", and then it's up to me to figure out what I did wrong. In Ada, linker errors are virtually unknown. The only times you ever get a linker error are due to (a) linking against modules written in other languages; (b) linking against object module archives (which is legitimate, but outside the scope of what is defined by the Ada language, or (c) a bug in the Ada language implementation (compilation system or whatever). You never get a link error when linking an ordinary, self-contained Ada program.

Once you get into run-time errors, it's a whole different ball game. A lot of times, finding the problem means debugging, which is more or less pain depending on the nature of the program and the nature of the error. For a simple, small program, it's not bad. For a large system that's heavily state-dependent and timing-sensitive, debugging can be next to impossible, i.e., doing it is going to require a serious investment in time and creativity.

The "exception" error mode is preferable because (a) it gives you a good hint of where to start when debugging or otherwise investigating the problem, and (b) it represents a boundedness on the error behavior of the program; that is, the error is being "caught" at some point by the program itself rather than going on to wreak more havoc. The program may have no better way to deal with the exception than to terminate, but in that case this is still better than not having raised an exception at all. When an error isn't caught by a run-time check, often the result is a chain-reaction of cascading error effects in the program, and it's not uncommon to begin investigating by debugging a second- or third-order downstream effect of the error (for example, the error causes corrupted data which is later read and causes the observable incorrect behavior).

Obviously, in C all run-time errors fall into the last category (unbounded run-time error), since C doesn't have exceptions. C++ has few run-time checks (bad_cast, bad_typeid) that throw exceptions, so unhandled exceptions usually originate with an explicit "throw" in the program rather than a language-defined check. Java defines a few more run-time checks (such as the array vounds check), but not as many as Ada. Ada defines a large number of run-time checks that raise exceptions, which would otherwise result in unbounded errors. Better yet, Ada's language rules are constructed in such a way that the compilers can often optimize away a suprising number of the language-defined checks. Java doesn't have this ability to the same extent. (It's often asked, "Don't the run-time checks carry a lot of run-time overhead?" The answer is, first of all: "Not as much as you might think", but more importantly, Ada gives you the choice. All the run-time checks can be suppressed, either through pragmas in the source code or compilation options. So you get to decide the cost-benefit tradeoff as you see fit).

Blow-by-blow, here are the technical aspects I see contributing to the factors of "expressive power" and/or "early error detection".

1) I think one of the coolest things about Ada is its 'package' construct. Packages represent the programming concept of a "module" and are absolutely fundamental in Ada. The package construct unifies, very cleanly and elegantly, three important concepts: (1) encapsulation (which is about privacy, i.e. hiding an abstraction's representation from its clients), (2) separation of interface and implementation, and (3) namespace control. Every package has a construct called a "specification" (the interface), and most packages, depending on the contents of the specification, also require a "body" (the implementation). The idea of separation of interface and implementation calls for more than just textual separation, it implies a "contract" specified by the interface which the implementation is obligated to fulfill. So in Ada, if the body is incomplete or incorrect with respect to the spec, you get an easy-to-understand compilation error when you try to compile the body. If you do not provide a body for a package that requires one, then you'll get a prelinker error when you try to link the program (not a linker error complaining that a screenful of symbols is undefined, but a clear error message that you are "Missing body for package Foobar" or whatever).

Compare this with C/C++. The interface is typically given by a ".h" header file containing extern declarations, and the implementation is given by a .[cC] file. But there is no language-defined correctness/completeness relationship between the header file and the implementation file, and the identification of either one with a "module" is entirely notional. If the "implementation" doesn't match, the code is still perfectly legal and will compile just fine. The backstop for catching this is the linker, when it can't resolve all the symbols. Moreover, you can only do this if you are in a position to link a main program, which is an annoyance when developing libraries or developing components of large software projects. And it's quite easy to violate the interface/implementation contract in ways that are not caught by the linker, so and so will cause a run-time error. C/C++ has three separate mechanisms to handle the three aspects of modularity: classes for encapsulation, namespaces for namespace control, and the .h/.c convention to simulate separation of interface and implementation; but the three mechanisms don't fit together snugly. Some other random notes... If you want to inline a member function, it must go in the class declaration, i.e. the header file (thus violating separation). The "namespace" construct in C++ also is inferior to the namespace control provided by Ada packages (don't have time to go into detail on this). In C/C++, namespace control must largely be implemented through ad-hoc policies that must be manually checked and enforced by a human "name czar" (see the book "Large-Scale C++ Software Design", by John S. Lakos -- it covers high-maintennance techniques for working around this and other problems that don't exist -- at least to nowhere the same degree -- in Ada, such as circular compile- and link-time dependencies). Ironically, one gripe against Ada is that it has too many rules. With other languages, instead of language-definition rules that work with you to help you express intent, you have to submit to labor-intensive project policies if you want the project to succeed.

Compare with Java and Eiffel... In both of these languages, a class's interface and implementation are not separated. Java has a mechanism (the "interface" construct) that can be used to simulate this, but that's not really what it's meant for. I think Eiffel also has a construct that can be used to achieve some separation of interface and implementation. But in both cases it would be somewhat onerous to implement a "modularized" design using these features, and the result would be code written in an unnatural style for those languages.

Embracing packages is the "library unit" concept in Ada which allows for true separate compilation of modules while maintaining semantic relationships between them. One result is that the reliance on makefiles for codifying compilation dependencies is rendered obsolete; the compilation system can do all the necessary dependency analysis on the fly.

The bottom line is that since modularity is fundamental to programming, it should be primitive in a programming language.

2) Ada has a powerful type system. Some have called this "strong typing", and strong static type checking is indeed part of it, but not all. It's not just that the type system is "strong", it's that it's also "rich". One aspect of this is the ability to create user-defined types -- not just record types (which are like C/C++ structs and classes), but user-defined numeric types which are distinct from each other and the predefined numeric types, user-defined array types (most array types in Ada are named, while in C-class languages they are all anonymous), real enumeration types that are not aliases for integers and can be used as array index

types (and that don't collapse into ints as soon as you get in with a debugger), and more. Another aspect of "rich" typing is the very cool concept of "type/subtype" (no time to go into this right now, and covered in at least two other recent threads on comp.lang.ada).

The classic example of why you need distinct types even for numbers is something like this:

```
function Area (Radius : Meters)
        return Square_Meters;
```

where the types "Meters" and "Square_Meters" both happen to have the same representation (say, a floating-point number), but are clearly not the same type. If you take something of type Square_Meters and try to pass it as the parameter to Area, you want the language to tell you at compile time.

Packages and the type system head up my short list of technical advantages. A few others:

3) Generic units, which are similar to C++ templates except that they are almost perfectly type-safe and compile-time checkable. They also implement a programming concept called "constrained genericity", which you can read about on the Web or wherever (no time to go into it here). For some, in whose minds Ada went overboard in requiring explicit instantiation of generics, Ada generics do not represent the ideal but are still preferable to C++ templates. When you make a coding mistake when working with Ada generics, you get brief and informative compile-time error, where the comparable mistake in C++ can result in a linker error message that is truly epic in size and whose cryptic syntax renders it virtually unreadable (if you've ever used STL, then you know what I'm talking about! :-)

4) Safe pointers.

[There was no point 5) in this list. -- dc]

6) A crisp model for inheritance and dynamic polymorphism that is *not* based on the idea of a "class". IMHO, class-oriented languages (Simula/Smalltalk/C++/Java etc.) embody an intellectual error in their treatment of encapsulation (privacy and primitive operations, a.k.a. methods), by making the "class", which is really a type definition, also the unit of modularity. The conflation of "module" and "type" in the notion of "class" results in all kinds of distortions: special syntax and sematic complexities for various types of constructors (constructors as a language-level concept do not exist in Ada, since they are unnecessary without classes), the need for a "singleton" idiom, and the need for "friend" classes.

Also, Ada's dispatching model is nice and clean. Inheritance does not imply dispatching, and dispatching is a property of the method invocation, not just the method declaration. And you can dispatch on the return type of a function, not just parameter types.

7) Limited types. In C++ if you want to define an abstraction that retains complete control over its own instances, e.g. does not allow clients to copy instances or test for equality, you have to jump through some hoops -- declaring the abstraction as a class, then declaring private equality/inequality operators private and constructors. In Ada, the idea of a "limited type" is primitive, and you get it by including the single word "limited" in the type declaration.

8) True array types, including constrained and unconstrained array types and multidimensional arrays. The concept of an "array" is another fundamental programming concept, and collapsing to the pointer-based, machine-model level a la C/C++ doesn't do it justice.

9) Support for tasking and task communication/synchronization, at a higher and nicer level of abstraction than the "thread" level. The tasking model allows all kinds of errors to be caught at compile time that are simply impossible when coding to thread-level library routines. The task priority model is unified with interrupt priorities.

10) To make it an even 'Top-Ten List' :-)... The Ada Reference Manual is a masterpiece of definition. That's about all I can say about it!

It could go on, the list of advantages by no means ends there. OK, fine-grained control over the machine-level representation of data structures... package elaboration... lexical scoping... better string handling...

Compared to other mainstream languages, Ada holds a lot of cards in terms of technical advantages. Some technical disadvantages:

1) A lot of people think Ada would be better if it had more support for something like Eiffel-style "Design By Contract" (preconditions, postconditions, invariants). Personally, I'm undecided, but quite intrigued.

2) That's about all I can think of right now. That doesn't mean there aren't a lot of things I'd like to see improved, I just don't think the things on my wish list rise to the level of "disadvantages", especially compared to C/C++/Java.

> I read in Jargon File that "hackers find Ada's exception handling & inter-process communication particularly hilarious." among other stuff. Why is that?

In all honesty, nobody knows. This is just somebody talking out of the wrong orifice :-) The Jargon File seems to be a pretty good source for information about jargon, i.e. slang terms. I doubt if it's much good for anything else. In addition to the "particularly hilarious" nonsense, the JF entry perpetuates the myth that Ada was "designed by committee" (patently un true, as a matter of public record) and refers to Ada's "elephantine bulk", which is hardly fair... Once the C++ standard was published at long last, it was basically just as huge, and even at that, success in C++ still depends on a large knowledge structure that falls outside the language itself -- linker and makefile details, the standard libraries, threads libraries... but mostly a vast body of knowledge of "pitfalls" about which whole books have been written. These pitfalls are all the same kinds of things that were designed away in Ada, whose language definition is roughly the same size.

Mark Lundquist, Rational Software

*From: "Marin David Condic"
<marin.condic@pacemicro.com>
Date: Thu, 5 Apr 2001 18:06:07 -0400
Subject: Re: Learning Ada (newbie)
Newsgroups: comp.lang.ada*

Many Kudos. [...] I like the commentary about catching errors early. You might want to note that catching errors early is not just a matter of coolness or intellectual tidiness. It translates very directly into $$$$$ saved! (I'm currently doing a *lot* of C programming and getting quite urinated-off at the lack of checking that makes me have to get code loaded into the box, executed, exploded and debugged just to discover some kind of parameter mismatch happened that I could have fixed at compile time & saved myself the whole development iteration!!!) Money is a pretty strong incentive for the commercial developer - it just isn't well known that Ada will, in fact, save you quite a bit of it.

Marin David Condic, Senior Software Engineer, Pace Micro Technology Americas, www.pacemicro.com, e-Mail: marin.condic@pacemicro.com, Web: http://www.mcondic.com/

References to Help Make a Case for Ada

*From: Siddhartha Ray
<SiddharthaR@TucsonEmbedded.com>
Date: Tue, 10 Apr 2001 17:06:46 -0700
Subject: Ada recommendation
To: "GNAT Discussion List"
<gnatlist@lyris.seas.gwu.edu>*

I have been given charge of coming up with a proposal to use Ada within our company.

I have to provide my company documentation and references plus make my own proposal on why we should select Ada 95 over C/C++. If this proposal is successful then we will go ahead and start Ada 95 training here and start moving away from C++ and towards

Ada 95. Lot of people here are getting absolutely tired and fed up of C++ and we do not want to risk using C++ in our upcoming project contrary to what a lot of C++ supporters want us to do. But we just had a long bout with C++ usage in the past year and a half and it has become a nightmare. However, the people who use C++ are in complete denial of the problems.

However, this proposal has to also document why we should NOT use plain C.

Since people on this list are strong believers in Ada and its advantages, could you direct me to all the information that would help me. I would also like some references to organizations that I can contact and get actual documentation and guidance. Organizations like maybe AdaIC or some such other organization that keep a track of Ada projects/successes/failures etc.

From: "Holmes F. Boroughf" <Boroughf.HF@acm.org>
Date: Tue, 10 Apr 2001 23:18:23 -0500
Subject: RE: Ada recommendation
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

This was the subject of a lot of debate in the US Dept. of Defense. The Air Force Software Technology Support Center has a publication "Crosstalk" which devoted quite a bit of space to this debate. Many of the discussions made a good case for Ada 95. The URL of their site Is http://www.stsc.hill.af.mil/ => Crosstalk => Search back issues.

Below is the head line from a search there. "Documents 1 to 10 of 33 matching the query " ("Ada95" or "Ada 95") AND "C++" ". "

This is one of many places that carried on this debate.

I wish you the best of fortune. Let the facts win over emotion and Ada 95 will be your language.

From: "Jean-Pierre Rosen" <rosen@adalog.fr>
Date: Wed, 11 Apr 2001 10:18:10 +0200
Organization: Adalog
Subject: Re: Ada recommendation
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

However, this proposal has to also document why we should NOT use plain C.

My 0.02 Euros: It is not an issue of programming language, but an issue about the level of abstraction where you want to address your problems.

Simply stated: C is the best language for programming a computer; Ada is the best language for developping software applications.

(Please don't quote me on the first half without the second one ;-)

If you think that a program is nothing but a sequence of instructions executed by a computer, then by all means go for C. But if you start thinking about structures, modules, organization, maintenance, etc. Ada is the way to go. So I think that the way to address your question is two-level: 1) Decide at what level you want to do your programming tasks 2) Choose a language that matches your answer to 1)

J-P. Rosen, rosen@adalog.fr, http://www.adalog.fr

From: tarjei <tarjei@online.no>
Date: Wed, 11 Apr 2001 14:36:03 +0200
Subject: RE: Ada recommendation
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

> However, this proposal has to also document why we should NOT use plain C.

You should search comp.lang.ada for arguments for this discussion. Some claim that plain C works better than C++. That means that there is less effort to solution (ETS) with C than C++.

When people estimate how much time they have spent creating a piece of software one tend to forget to include the time it took to get it to work. That is the time it took from the software compiled the first time until it did what it should do. That is usually why people think that the ETS of C is less than that of Ada. They get a rude surprise when they actually measure it.

C and C++ appeals a lot to our tinkering instinct, but as professionals we need to measure.

Remember that a lot of people are concerned with C++ because they think that it will look good on their resume. Don't forget that the number of Ada jobs is steadily rising.

The important bit about Ada, Eiffel and Delphi is that they offer a culture. If you embrace that culture it will pay you back even if you do things in C and C++.

My philosophy about Ada.

Everybody who work in large and not so large organizations should know that in order to be efficient you have to have a buraucracy. Whether one likes it or not, the process of software development is much like running those organisations. Let me assure you that if there was no buraucracy, we would all be very cold and very hungry. A lot of people fails to appreciate the enormous effort that is "behind the scene". [...]

Ada is an excellent tool which support the ideas, methods and state of mind which are neccessary in order to create resilient and reliable software.

From: Laurent Guerby <guerby@acm.org>
Date: Wed, 11 Apr 2001 20:42:06 +0200
Subject: Re: Ada recommendation
To: "GNAT Discussion List" gnatlist@lyris.seas.gwu.edu>

You should have a look at <http://www.adaic.org/> which has a lot of advocacy material. I think also contacting Ada vendors should be a good idea, they might be interested in showing you the product and services they offer around Ada and maybe some of what their Ada clients are doing (this is useful to counter the "lack of support" and "lack of users" arguments). No doubt the Ada only vendors have lots of Ada advocacy in their bags ;-). Most Ada vendors are part of the Ada Resource Association <http://www.adaic.org/ara/>.

From: "Hans-Olof Danielsson" <Hans-Olof.Danielsson@swipnet.se>
Date: Wed, 11 Apr 2001 22:49:22 +0200
Subject: Re: Ada recommendation
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

> You should have a look at <http://www.adaic.org/> which has a lot of advocacy material.

Especially look at http://www.adaic.org/docs/flyers/mrp.html

[See also "Malaysia - Resource Planning System" in AUJ 21.1 (April 2000) for more on this Integrated Manufacturing Resource Planning (MRP) system for furniture manufacturers, first developed in many programming languages, including C/C++, but now in Ada95. --dc]

From: Michael Feldman <mfeldman@seas.gwu.edu>
Date: Wed, 11 Apr 2001 23:07:44 -0400 (EDT)
Subject: Re: Ada recommendation
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

> Can somebody point me to maybe a webpage that shows an increase in the use of Ada 95. Even a small increase in usage would suffice.

I'm not sure if this will help, but see http://www.seas.gwu.edu/~mfeldman/ada-project-summary.html

It's not a trend report, but a complete listing of everything I'm aware of. Needs a bit of updating, which I'll do after my semester ends here in mid-May.

From: "steved94" <steved94@home.com>
Date: Wed, 11 Apr 2001 20:31:14 -0700
Subject: Re: Ada recommendation
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

I wish you the best of luck in this endeavour. I went through the same process in my organization a few years ago, and now we use Ada 95. There have been no regrets on that decision.

An interesting thing I noticed in my research is that you will not find any documentation on projects where a shift has been made from Ada to C/C++ and increased reliablilty or efficiency (at least I didn't).

From: Siddhartha Ray <SiddharthaR@TucsonEmbedded.com>
Date: Thu, 12 Apr 2001 08:03:47 -0700
Subject: Re: Ada recommendation
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

[A pointer to Vision Systems. -- dc]. They use Ada95. Go to www.ghs.com and search for Vision Systems. They aparently changed from C++ to Ada95. This is a brand new project [...]

http://www.ghs.com/news/archive/990719vs2.html [...]is the direct link to the Vision Systems web page.

[See also "Ada 95 Accelerates Time to Market" in AUJ 21.3 (October 2000) for more on this video system product that also switched from C to Ada 95. -- dc]

From: "Hans-Olof Danielsson" <Hans-Olof.Danielsson@swipnet.se>
Date: Sun, 15 Apr 2001 18:54:07 +0200
Subject: Re: Ada recommendation
To: "GNAT Discussion List" <gnatlist@lyris.seas.gwu.edu>

Here is more interesting stuff [...] applicable to the above posting.

[This was a repost of a reply in comp.lang.ada to the question "What is the objection against *procedures* returning a value?" and to the subsequent remark "It would be nice to be able to ignore a function's return value.": -- dc]

A procedure may modify parameters, but may not return a value.

The practice in C, C++, and Java of ignoring return values from functions or methods may seem very convenient. In fact it is a fundamental violation of the contract defined for a function. This is the single greatest reason exceptions were added to C++, and included from the start in Java. A common C programming paradigm is to have a function return an indication of success or failure. That same function may also modify some or all parameter values. It is also very common in C to ignore the return value of a function. What is the obvious problem? If the return value indicates success or failure, and that value is ignored, then the programmer is ensuring erroneous execution. Whenever the function fails, the programmer using that function continues to use the modified parameter values as though the function succeeded. This usually leads to very difficult debugging sessions because the error may not become obvious in the program until some later code block. It also relies upon the sometimes lucky choice of test cases to generate the error condition. If your test cases do not generate the error, then you will release erroneous code to your customer, which they will find for you, usually at significant expense for your company.

In Ada you have a very strong contract concerning procedures and functions. Functions always return a value. That value has significant meaning. It cannot be ignored. Procedures are the form of subprogram to use when no significant value is to be returned.

Jim Rogers, Colorado Springs, Colorado USA

From: "Marin David Condic" <marin.condic@pacemicro.com>
Date: Tue, 15 May 2001 14:33:47 -0400
Subject: Re: Universities in the US
Newsgroups: comp.lang.ada,comp.lang.lisp

[...] Now that I am being forced (once again!) to program in C, I really realize how much I miss Ada. As much complaining as I've done about Ada's representation clauses - I now want to take it all back! I've got very nearly nothing in C to help me with data representation. The level of pain is so high in comparison to even weak support for representation in Ada, that I absolutely cannot fathom why embedded programmers love C so much and ignore Ada?

Perceptually speaking, I think Ada is experiencing some resurgence of interest in the programming community. Especially with student programmers. Perhaps there is becoming a realization that C/C++/Java/etc. just don't offer as many nice features as Ada does for serious, industrial-strength, larger-scale development. Hopefully, the trend will continue.

[Lieven Marchand <mal@wyrd.be> replied: -- dc]

It might. When I first encountered Ada in college, I hated it, partly because of a very bad compiler (a slow horror that tried to correct errors by inserting random tokens under VM/CMS). Now that I've had some experience with maintaining large programs hacked on by many people over the years, I start to appreciate a lot of the Ada philosophy.

[And Marin David Condic responded: -- dc]

I'd have to agree that in the early days of Ada, she basically shot herself in the foot by the bad quality (and huge expense!) of the compilers that first appeared. People became convinced that Ada would never work because it was "too complex" for anyone to build a descent compiler for the language.

Nowadays, the world is considerably different. Computers got more powerful and Ada compilers have become significantly better. Compiler technology now is such that Ada can produce code as good, or better, than just about any other popular language. Also, you have GNAT freely available and a number of other compilers (such as RR and Aonix) that are available at affordable prices. Maybe back in 1983, Ada was "ahead of its time" but today all the right things are in place to enable one to seriously consider Ada for any development effort.

Language Standards and Complexity of Compiler Writing

From: Ted Dennison <dennison@telepath.com>
Date: Fri, 04 May 2001 16:22:40 GMT
Subject: Lack of portability/standardization in C++
Newsgroups: comp.lang.ada

This issue [the lack of portability and standardization in C++ in practical life -- dc] has come up here a fair bit recently, mostly having to do with the validity of the claims on Mozilla C++ Portability Guide page (http://www.mozilla.org/hacking/portable-cpp.html).

There's a really good article up on kuro5hin today about this exact issue. The direct link is http://www.kuro5hin.org/?op=displaystory;sid=2001/4/25/22959/3540

[The article is entitled "The Balkanization Of A Popular Programming Language (Technology)" and dated Wed Apr 25, 2001. -- dc]

Since its dated last week, I doubt anyone is going to seriously try to claim that *this* is out of date. :-)

My favorite part is:

> One begins to wonder if it's been almost four years since the draft discussions and yet no single compiler implements the entire standard, then when will the additions currently being proposed finally make it into the standard and then into the real world in a way that is compatible across platforms and compilers? Two years? Four? Six maybe?

T.E.D. homepage http://www.telepath.com/dennison/Ted/TED.html

From: Ted Dennison <dennison@telepath.com>
Date: Fri, 04 May 2001 16:47:52 GMT
Subject: Re: Lack of portability/standardization in C++
Newsgroups: comp.lang.ada

There's also a thread in the responses that compares the situation to Ada's. The most interesting response there was this one: http://www.kuro5hin.org/?op=comments&sid=2001/4/25/22959/3540&cid=17#17, which was posted by someone involved in IBM's C++ compiler effort.

[Start of quote. It was in response to: -- dc]

> I don't think it is technically difficult to implement a standards compliant C++ compiler. It takes time, money and some talented programmers. It also requires a company or group that is

willing to make standards compliance a primary goal.

Let me tell you, you couldn't possibly be more wrong. I worked with the IBM group that developed VisualAge C++ v4. I sat in the interminable meetings where we tried to figure out just what the heck the template section of the specification meant. We were absolutely determined to produce the first fully standards compliant C++ compiler. And the group of programmers that worked on the project is the brightest collection of people I've ever seen in one place, working under one of the finest managers I've ever seen (and he was also a great programmer; the bulk of the template system was his).

In the end, we shipped a system that we thought met the standard. It turns out that we didn't even get templates right.

C++ is a horrible language. It's insanely complex, and the specification is among the most poorly specified, overcomplex, incomprehensible ream of rubbish that I've ever seen. I don't care how much time and effort you put into it: I do not believe that there will ever be a fully compliant C++ compiler. I'm not even sure that "fully compliant" means anything when applied to C++.

You mention Ada as a counterpoint, because people thought it was unimplementable. There's a big difference with Ada, and it isn't the political and resource issues that you claim. The Ada spec, while wordy, is a complete, thorough document. It can be difficult to read, because precise specifications are difficult to read, but it very clearly specified everything. The Ada specification is, in my opinion, just about the best language spec that could possibly be developed by a committee. And that is what makes Ada implementable.

The complexity of C++ absolutely dwarfs Ada, and the specification is disorganized, jumbled incomplete, and confusing. Writing a fully compliant compiler for C++ would be at least an order of magnitude more complicated than the original Ada, and the specification is far worse.

From: "James Hague"
<jamesh_fake@volition-inc.com>
Date: Tue, 15 May 2001 13:52:33 -0500
Subject: Re: Universities in the US
Newsgroups: comp.lang.ada,comp.lang.lisp

[In a thread about renewed interest in the programming community for Ada: -- dc]

Way back when, Ada had a bad reputation for being a bloated language and requiring bloated and unreliable compilers. What's amusing now is that Ada is a simpler language than what has become the norm (i.e. C++)! And writing a C++ compiler is more daunting than writing an Ada compiler.

From: "Marin David Condic"
<marin.condic@pacemicro.com>
Date: Tue, 15 May 2001 15:51:35 -0400
Subject: Re: Universities in the US
Newsgroups: comp.lang.ada,comp.lang.lisp

That should be pretty easy to understand. The syntax & semantics of Ada83 were not as ill-conceived with respect to clarity (well defined behavior) and expandability, so Ada95 didn't really require major gyrations to produce. Everything was more a matter of a variation on a theme rather than a bag hung on the side into which you throw new parts.

The syntax & semantics of C were maybe O.K. for the limited use originally expected, but in terms of clearly defined semantics and readily expanded semantics, C just didn't cut it. So when you try to expand it to C++, you've got incredibly complex rules of behavior with numerous special cases, exceptions and "implementation defined's". Just because of the screwy semantics of function operators and everything in C wanting to be a function, you get some of the most convoluted and strange rules of behavior for passing parameters within C++.

When thinking about C++, I am reminded of the Harley Davidson Design Principle: "If it breaks, make it bigger. If it sticks out, chrome it." :-) [...]

Ada was designed to be large (Full Figured?) C++ organically grew to be large. Hence its easy to see why C++ implementations are far harder to do.

Ada and Java - Impact on Programming Culture

From: Tony Lowe <tlowe@issintl.com>
Date: Mon, 26 Feb 2001 10:40:32 -0600
Subject: Re: sucky software
To: team-ada@acm.org

When I applied for my new job, and came in for an interview, my new boss said "I've never worked with Ada, but I hear that anyone who does is usually an excellent software engineer". Thus in one statement making my assertion that this is not a language issue, but a cultural one. Those who choose to program in Ada usually preferred a moderated approach to software development, preventing errors from happening. I would bet that the average person on this list actually has oil changes when they are suggested.

Saying Ada compilation slows you down is actual kind of silly, considering Java requires you add certain exception handlers simply to compile, and then the basic javac app does not bother to see if a dependent has been recompiled before running the application, calling the class file and finding out the method no longer exists...

The key may not be in how the final product is programmed, but in how the process is completed. I have found that if you produce a GUI executable that 'dances' early, the scrutiny about your ability to finish quickly is removed.

From: "W. Wesley Groleau x4923"
<wwgrol@ftw.rsc.raytheon.com>
Date: Mon, 26 Feb 2001 12:29:08 -0500
Subject: Re: sucky software
To: team-ada@acm.org

> Java requires you add certain exception handlers simply to compile,

It may be even worse than that. Say you discover that a certain utility method passes out invalid data if the file containing the valid data doesn't exist. This data causes an index out of range far downstream from the actual error.

The sensible solution, of course, is to check for the file's existence. But if missing, how do you recover? Only the unit that made the call has sufficient information to handle the fact that the file it asked for is missing. So we should throw fileNotFound, right?

Only that means you have to track down all the clients of this method and add the new 'throws' clause. Or write a less than optimum catcher in the utility method.

Which fix do you suppose they'll use, the fast one or the right one?

Java's Origin

From: "W. Wesley Groleau x4923"
<wwgrol@ftw.rsc.raytheon.com>
Date: Tue, 6 Mar 2001 11:22:22 -0500
Subject: Re: Problems with C-class languages
To: team-ada@acm.org

> Today the same lecturer again claimed that Java is the result of cherrypicking from Smalltalk, C++, and Ada. [...]

Either deliberate dishonesty or complete ignorance. [...]

The Java language is the result of a team at Sun having trouble with C and inventing a language that eliminates the items (and only those items) that gave them the most trouble.

Now I can't prove that, but that is what Sun's own comments imply. I can prove, if I care to make the effort, that Sun produced a 30-minute infomercial in which one of their people opened a C or C++ book in front of the camera, with lots of stuff lined out, and said something to the effect of, "We just crossed out everything unsafe, and that's how Java was designed."

I can also prove that at least one Java book from Sun and at least one not from Sun say that Java does not do X because X is not safe, where X is any of several things that Ada 95 does "safely ". (And some of them were in Ada 83.)

I can also (if the page still exists) find a Web page showing how great Java is by comparing features to other languages -- but avoiding even a mention of Ada or Eiffel.

I can, but will I? I don't feel much like fighting any more. I think I'll change careers--to Amish-style farming.

Students prefer Ada over Java

From: "Crispen, Bob" <Robert.Crispen@hsv.boeing.com>
Date: Mon, 5 Mar 2001 13:57:49 -0600
Subject: Re: Students prefer Ada over Java
To: team-ada@acm.org

John McCormick <mccormic @cs.uni.edu> wrote:

> This semester I learned through a student taking the Algorithms course that no student is using Java. A few are using C or C++ , one is using PERL , and the rest are using Ada. [...] I find fact that none are using Java particularly enlightening as Java is the last language with which they had significant programming work.

I wanted to like Java. God knows I did. And for a while I held onto the hope that the Java runtime would improve or the state of computers would improve to the point where it wasn't buggy, crash-prone, incapable of doing what you need, and slooooooooooooooow.

Java borrowed a number of very good things from Ada: interfaces and array bounds checking are notable standouts. Java avoided some of the worst pitfalls of that patchwork of a language C++. They didn't borrow enough, and perhaps they didn't avoid enough, but that's neither here nor there. When a more mainstream, more heavily hyped language than Ada ever was uses language features based on sound software engineering principles, everybody benefits.

That Java (and PHP and perl and ECMA-262 and...) sorta look like C instead of sorta looking like Pascal is a *huge* don't-care for me. All the really interesting stuff is in the library calls anyway. That it uses C++'s soggy inheritance instead of (what seems to me) Ada's crisper model is a shame, but not something you can't work around.

That Microsoft muddied the waters with Visual J++ is a tragedy on a micro scale (I could almost hear the sobs on the other end of the email as I told some poor kid that he'd have to port his VJ++ code to Java to get it to work where he needed it to), but on a macro scale, it was a nop. Not even noise.

And Java zealots make the most rabid of Ada zealots look like rank amateurs. Perhaps they're so fanatical *because* the future hope of Java is so far from the reality, while Ada fans generally got some immediate gratification -- Ada compilers have always been pretty good -- to cool their fires. Java fans have been wandering in the desert for years and years without a drop of water in sight.

But at the end of the day, you just gotta face it: Java as it exists here and now doesn't just suck, it blows chunks. It crashes your web browser -- I run 3 web browsers, and java that two of them like will crash the third; it fails for mysterious, inexplicable reasons (one of my favorites is the number of people who've had to load the JDK *and* the JRE); and did I mention that it was mind-numbingly, frustratingly, maddeningly slooooooooow?

So nobody's using Java is far from a surprise to me.

That they're using Ada is a tribute to you, and you should be proud of your work in turning out people who'll be able to write software that doesn't suck.

Bob Crispen, bob.crispen@boeing.com

From: Geoff Bull <gbull@acenet.com.au>
Date: Thu, 8 Mar 2001 11:32:33 +1100
Subject: Re: Students prefer Ada over Java
To: team-ada@acm.org

[...] The reason that students prefer Ada over Java for an algorithms class is almost certainly Ada's much superior abstraction capabilities. And not being forced to used objects even when you don't want them.

Run-time Performance

From: john.mccabe@baesystems.com
Date: Thu, 8 Mar 2001 10:42:05 +0000
Subject: Re: Students prefer Ada over Java
To: team-ada@acm.org

> Yeah, [Java]'s not the faster language out there. But let's step back a few years and remember that Ada once had speed problems. It's not until the compilers improved that the reputation started to fade away.

Yes, but let's look at it in context. Ada (83) was pretty much by far the most complex high level language devised at the time. It also required the compiler vendors to develop a multi-tasking runtime system, and to provide masses of checks for array ranges and so on.

They also had to contend with microprocessors with no, or very little, built in support for multitasking, particularly as far as task-switching is concerned. There has been a lot of work done in this area since 1983!

On the other hand, pretty much everything that exists in Java already exists in other languages. Even the JVM is, as has been pointed out recently, not exactly an original concept!

Basically what I am saying is that not only was there a vast learning curve involved in creating Ada compilers, but the hardware support for many of the features of the language just didn't exist. This is absolutely not the case with Java. [...]

From: "Crispen, Bob" <Robert.Crispen@hsv.boeing.com>
Date: Thu, 8 Mar 2001 07:57:00 -0600
Subject: Re: Students prefer Ada over Java
To: team-ada@acm.org

[In reply to references to "the JVM": --dc]

"The" JVM? I have at least four I know about on my computer: the one built into Netscape, Microsoft Virtual Machine, Sun's JRE 1.3 (Opera uses this), and the one in the JDK.

With the exception of the two from Sun (I *think* with that exception!) each of them has different versions, capabilities, and bugs.

Since some folks have been a little skeptical that I could find really sucky Java applications -- applications that are otherwise very good, but that Java made sucky -- allow me to give a few examples.

I have two Java applications from Trapezium: Vorlon, a free VRML syntax checker, and Chisel, a reasonably inexpensive syntax checker and cleaner/decimator/etc. Chisel runs on my machine. Vorlon doesn't any more. Nobody knows why.

When you ask Chisel to open a file, sometimes it brings up the file open dialog box, sometimes it doesn't. Sometimes the file text display updates, sometimes it doesn't. This is on a machine a month old, with a brand-new install of WinMe, and a *lot* of CPU power, memory, and disk. I believe this is due to my getting ahead of the program and clicking on a control before the program is ready for me to do that. I'm seldom able to do 9 or 10 file operations in a row, regardless of what they are.

Perhaps you're familiar with Xeena, an XML development tool by IBM written in Java that I've occasionally used to follow the development of X3D (sort of VRML in XML). One of my colleagues reported that on a 500 MHz Pentium III, Xeena takes 30 minutes to bring up a file ready to work on.

Or, rather, it does on those occasions it doesn't simply hang or crash sometime before that 30 minutes has expired.

Flamingo Optimizer (alas, no longer with us) was a multi-algorithm mesh decimator that kept texture coordinates straight -- which I think is magic. Written in Java.

Trouble is, a multi-algorithm mesh decimator is best used when you tweak a parameter, see what the result is, tweak again, and so on. Cycle times were simply too long. And it would crash after five or six iterations.

I can give examples all day of software in Java that not only sucks but sucks horribly -- software written by people who are plenty smart who've done the best they can with a JVM (or set of JVMs) that might as well be wearing a label "Trabant inside". [...]

From: Geoff Bull <gbull@acenet.com.au>
Date: Fri, 9 Mar 2001 12:07:17 +1100
Subject: Re: Students prefer Ada over Java
To: team-ada@acm.org

[...] There are a number of obstacles to good Java performance. The first is that the language design only allows classes, the only form of abstraction provided, to be allocated on the heap. Sun's JVM has a very fast allocator. I have found Java code will give "native" code a run for its money if both are doing similar amounts of heap allocation.

Another bottleneck I have uncovered with Java is the inability to turn off run time checks. This is noticeable with "generic" collection classes - when casting an Object back to its true type a runtime check is performed. In some cases I have gained massive performance improvements by replacing a class in java.util with my own class specific version.

One reason for using Java is that one can often get a program going in less time than with C/C++. This is also true of Ada, of course, but most people don't want to know it.

Numerical Computation and Ada 95

From: "N&J" <nikogian@hotmail.com>
Date: Wed, 9 May 2001 22:13:31 +0300
Organization: National Technical University of Athens, Greece
Subject: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

I have posed the question whether Ada95 is suitable for numerical programming at the newsgroup of numerical analysis.

I asked if Ada95 is better that Fortran90/95 and C++ for writing programs for numerical computation and the reply was that the Ada language specification requires extensive run-time checks which slows down the programs. I thought it would be better to ask your opinion too. I have noticed that there are too few numerical programs in Ada95 for numerical computation is the above fact the real cause for this? Finally would you suggest someone to use Ada95 for numerical code?

From: Ted Dennison <dennison@telepath.com>
Date: Wed, 09 May 2001 19:43:05 GMT
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

This has been asked many times before. You might have been better served by doing a groups.google.com search on it.

The basic conclusions were generally:
- Compilers vary so much in the quality of their optimization (even within the same language) that it really doesn't make sense to talk about relative speeds of different compiled languages.
- There is nothing about Ada that makes it inherently slower than any other language.
- Theoreticly, one ought to be able to get *better* optimization out of Ada than Fortran or C for the same effort, as all the rules and limitations of Ada language give the compiler a lot more information to work with.
- Fortran users tend to care a lot more about number-crunching speed than other language users, so their compilers generally optimize that stuff better.
- In real-world terms none of this hypothetical stuff matters. What you need to do is compare the speed of the code generated by the compilers you are considering, for the kinds of operations you are going to be performing. Even here you have to be careful, as its easy for a neophyte in any langugage to stumble over a construction that is needlessly difficult for their compiler to optimize.

Just to give an example, the Aonix Ada compiler for NT doesn't provide *any* optimization options whatsover (at least it didn't last I used it). The Gnat Ada compiler provides the typical "-On" where n is in 0..3, along with some others. The x86 vxWorks GreenHills compiler we use here for real-time work provides flags for all sorts of arcane optimization options. Now it would be quite easy for someone to compare what the Aonix compiler does with an algorithm to what a serious numerical Fortran compiler does, but it wouldn't exactly be a fair comparison.

From: Dr Adrian Wrigley <amtw@linuxchip.demon.co.uk>
Date: Thu, 10 May 2001 01:41:35 +0100
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

I believe that Ada95 is well suited to numerical programs. Although I have only used the "77" flavor of Fortran, and prefer C to C++, the usual advantages of Ada shine:

Legible programs, more thorough correctness checks, etc.

I would also add that Ada generics make it much easier to express the algorithms, without being burdened by instance specific details. If the code is algorithmicly rich, you'll get efficient, correct code in less time than the other languages mentioned, once you know the languages equally well. Interfacing to other language libraries is usually straightforward, but still more work than keeping it all in one language. If you're just bolting together other people's code, the pressure is always to use the same language as they do.

Another thing I have found is that writing correct multi-threaded code is much easier with Ada95 than anything else I have used, making it simple to take advantage of multiprocessor machines, and concurrent computation and I/O. This could be of relevance with large memory problems going into swap a lot.

The issue of performance is one of my biggest worries when choosing Ada95 over other languages, when speed is critical. Comments elsewhere in this thread suggest that this is not normally a problem, but I have found that great care is needed to avoid inocuous constructs which result in slow code. Sometimes this prevents writing the code in the most "natural" way. Things may be better now with the latest GNAT (I developed most of my performance-critical code under GNAT 3.11, 3.12). Of course, you need to take great care in C or C++ too. Fortran(77) seemed to be a lot more robust in performance terms, perhaps because it was quite limited in what you could do. Try some simple example comparisons in your proposed environments. They may not be representative, but at least you'll have some idea what to expect with "real" code.

A lot will depend on whether you can "go it alone", or whether you need support from your colleagues, boss, customers etc., as well as the size of project and penalty for failure. If you are in a big department hooked on F90, or working on a major new project for your employer, the biggest risks may be political. You would need a lot of confidence, with sound justifications to introduce Ada. The skeptics will be convinced that all the problems you encounter are due to the use of Ada. Usually they'll be wrong. If you don't have the confidence to use Ada for a project, don't try. Go home and familiarize yourself with the technical issues surrounding the language on your own (Linux) machine.

From: tmoran@acm.org
Date: Thu, 10 May 2001 06:48:28 GMT
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

[...] A simple minded transliteration from one language to another can have pitfalls, accessing arrays in row-major vs column-major order being an example. When I went to translate an algorithm from "Numerical Recipes in Fortran" it became obvious during the translation to Ada that the Fortran version was doing some extra, wasted, iterations. Do you want to compare the Fortran version against an exact translation that includes those wasted cycles, or against an Ada version that preserves the algorithm, but not the poor coding? It's not always obvious what's the appropriate comparison.

From: James Rogers <jimmaureenrogers@worldnet.att.net>
Date: Wed, 09 May 2001 21:57:40 GMT
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

You might look at the work done by Martin Stift at http://fedelma.astro.univie.ac.at/web/home.html

Martin has very successfully used Ada in the field of Astrophysical calculations. This is a field where Fortran is the traditional language of choice. Martin explains why he believes Ada is a much better choice.

Jim Rogers, Colorado Springs, Colorado USA

From: stift@fedelma.astro.univie.ac.at (Martin Stift)
Date: Thu, 10 May 2001 13:54:19 +0000 (UTC)
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

Well, I have had the chance to compare the numerical performance of Ada83 and of Ada95 with Fortran77 with quite satisfactory results.

Back in 1994 I had a program written in Fortran77 which carried out spectrum synthesis in magnetic stars. This program had been translated into Ada83 with considerable restructuring of the code in order to take advantage of the superior expressive power of Ada83. The code did mostly number-crunching.

Running the respective Fortran and Ada83 codes on a DEC Alpha under OpenVMS resulted in the Ada code being about 5% slower than the Fortran code. At that time I had 20 years experience in Fortran and 1 year experience in Ada83.

In 1997 I definitely switched to Ada95, working on a Silicon Graphics Origin200 4-processor server. In the beginning the GNAT compiler running under IRIX had to use the old and slow o32 ABIs whereas with Fortran one could use the much faster new n32 ABIs. In some extreme cases, execution times could differ by as much as a factor of 3. However, when using the o32 ABIs for both Fortran and Ada95, performance was about the same to within 10-20%.

The situation has greatly improved since. Although SGI is known for its advanced Fortran compiler technology, the GNAT compiler by now performs impressively. The famous "abstraction penalty" can be encountered both in Fortran and in Ada95. The calculation of the Voigt-Faraday function, which is the most time-consuming part of my code, in Ada95 takes about the same time (within 5-10%) as with Fortran.

I made another test with a celestial mechanics program which I translated directly into Ada without any substantial modifications to accomodate Ada features. Again, the program just does number-crunching, extensively using more-dimensional arrays which should be treated very effectively in Fortran. Still, with GNAT 3.13, the Ada95 version was *faster* than the Fortran77 version.

One of the major advantages of supercomputing with Ada95 is the fact that it provides a thread-parallel model. You don't have to resort the MPI but can parallelise your code with kernel language constructs. Protected objects have very little overhead and are perfectly suitable for synchronisation. Tasks also are easy to handle. At present, I am running my codes on an Origin2000, using up to 64 processors (Amdahl's law makes it impracticable to use more processors for the problems at hand).

Summarising I can say that Ada95 is a viable alternative to Fortran. It is fast and safe, the parallel constructs are easy to use. More information on supercomputing with Ada95 can be found in

Stift, M.J., 1998. Astrophysical Supercomputing: Ada95 as a safe, object oriented alternative. In Reliable Software Technologies - Ada Europe'98, ed. L. Asplund, Lecture Notes in Computer Science, Vol. 1411, p.128-139.

Stift, M.J., 1998. Scientific programming with Ada95: object oriented, parallel and safe. Computers in Physics, Vol. 12, No. 2, p. 150-158.

These publications give some ideas as to how the Ada95 features can be applied to (astro)physical problems. Synchronisation as presented in 1998 is somewhat outdated now, the solutions presented in the above-mentioned papers are only useful on small machines like my Origin200. They have in the meantime been replaced by much more efficient algorithms that work on large supercomputers.

From: "Martin Dowie" <martin.dowie@nospam.baesystems.com>
Date: Thu, 10 May 2001 09:00:42 +0100
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

Taken from: http://www.adaic.org/docs/present/whyada/tsld013.htm

"For duplicate functions written in C and Ada for a C30 DSP using the Tartan compiler, comparisions indicated that the Ada code was faster than the C code."

[and from another message: -- dc]

A fuller account can be found at http://www.adahome.com/Ammo/Stories/Tartan-Realtime.html by John Stare of Tartan Inc.

From: Jeffrey Carter <jeffrey.carter@boeing.com>
Date: Thu, 10 May 2001 18:21:58 GMT
Organization: The Boeing Company
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

[...] Back in the dark days of Ada 83, the highly optimized Tartan Ada compiler produced faster code than the C benchmarks Tartan used to convince people to buy their *C* compilers instead of using the free one that TI shipped with their DSPs. An article in _Ada Letters_ described the features of Ada and C that allowed them to do better optimization for Ada than for C.

There was also a case of a Tartan Ada compiler producing faster code than hand-optimized assembler. This was described in a TRI-Ada presentation. The people developing the Ada made no special effort to make the code fast, because the purpose of the project was to prove that Ada was NOT fast enough. The project failed, [...]

In general, using modern compilers and modern processors, equivalent programs will produce equivalent execution times.

From: Ted Dennison <dennison@telepath.com>
Date: Thu, 10 May 2001 14:11:34 GMT
Organization: http://www.newsranger.com
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

> For duplicate functions [...] the Ada code was faster than the C code.

As I remember, Bevin Brett reported similar results with the DEC Ada compiler. The problem with taking this at face value is that there's no way to tell if this might be due to the effects of the language, more effort being spent on optimizing Ada (at the time it presumbably had more military users than the C compiler), utilizing lessons-learned from implementing the C compiler earlier, or the superior brain-power of Bevin and everyone else involved with the Ada compiler.

Still, its a nice piece of evidence to have in one's pocket when those "Ada is slow" threads pop up.

From: Laurent Guerby <guerby@acm.org>
Date: 10 May 2001 21:07:40 +0200
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

[...] the reply was that the Ada language specification requires extensive run-time checks which slows down the programs.

You can selectively turns checks off in Ada using language feature, the information you have just shows that the people you've talked to know nothing about Ada. [...]

> I have noticed that there are too few numerical programs in Ada95 for numerical computation is the above fact the real cause for this?

There is some numerical code available in Ada on the web, plus bindings to traditional computing kernels. Most big Ada software is for military or industry which is not widely known and distributed (as opposed to Fortran in the scientific community).

> Finally would you suggest someone to use Ada95 for numerical code?

At work we have 250 KSLOC of Ada 95 doing financial number crunching on a variety of platforms (SGI O2000 64 procs, Linux clusters of various size, Solaris, NT), and Ada definitely does the job fine.

We turned checks off on one tight loop to beat third party code in performance, otherwise we prefer the safety of checks against a 10-20% speed improvement.

From: "N&J" <nikogian@hotmail.com>
Date: Thu, 10 May 2001 23:49:16 +0300
Organization: National Technical University of Athens, Greece
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

First of all I would like to thank everyone for his/her help. From what you said I understand that it is pointless to compare languages in their timing efficiency. Anyway, I am convinced that Ada95 is suitable for numerical computation and I am going to use her. I would also like to refer everyone interested in the subject to:

Jan Verschelde: Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. ACM Transactions on Mathematical Software 25(2): 251-276, 1999.

PHCpack is a large general purpose package which solves systems of polynomial equations (using homotopy continuation methods). In this paper the author explains why he chose Ada for writing this package.

[Jan Verschelde, then at the Department of Computer Science at the K.U.Leuven in Belgium, provided PHC sources and binaries to Ada-Belgium. Check out the web-page "Free Ada Software provided by Belgian Ada users" at URL http://www.cs.kuleuven.ac.be/~dirk/ada-belgium/software/#PHC which also contains a pointer to the latest version of this package. -- dc]

From: "Marin David Condic" <marin.condic@pacemicro.com>
Date: Fri, 11 May 2001 10:03:11 -0400
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

[...] People have done timing comparisons, but you absolutely must remember at all times that you are NOT comparing Ada to Fortran and C/C++. What you are comparing is the relative speed of code generated by two (or more) different COMPILERS. As others have observed, since Fortran programmers tend to want highly optimized math operations, the guys who write Fortran compilers tend to concentrate on this area. That doesn't mean you won't discover that Fortran Compiler X produces slower code than Ada Compiler Y. In point of fact, some Ada compilers are quite good at optimizing math operations.

I think you would also find that the way Ada goes to such great pains to precisely define the behavior of its mathematical types could be a major help to you in developing your algorithms. Get hold of the Ada Reference Manual and look at the definitions of the numeric types and the annexes that relate to math and I think you'll be impressed with the facilities available. (Sometimes it is a bit obscure, so be sure not to miss the parts that talk about all of the attributes available for numeric types. Many are extremely useful and aren't paralleled in other languages.)

From: "Jean-Pierre Rosen" <rosen@adalog.fr>
Date: Fri, 11 May 2001 18:28:07 +0200
Organization: Adalog
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

Another point I'm surprised nobody mentioned: provided your compiler supports Annex F, you get guarantees about the numerical accuracies of computations, including the accuracy of the math library. AFAIK, Fortran tells you nothing about the accuracy of sin, log, etc. So if you are concerned not only by fast results, but also by accurate results, Ada is a big winner.

From: "Marin David Condic" <marin.condic@pacemicro.com>
Date: Fri, 11 May 2001 14:28:50 -0400
Subject: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

[...] When I was a Fortran programmer and started learning Ada, I was very impressed with the fact that Ada seemed to have so much better support for math than did Fortran. (Aside from lack of Log & trig at the time. :-) Certainly, Ada has added lots of new mathematical capabilities in Ada95 only by now I don't know if Fortran has caught up.

At the time, I was very impressed that Ada let you specify the accuracy you required and/or check the accuracy that was available. Ada defined model numbers and safe numbers that gave one a model for understanding how calculations were going to be made & thus one could plan for it. Fortran basically left the mechanics of the computations up to the machine with no rigorous formal definition of behavior. (Or at least it was not nearly as rigorous as the definition Ada provided.) Lots of this is a major aid in numerical analysis when you're trying to figure out how good your computations are going to be.

Now the numerous attributes available for numeric types as well as adding support for decimal & modular types would have me claiming that there is quite a bit more support in Ada for number crunching than one gets with Fortran - but the last Fortran I used was Fortran77. If later versions have caught up, I'd like to know about it, but I'd recommend that Fortran number crunchers take a serious look at Ada's numeric support because they just might find it to be superior. (That, and its hard to beat Generics for building computations once & reusing them with different numeric types!)

From: gdm@popremove.mydiax.ch (Gautier de Montmollin)
Date: Sun, 13 May 2001 21:42:48 +0000 (UTC)
Subject: Re: Re: Numerical Computation and Ada95
Newsgroups: comp.lang.ada

[...] The good surprise is that you can e.g. make a generic sparse matrix package whose instantiations run as fast as their Fortran 77 equivalents for various precisions. I did compare on an Alpha server under OpenVMS, with Ada & F77 compilers from Compaq. It required a bit of tuning before reaching the original, of course, since I coded the sparse matrix type as an unconstrained record with arrays in it, a thing a bit more "clothed" than the arrays passed the in F77 code.

I needed to use that code in a project programmed in Ada, so the question was either to interface to the F77 code or to make an Ada version. In global programmation time it was also a big benefit since I could use later the same code with GNAT on e.g. PC platforms. If you use the correct switches, like -O2, -gnatpn, -funroll-loops (GNAT), you obtain a very nice performance. Things like ":=", "others=>" with big objects are poorly coded by some Ada compilers, but otherwise the job is very well done; abstraction often helps performance.

In addition, in the debugging phase, you compile with the checks ON and you find lots of bugs about array bounds. This is also an absolute plus! BTW, interesting to mention that translations from Fortran very often show latent or undetected bugs like 0 index reached in an array meant to be on 1..N, etc.

Gautier -- http://www.diax.ch/users/gdm/

Software Quality: Why is There so Much "Sucky" Software?

From: "Deller, Steve"
<sdeller@rational.com>
Date: Wed, 21 Feb 2001 21:11:49 -0800
Subject: Re: sucky software (was: The Good and Bad News about Java)
To: team-ada@acm.org

> [...] in my 33 years in industry, I have never known a developer who wanted to produce sucky software (although I have known many managers who did, consciously). Rather, the developers either 1) don't see their software as being sucky -- a training/competence issue -- or 2) they tolerate suckiness in order to meet deadlines, ie, they accept the development schedule along with all the other requirements their product must meet. [...]

When I was a few years into programming, and just starting a job at CSC, I had the VP of the entire division come to me to write a piece of software. He wanted it FAST. I told him I could do it in two weeks. He said he needed it in one and told me to do it QUICK and DIRTY. Now I am known to be stubborn at times :-), and insisted I did not know HOW to make quick and dirty software, only how to develop software correctly. This led to a him eventually ORDERING me (shouting at me) to do it QUICK and DIRTY.

I went off and came back to him when things had cooled off and told him the program was done. He wanted to see it. It immediately failed, since in reality is was the assembly equivalent of

```
begin
  null;
end;
```

He immediately demanded to know what the hell was that. I told him that was DIRTY software. I asked him just HOW DIRTY did he want his software.

The upshot is that he left in a huff and gave the project to someone else that promised it in a week. Two weeks later, I had a proper working version of the software, having worked on it on my own overtime. I told you I was stubborn.

I gave it to the VP. He was happy to have the software, but was understandably cool toward me. However, it turns out the other group that was working on the QUICK and DIRTY version was STILL debugging three weeks later.

I have always been very proud of that interaction. The only difference from then to now is that I've learned to hold the same line without all the heat and argument.

In subsequent experience, I have met managers that understand schedule/quality/capability tradeoffs, managers that understand it and don't want to recall it, and managers that don't understand it. For the latter two catagories, starting with begin;null;end can sometimes lead to a reasonable dialog.

From: "Dale Jr, William"
<william.dale.jr@lmco.com>
Date: Thu, 22 Feb 2001 13:27:30 -0800
Subject: Re: sucky software (was: The Good and Bad News about Java)
To: team-ada@acm.org

My story would go in a similar vain but the second team did get the program built in one week and it did work 'OK' and it made management very happy. I was placed on the list for the next lay-offs and they were promoted.

If you can spit working binary out your left ear then you are management's pet.

Then, three months later they wanted to use it as the basis of a new product and nobody could read it much less figure out how it worked. Project scrapped.

This is the much more common experience ;-(

From: "C. Daniel Cooper"
<cooper@longshot.ds.boeing.com>
Date: Fri, 23 Feb 2001 10:27:23 -0800
Subject: Re: sucky software (was: The Good and Bad News about Java)
To: team-ada@acm.org

I'm sure these stories abound. In my own equivalent, the other team's code was delivered to the customer in half the time, and I was laid off. Their code was "mostly working". [...]

In my case, the customer was *not* impressed: The team kept going back to them with patch after patch. A few months later, the company folded and no longer exists.

But the point of my original email was not to elicit these stories (satisfying as they are); instead, it was an attempt at understanding the phenomenon. Why does "the other team" always seem to exist? Earlier, I wrote,

> the developers either 1) don't see their software as being sucky -- a training/competence issue -- or 2) they tolerate suckiness in order to meet deadlines, ie, they accept the development schedule along with all the other requirements their product must meet.

The common thread in our stories is that we reject the premise of #2: we refuse to accept a deadline that will induce sucky software. But the "other team" does accept it. Are their standards lower than ours? When I talk to these developers, I get a different perspective: they see themselves as "pragmatic", "getting the job done", "not a blue-sky academic", etc. Our view is that "suckiness is intolerable and the deadline is unrealistic"; their view is that "flawless software delivered late is sucky ". They have made a tradeoff that to us is abhorrent: some degree of suckiness is tolerable versus missing the deadline; or more simply, some degree of suckiness is always tolerable.

I really don't like arguing this point, but I think it's a reality that we in the Ada community tend to be blind to. We seem to think (hope) we can eradicate suckiness, via better tools and environments, better training and curricula, better processes and methods, and yes, by using Ada versus other languages. But if all these were in place, would suck-inducing deadlines go away? I think not; they would only get shorter :-(

Am I being overly pessimistic here? Is there some Utopia we can aim for, where sucky software is no longer built? Or, are we just being unrealistic in hoping so, since some degree of suckiness will always be tolerable no matter how things improve? I ask this from a software engineering maturity perspective, subsuming Ada in the question: would our profession be better served by learning to cope with suckiness (rather than just disdaining it), making its assessment an overt part of the engineering process? This seems to be an area that academia doesn't want to address, and industry doesn't want to admit (but that maligned managers seem to accept). For starters, answering these questions will force us to define just what "suckiness" is: as indicated above, the view of the "other team" already disagrees with our view.

C. Daniel Cooper, Adv Computing Technologist, 206-655-3519, CDaniel.Cooper@Boeing.com

From: Alan and Carmel Brain
<aebrain@dynamite.com.au>
Date: Sun, 25 Feb 2001 00:06:31 +1100
Subject: Re: sucky software (was: The Good and Bad News about Java)
To: team-ada@acm.org

Mine is a sort of mixture: my code was delivered on time, the other team's still hasn't been delivered, their code isn't even "partially working" even though the requirements were reduced for them. But my annual performance review gave me the lowest possible rating, "No area satisfactory", and I'm leaving before I get fired for incompetence. The managers doing the review were all members of the other team.

CMM Level -3.
http://www.omaha.org/spin/cimm.htm

From: "S. Ron Oliver" <sroliver@csc.calpoly.edu>
Date: Fri, 2 Mar 2001 10:55:09 -0700
Subject: Re: sucky software (was: The Good and Bad News about Java)
To: team-ada@acm.org

Well, in my "many" years of experience I have often been on projects with "hot dog" programmers who swore by "quick and dirty". Early on these were assembly language hot dogs. More recently they have been C language hot dogs. My experience with these "hot dogs" has been that they invariably fail to deliver their fair share of the work on time. And when they do deliver, their work is invariably sucky. Yet, management tends to be happy with them. My observation is that they are much better at convincing management they do good work than they are at doing good work.

> [...] it was an attempt at understanding the phenomenon. Why does "the other team" always seem to exist?

This is a key question that lead me, almost 10 years ago, to start studying the whole "sucky software" issue. It was very puzzling to me then. I still don't feel I fully understand the phenomenon, but I do understand it a lot better than I did then.

"Hackers" thoroughly love being around computers and software. They are in this business because it is FUN, not because they have any particular commitment to the profession. To make matters worse, they have managed to get into the business without receiving any significant education or training to qualify them to be in the business.

One reason they are able to "impress" management is their deep love for software. Their enthusiasm is often mistaken for competence. At the same time, they are generally very intelligent - not re. software, so much, but in general. They use their intelligence to learn what buttons are most successful with management, and work those buttons skillfully.

There are other aspects to this phenomenon, but I think these are 2 key aspects of it.

[...] Throughout (U.S.) industry there is little evidence that quality is a serious consideration when it comes to "getting the job done".

[...] they genuinely believe software can't be implemented correctly in the first place. If it can't be done right with a LOT of effort, it might as well be done wrong with less effort.

Assuming that management gets proper training, I think deadlines would almost become a non-issue eventually. That is, both developers and managers would focus more on steady, productive progress. And, since far less time (currently something like 80-90 % of developer time) would be spent on maintenance, it would appear that projects are coming in ahead of schedule most of the time. But the detailed development of this argument is quite lengthy.

What we are talking about here is a substantial cultural change, after which "suckiness" not only would not be tolerated, but would not even be an option. This, admittedly, is a BIG change. But we have to start somewhere.

S. Ron Oliver, semi-retired professor of Computer Science and Computer Engineering,
www.csc.calpoly.edu/~sroliver

Tired of sucky software? Check out www.caressCorp.com and follow the links to software sucks and The Oliver Academy.

Don't Believe the Hype

From: Tim Bradshaw <tfb@tfeb.org>
Date: 09 May 2001 18:30:01 +0100
Organization: The Tardis Project
Newsgroups: comp.lang.lisp

> CORBA is not a problem; it's a problem solver. It solves the problem of moving data across the network. It also solves the problem of managing objects, threads, and persistence.

It's this kind of answer that bugs me. These things are open problems which don't *have* general solutions. If CORBA claims to solve them it's lying. But this is what you would expect. Richard Gabriel was wrong: worse is not better, lying is better. Languages and systems succeed in the marketplace to the extent that their proponents lie about what they can do.

Once you realise this you can see it everywhere.

Windows does better than Unix because Windows proponents tell you that it will solve things that your Unix system people keep telling you are hard. The Unix people are right: they are hard, and Windows does not solve them, but by the time you realise that it's too late because it has you by the throat.

Java does well because Sun told all sorts of lies about what it could do and exaggerated the rest. C++ does well because people tell at least two big lies: OOP solves all programming problems and C++ does OOP.

Virtually any language or system does well because its proponents lie about other languages - Lisp people know this only too well. GC is slow, Lisp is an interpreted language - need I go on?

Lisp does badly because we refuse to lie. When people ask us if we can solve insoluble problems we say that we can't, and because they expect us to lie to them, they find some other language where the truth is less respected.

Back to CORBA. If it did what you say then building massively parallel systems would be easy. But it's not easy: in fact no one knows how to do it at all in general, and those who have written massively parallel systems will tell you, at length, just how hard it is.

Don't believe the hype.

The State Of The Embedded Systems Industry

URL: http://www.ddci.com/news_vol2num1.shtml
Subject: DDC-I Online News January 2001 Vol. 2 Issue 1

2000 SIGAda Award Winner Dr. Joyce Tokar Speaks Out On the State Of The Embedded Systems Industry

Phoenix, AZ Dr. Joyce Tokar, who has invested more than fifteen years of research and development in the improvement of embedded systems technology, was recently recognized with the Outstanding Ada Community Contribution Award at the 2000 ACM (Assoc. for Computing Machinery) SIGAda Conference, held November 14th-16th at the Johns Hopkins University Applied Physics Lab in Laurel, Maryland.

While Tokar's chosen field of embedded systems technology, and the Ada programming language, have long been regarded as the stuff of fighter planes and Space Shuttles, today embedded systems play a critical part in the business and consumer technology markets and Ada is playing an increasing role.

"There was discussion about where Ada is going and some surprises at the conference," says Tokar, who serves as the Vice President of Technology at Phoenix-based software development tools provider DDC-I. "Canal+ Technologies uses satellites to deliver interactive TV programming, and they gave a presentation about the ground system that pushes their signals, which was written in Ada. While everybody keeps saying Ada is a military language, this application couldn't be any less military."

Since the lifting of the Department of Defense mandate in 1996 to develop military embedded systems using Ada, the language is asserting its influence in many non-military applications where dependability is paramount. The term "safety-critical" is often used to describe embedded applications where Ada is used, but two distinct measures of the term are emerging. While safety has always been measured in human lives in the military and aerospace world, in a networked world safety is simply a synonym for "uptime," and when a critical network system is down, money burns.

"The language has moved from being used exclusively for military and space applications into realms like high-speed rail, commercial air transportation, process control, and even networking hardware," she says.

Tokar's SIGAda award was the culmination of a steady leadership role in the Ada community, which includes membership in the US Technical Advisor Group to ISO Working Group 9, which is responsible for the languages definition and evolution. In addition, Tokar participated in the International Real-Time Ada Working Group, where she contributed to the definition and standardization of the Ravenscar Profile, a tasking subset of Ada 95 suitable for use in safety-critical, high-integrity real-time systems.

She finds it exciting to finally see the language stand on its own feet: "I see the embedded systems market becoming more and more enamored by high-level programming languages, of which Ada is one. I think there's a lot of discovery going on right now. For assembly language programmers, just moving to C is a big step. Meanwhile, the C++ guys are working hard to figure out how can we do this on an embedded chip. Ada gives you everything that you have in C++ and its been tailored for embedded chips."

According to Tokar, as a direct result, a number of real-time operating systems vendors are announcing "safety critical" versions of their products, due to increasing interest in expanding the development of real-time, safety-critical applications beyond assembly language. Meanwhile, Ada has been providing facilities for safety-critical, real-time high-level application development for many years, but the embedded systems community is only just now beginning to catch on.

Why? "Because the embedded world is absolutely exploding right now," she explains.

Unfortunately, Ada is still struggling with perception issues according to Tokar, despite proven superiority in many applications, military and otherwise. She offers a recent article in EE Times (9/25/2000) as evidence.

"The article was about the programming language choices available to the embedded programmer. It suggests that the primary choices are C, C++ and Java," she says. "It did go on to say that there are difficulties in using Java, due to the lack of standards and the ongoing debate over real-time extensions, and that C and C++ are weak in their support of multi-threaded applications. However, it didn't mention the one language that's standardized (ISO), supports multi-tasking and object-oriented programming, is currently in wide use, and designed and developed specifically for embedded systems. Ada."

Originally designed under DoD guidance for the real-time, embedded systems applications inherent in mission-critical military systems, it was apparent early in Ada's lifetime that the language was also well suited to plenty of safety-critical embedded applications. Today, the Boeing 777 circles the globe with Ada onboard and TGV trains across Europe use an Ada-coded collision avoidance system. With full support for object-oriented programming introduced into a revised Ada standard in 1995, the use of the language in a wider range of commercial embedded applications, like network hardware, was underway.

Supporting a full tasking model within the core definition of the language that's designed to interoperate smoothly with the object-oriented features of the language like dispatching and inheritance, it provides full task synchronization supporting coordinated exchange of data between threads as parameters. It offers data-oriented synchronization to facilitate asynchronous and protected exchange of data between threads. The language standard also includes the definition of specific profiles to support the use of the tasking model in real-time applications, going on to define a domain specific profile for high-integrity systems.

"The article in the EE Times appeared to concentrate on the tools presented at the recent Embedded Systems Conference in San Jose, and there were several Ada vendors at that show," Tokar says. "So, when contemplating the challenge of choosing the right development language for an embedded application, remember that there is a ISO standard language that supports multi-threaded applications within the language definition. Ada is ready when you are."

Butterfly Catcher Game's Tutor has a Very Fine Name!

From: "Thomas A. Panfil" <t.panfil@gte.net>
Date: Sun, 25 Mar 2001 00:57:23 -0500
Subject: Butterfly Catcher Game's Tutor has a Very Fine Name!!
To: team-ada@acm.org

I just sampled a new shockwave game announced by the Macromedia folks, and was very surprised at the name of the girl in the pink dress who is the butterfly catcher. You learn her name by clicking on the "How to Play" link, and reading through the instruction book. She introduces herself, and also signs it at the end.

http://www.shockwave.com/bin/shockwave/entry.jsp?page=/content/loop/loop.html

Does anyone think that there is some real significance to her having this particular name? Might someone on the list actually know?

Tom Panfil, Treasurer, Baltimore SIGAda, http://www.jhuapl.edu/sigada/, Thomas_A_Panfil@acm.org, tapanfil@ieee.org

The Ada Business

From: Emmanuel Briot <briot@gnat.com> [On behalf of Robert Dewar]
Date: Wed, 09 May 2001 08:40:51 GMT
Subject: Re: [ANNOUNCE] XML/Ada 0.5 released
Newsgroups: comp.lang.ada

> It would be interesting to hear your [= ACT's -- dc] former competitor's views on their reasons for abandoning the Ada compiler business. [...]

I do not know of other vendors "abandoning" the Ada business. Far from it, all the ARA vendors report increasing business in the Ada market.

What distinguishes Ada Core Technologies is that we are, as far as I know, the only Ada compiler vendor whose primary business is in the Ada market (and I am happy to report that we can pay our bills just fine :-)

Robert Dewar, Ada Core Technologies

Conference Calendar

This is a list of European and large world-wide events that may be of interest to the Ada community. More information on items marked ♦ is available elsewhere in the *Journal*. The information here is extracted from the online *Conference announcements for the international Ada community* at http://www.cs.kuleuven.ac.be/~dirk/ada-belgium/events/list.html on the Ada-Belgium webserver. These pages contain full announcements, calls for papers, calls for participation, programmes, URLs etc and are updated regularly.

2001

09-11 July **International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA'2001)** Las Vegas, Nevada, USA Includes: sessions on Parallel Computing and Ada; Programming (Ada) and Computational Intelligence.

09-11 July **2nd International Conference on Parallel and Distributed Computing, Applications, and Techniques (PDCAT'2001)** Taipei, Taiwan Topics include: Programming languages; Parallel programming paradigms; Parallelizing compilers; Object-Oriented technology; Tools and environments for software development; etc.

16-17 July **6th International Workshop on Formal Methods for Industrial Critical Systems (FMICS'2001)** Paris, France Topics include: Tools for the design and development of formal descriptions; Verification and validation of complex, distributed, real-time systems and embedded systems; Case studies and project reports on formal methods related projects with industrial participation (e.g. safety critical systems, mobile systems, object-based distributed systems); etc.

18-23 July **Conference on Computer-Aided Verification (CAV'01)** Paris, France

22-25 July **5th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI'2001)** Orlando, Florida, USA. Includes invited sessions on: Software Quality (Standards, Metrics, Models, Tools and Human Aspects); Wireless Software Engineering; etc.

22-26 July **2001 Rational Software User Conference** Denver, U.S.A.

29 July – 3 August **Technology of Object-Oriented Languages and Systems (TOOLS USA'2001)** Santa Barbara, CA, USA. Topics include: O-O verification and testing techniques; Components, frameworks, and reuse; Distributed and intelligent objects and agents; Standardization of languages and methods; Experience reports with O-O technology; etc.

23-24 August **4th Nordic Workshop on Software Architecture (NOSA'2001)** Odense, Denmark. Deadline for submissions: August 5, 2001 (position papers). Deadline for registration: August 5, 2001.

26-28 August **Australian Software Engineering Conference (ASWEC'2001)** Canberra, Australia. Topics include: Specification and verification; Software engineering environments; Component-based software engineering; Software maintenance and evolution; Reuse and re-engineering; Software standards; Empirical studies of software methods and tools; etc.

27-29 August **7th International Conference on Object-Oriented Information Systems (OOIS '2001)** Calgary, Canada Theme: "Object-Oriented and Web-Based Frameworks for Information Systems" Topics include: OOIS Architectures (OO components/COTS, OO patterns, OO middle-ware, OO distributed systems, ...); OOIS Processes (Reuse processes, Maintenance/support processes, ...); Web-based OOIS (On-line courses/tutorial systems, ...); etc.

27-31 August **5th IEEE International Symposium on Requirements Engineering (RE'01)** Toronto, Canada.

28-31 August **European conference on Parallel Processing (Euro-Par'2001)** Manchester, UK. Topics include: Support Tools and Environments; Compilers for High Performance; Distributed Systems and Algorithms; Parallel Programming: Models, Methods and Languages; Object-Oriented Architectures, Tools and Applications; Parallel and Distributed Embedded Systems; etc.

28-31 August **Working IEEE/IFIP Conference on Software Architecture (WICSA'2001)** Amsterdam, The Netherlands

03-07 September **2001 International Conference on Parallel Processing (ICPP'01)** Valencia, Spain. Sponsored by The International Association for Computers and Communications. In cooperation with The Ohio State Univ. and U. Politec. de Valencia. Topics include: Programming Methodologies and Tools, Compilers and Languages, Internet Computing, OS and Resource Management, Network-Based Computing, Web and Multimedia, etc.

03-07 September **6th International Conference on Parallel Computing Technologies (PaCT'2001)** Novosibirsk, Russia Topics include: Languages, environment and software tools supporting parallel processing; Teaching parallel processing; etc.

04-06 September **EUROMICRO Workshop on Software Process and Product Improvement** Warsaw, Poland In conjunction with the 27th EUROMICRO Conference Topics include: Component-based software development; Dependability of software systems; Software engineering standards; etc.

04-06 September **EUROMICRO Workshop on Component-based Software Engineering** Warsaw, Poland In conjunction with the 27th EUROMICRO Conference Topics include: Component Development Processes; Design, Implementation, Testing; Component Specification; Components for Real-time Systems; Case Studies; etc.

04-07 September **Parallel Computing 2001 (ParCo2001)** Naples, Italy.

04-07 September **5th International Enterprise Distributed Object Computing Conference (EDOC'2001)** Seattle, Washington, USA Topics include: Software architectures and component based development for Enterprise systems; Architecture and interoperability issues in large-scale enterprise systems; Modelling, methodologies and technology for component based development; Deployment, operation, maintenance and evolution of Enterprise components and services; Performance, scalability and reliability in Enterprise software; Fault-tolerance and dependable Enterprise systems; Case studies and experience reports; etc.

06 September **ICFP Workshop on Semantics, Applications and Implementation of Program Generation (SAIG'01)** Firenze, Italy. Topics include: Semantics, type systems, and implementations for multi-stage languages; Run-time specialization systems (e.g. compilers, operating systems); Program synthesis from high-level specifications; etc.

08 September **1st Workshop on Multi-Language Infrastructure and Interoperability (BABEL '01)** Firenze, Italy. Topics include: Compilation of high level languages to common executable formats such as Sun's Java Virtual Machine, the DRA's ANDF or Microsoft's .NET Common Language Runtime; Defining and using bindings for component middleware such as OMG's CORBA or Microsoft's COM; Language constructs to support interoperability between different languages, particularly from different paradigms (e.g. OO/functional); Multi-language backends for compilation and/or analysis; Multi-language development environments and tools (e.g. debuggers, profilers); etc.

08-12 September **International Conference on Parallel Architectures and Compilation Techniques (PACT'2001)** Barcelona, Catalunya, Spain Topics include: Parallel architectures and computation models, Compilers for parallel computer systems, Applications and experimental systems studies, Parallel programming languages and algorithms, etc.

09-13 September **3rd International Symposium on Generative and Component-Based Software Engineering (GCSE'2001)** Erfurt, Germany Deadline for submissions: May 14, 2001

10-13 September **3rd International Conference on Product Focused Software Process Improvement (Profes'2001)** Kaiserslautern, Germany Topics include: Quality of Software in Embedded Systems, Telecom and Internet Applications, Mobile Applications and Services; Technology Transfer; Industrial Experiences and Case Studies; etc.

10-14 September **Joint 8th European Software Engineering Conference (ESEC)** and **9th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE-9)** Vienna, Austria. Includes:

10-11 Sept. **ESEC/FSE-9 - International Workshop on Principles of Software Evolution (IWPSE'2001)** Topics include: Evolution of requirements and environments; Methodology for evolutional design and development; Configuration and change management for evolution; Development support environment for evolutional; Experience and lessons learned from evolutional software systems; etc.

11 September: **ESEC/FSE-9 - Workshop on Composition Languages (WCL'2001)** Topics include: Programming paradigms for software composition; Type systems for composition languages; Portability issues; Interoperability issues; Implementation techniques for composition languages; Scalability and extensibility of the language abstractions; Case studies of composition language design; Case studies of applications using composition languages; etc.

16-19 September **Conference on Communicating Process Architectures 2001 (CPA'2001)** Bristol, UK. Topics include: concurrent design patterns and tools; safety and security issues (race-hazards, deadlock, livelock, process starvation,...); language issues; applications: scientific (including graphics and GUIs), engineering (including embedded, real-time and safety-critical), business (including mobile and e-commerce) and home (including entertainment); etc.

17-19 September **3rd IFIP WG 6.1 International Working Conference on Distributed Applications and Interoperable Systems (DAIS'2001)** Krakow, Poland Topics include: Experiences from development of distributed applications; Scalability issues in distributed applications; Experiences with distributed platforms and their scalability; Components and frameworks for distributed applications; etc.

18-20 September **3rd International Symposium on Distributed Objects and Applications (DOA'2001)** Rome, Italy Topics include: Design patterns for distributed object design; Database services, in particular persistency, transaction, query and replication services; Integration of distributed object and Web technologies; Interoperability-supporting environments; Methodologies to develop distributed object applications; Reintegration of legacy systems in DOC environments; Design of CORBA, COM- and Java-based broker applications; Reliability, fault-tolerance and recovery; Real-time ORB middleware; Reports on Best Practice; etc.

25-28 September **Technology of Object-Oriented Languages and Systems (TOOLS Eastern Europe 2001)** Varna, Bulgaria. Theme: "Emerging Technologies, Emerging Markets" Topics include: Theory and practice of object-oriented software construction; Processes, methods and tools of object technology in practical business applications; Software reuse: principles and reality of software asset management; Patterns in analysis and design; Examples of successful implementations of object technology; etc.

♦ 30 Sept. – 04 October **2001 ACM SIGAda Annual International Conference (SIGAda'2001)** Twin Cities, Minnesota, USA (approval pending by ACM)

01-05 October **4th International Conference on the Unified Modeling Language (UML'2001)** Toronto, Canada Theme: "Modeling Languages, Concepts and Tools"

02-05 October **8th Working Conference on Reverse Engineering (WCRE'2001)** Stuttgart, Germany Topics include: Experience reports (successes and failures) on reverse engineering or reengineering efforts; Techniques, tools, and enabling technologies for reengineering, reverse engineering, renovation, reuse, and migration; Software visualization; Software evolution and reengineering; Integration of reverse engineering and forward engineering; Code-based management systems to support reverse engineering; Wrapping and interfacing legacy systems; Formal methods in reverse engineering; etc.

03-05 October **15th Brazilian Symposium on Software Engineering (SBES'2001)** Rio de Janeiro, Brazil. Topics include: Industrial applications of Software Engineering; Component-based Software Engineering; Methods, Techniques, Languages and Tools for Software Engineering; Software Maintenance; Software Quality; Software verification, validation and testing; etc.

08-12 October **25th Anniversary Annual International Computer Software and Applications Conference (COMPSAC'2001)** Chicago, Illinois, USA Theme: Invigorating Software Development Topics include: Component-based software development; Object-oriented technology; Safety and security; Software reliability; Distributed systems; Embedded systems; Internet and Web-based systems; Middleware systems; etc.

10-12 October **European Software Process Improvement Conference (EuroSPI'2001)** Limerick Institute of Technology, Ireland.

14-18 October **16th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'2001)** Tampa Bay, Florida, USA Deadline for submissions: July 20, 2001 (Posters, Demonstrations, Doctoral Symposium, and Student Volunteers)

17-19 October **Colloque Francophone sur la Modelisation des Systemes Reactifs (MSR '2001)** Toulouse, France

♦ 24 October **Symposium on Reliable Object-Oriented Programming (SROOP)** London, UK. OO has become a key feature of system design and implementation. Reliable systems are increasingly using OO techniques, often replacing traditional structured approaches. Programming languages like Ada 95, Java and C++ each offer subtly different ways of representing objects, their attributes and their methods. How can those features be used reliably? Are there features that are unsafe or inappropriate? What architectures can be employed to make implementation easier or more verifiable? Where do patterns and frameworks fit in?. This Symposium will address many of these issues and more.

28-31 October **20th IEEE Symposium on Reliable Distributed Systems (SRDS'20)** New Orleans, USA Topics include: Distributed systems with reliability, availability, security, safety, and/or real-time requirements; Security and High Confidence Systems; Formal methods and foundations for reliable distributed computing; Distributed objects and middleware systems; Distributed and Web-based application systems; etc. Deadline for submissions: April 2, 2001

06-10 November **IEEE International Conference on Software Maintenance (ICSM'2001)** Florence, Italy Theme: "Systems and Software Evolution in the era of the Internet" Topics include: Design for maintenance; Internet and distributed systems; Software reusability; Tools and environments; Commercial off-the-shelf (COTS); Freeware and open source applications; Programming languages; etc.

12-16 November **5th International Internet & Software Quality Week Europe** Brussels, Belgium. Theme: "Internet NOW!" Topics include: Productivity and Quality Issues; Process Improvement; Real-Time Software; Object Oriented Testing; Application of Formal Methods; Cost/Schedule Estimation; Software Reliability Studies; E-Commerce Reliability; Quality of Service (QoS); Risk Management; etc.

12-16 November **IFIP/ACM International Conference on Distributed Systems Platforms (Middleware'2001)** Heidelberg, Germany Topics include: integration of middleware platforms with web and Java technologies; real-time middleware platforms including real-time ORBs; reliable middleware platforms including fault-tolerant ORBs; applications of middleware technologies including telematics and commerce; distributed systems management and interactive configuration and development tools; etc

22-23 November **6th Australian Workshop on Requirements Engineering (AWRE'2001)** Sydney, Australia. Deadline for submissions: August 14, 2001.

26-29 November **16th IEEE International Conference on Automated Software Engineering (ASE '2001)** San Diego, USA.

28 Nov. – 01 Dec. **12th International Symp. on Software Reliability Engineering (ISSRE'2001)** Hong Kong.

03-06 December **22nd IEEE Real-Time Systems Symposium (RTSS'01)** London, UK. Topics include: embedded systems, software engineering, programming languages and run-time systems, middleware systems, design and analysis tools, formal methods, case studies, applications, etc. Including:

03 December **Workshop on Real-Time Embedded Systems**

04-07 December **8th Asia-Pacific Software Engineering Conference (APSEC'2001)** Macau, China. Topics include: component-based design techniques, concurrent systems, design patterns, distributed systems, formal methods, object-oriented analysis and design, programming languages, reactive and real-time embedded systems, reengineering and reverse engineering, software. quality, software reusability, software maintenance, software engineering for the Internet and the E-commerce, software engineering education, tools and environments, etc.

10 December

Birthday of Lady Ada Lovelace, born in 1815 – Happy Programmers' Day!

10-11 December

2nd Asia-Pacific Conference on Quality Software (APAQS'2001) Hong Kong. Topics include: Economics of software quality and testing; Performance and robustness testing; Quality evaluation of software products and components; Reliability; Review, inspection, and walkthrough; Software quality education; Static and dynamic analysis; Testing of object-oriented systems and real-time systems; Validation and verification; Application areas such as component-based systems, distributed systems, embedded systems, information systems, etc.

17-20 December

8th International Conference on High Performance Computing (HiPC'2001) Hyderabad, India. Topics include: Parallel Languages & Compilers; Distributed Systems; Programming Environments; Embedded Systems; etc.

2002

16-18 January

27th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'2002) Portland, Oregon, USA.

23-25 January

8th International Conference on Languages and Models with Objects (LMO'2002) Montpellier, France. Topics include (in French): Programmation par objets (Langages, interpretation, compilation; modeles d'objets pour la programmation; objets et types; environnements de programmation; etc.); Composants et objets en reseau (Modeles de composants a objets; interactions de composants; developpement a base de composants, composants reutilisables; objets et composants distribues, repartis; acteurs, parallelisme; objets et internet; interoperabilite); Genie des objets (Cycle de vie des objets; retro-conception, evolution des programmes, versions; surete des programmes, specifications formelles; methodes d'analyse et de conception objet, UML; ingenierie des modeles et des meta-modeles; reutilisation, architectures logicielles reutilisables et a base de composants; hierarchies, frameworks, patterns); Applications; etc. Deadline for submissions: August 31, 2001

10-13 March

2002 ACM Symposium on Applied Computing (SAC'02) Madrid, Spain. Deadline for submissions: September 1, 2001 (papers, tutorials)

20-22 March

5th IFIP International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS'2002) Twente, The Netherlands. Topics include: Specification and analysis techniques for distributed systems; Semantics of object-based programming languages; Design and software life-cycle of object-based distributed applications; Applications to telecommunications and related areas; etc. Deadline for submissions: September 1, 2001.

14-15 March

2nd International Software Process Improvement and Capability determination Conference (SPICE'2002) Venice, Italy. Deadline for submissions: September 14, 2001

♦ 09-12 April

11th International Real-Time Ada Workshop (IRTAW'2002) Mont-Tremblant, Quebec, Canada. Deadline for position paper submissions: November 1, 2001.

13 April

ETAPS2002 – 2nd Workshop on Language Descriptions, Tools and Applications (LDTA'2002) Grenoble, France. Deadline for paper submissions: December 15, 2001.

17-21 June

8th International Symposium on Software Metrics (Metrics'2002) Ottawa, Canada. Theme: "Measuring and Managing Software Risks in the Age of Internet" Deadline for submissions: August 15, 2001 (abstracts), August 31, 2001 (full papers).

♦ 17-21 June

7th International Conference on Reliable Software Technologies - Ada-Europe'2002 Vienna, Austria. Sponsored by Ada-Europe, in cooperation with ACM SIGAda (approval pending) Deadline for paper, tutorial, workshop, poster submissions: October 31, 2001.

10 December

Birthday of Lady Ada Lovelace, born in 1815 – Happy Programmers' Day!

ACM SIGAda Annual International Conference – SIGAda 2001

Twin Cities, Minnesota

30 September - 4 October 2001

The Twin Cities of Minneapolis and St. Paul Minnesota have been selected to host this year's annual international SIGAda conference. The Twin Cities have a long history of applying the Ada language. An early Ada project was done in Minneapolis at Honeywell Systems & Research Center as part of the language definition phase in the early 1980s. Today, the upper Midwest continues to see significant Ada work being done at companies such as United Defense in Minneapolis, Minnesota and Rockwell Collins in Cedar Rapids, Iowa. The conference will showcase the region and the support for Ada as well as expose the usage of Ada around the world.

Constructing reliable software is an engineering challenge. The application of methods, tools, and languages interrelate to make the challenge easier or more difficult. This conference focuses on the interaction between these three aspects of software engineering, especially how features in a language such as Ada drive the tools, methods, and ultimately correctness, reliability, and quality of the resulting software. Papers have been solicited that analyze Ada with respect to these factors or in comparison with other languages. This conference will gather industrial experts, educators, software engineers, and researchers interested in developing and testing reliable software. Technical papers and experience reports have been solicited in the following areas:

- Reliability needs and styles
- Safety and high integrity issues
- Standards
- Use of ASIS for new Ada tool development
- Relationships between Ada and real-time Java
- Use of Real-Time CORBA
- Use of the Ada Distributed Systems Annex
- Process and quality metrics
- Mixed-language development
- Ada education
- Real-time networking/quality of service guarantees
- Fault tolerance and recovery
- Distributed system load balancing
- Performance analysis
- Debugging complex systems
- Integrating COTS software components
- Testing and validation
- Information Assurance

Although the deadline for technical papers and experience reports has passed, we are interested in your participation in the following categories**:**

Workshops and Birds of a Feather (BOFs) are focused work sessions, which provide a forum for knowledgeable professionals to explore issues, exchange views, and perhaps produce a report on a particular subject. We are currently soliciting workshops and BOFs for SIGAda 2001. The focused workshop has a defined goal; a report is expected which will be published in Ada Letters. The BOF gives you the opportunity to discuss things of importance in an informal venue; no report is expected. Those interested in proposing a workshop or BOF should contact the SIGAda 2001 Conference Workshops Chair, James E Hassett <hassett@acm.org>. Workshop and BOF proposals should contain a title as well as a short abstract identifying its goal/purpose/outcome.

Panel Sessions gather a group of experts on a particular topic who present their views and then exchange views with each other and the audience.

Posters provide a different forum for both completed work and work in progress. Poster proposals should be 1-2 pages in length. We particularly encourage students to showcase their work as a poster at SIGAda 2001.

Those interested in proposing a panel session or poster should contact the SIGAda 2001 Conference Program Chair, John McCormick <john.mccormick@uni.edu>.

Vendor Participation, including presentations from vendors on their products and services. For specific information, please contact: Hal Hart <Hal.Hart@ACM.org>
Please submit questions on the conference to the Conference Chair, Paul Stachour <Paul.Stachour@stachour.com>.

SIGAda 2001 is sponsored by ACM SIGAda, in cooperation with Ada Europe, SIGAPP, SIGCAS, SIGCSE, SIGPLAN, and SIGSOFT

The conference is shaping up to be exciting and extremely valuable to those developing systems in Ada. This is a good time to make your plans to attend!

For additional information and latest updates, please visit the Conference website:

http://www.acm.org/sigada/conf/sigada2001/

Symposium on Reliable Object-Oriented Programming (SROOP)

Wednesday 24th October 2001
The Institution of Electrical Engineers, Savoy Place, London

***Question*: What has object-oriented design and programming to offer the implementation of reliable systems?**

***Answer*: A lot!**

OO has become a key feature of system design and implementation. Reliable systems are increasingly using OO techniques, often replacing traditional structured approaches. Programming languages like Ada 95, Java and C++ each offer subtly different ways of representing objects, their attributes and their methods. How can those features be used reliably? Are there features that are unsafe or inappropriate? What architectures can be employed to make implementation easier or more verifiable? Where do patterns and frameworks fit in?

This Symposium will address many of these issues and others. The audience will consist of programmers, system designers, tool vendors, managers responsible for tools and languages, and those involved in the assurance of quality in reliable systems. There will be a healthy mix of principles and practice.

We invite contributions on any of the following themes:

- object-oriented design and programming
- system reliability
- system architectures
- patterns and frameworks
- design languages and notations supporting reliable design
- applications of the above particularly in areas such as user interfaces and Internet programming

The organisers are particularly interested in reports of experience gained in *applying* the above.

Call for Papers

Authors are invited to submit abstracts (recommended length no more than one side of A4) addressing the above themes. The abstracts, which will be refereed, should be submitted by email to Rod Chapman at rod@praxis-cs.co.uk. Acceptable formats are Microsoft Word, HTML or plain ASCII. The deadline for submissions is Monday 18th June**. Authors will be notified approximately four weeks later. Accepted abstracts will be distributed to delegates at the symposium. Selected full papers will be published after the Symposium in the *Ada User Journal*.**

Further Information: SROOP Administrator, PO Box 322, York, YO10 3GY, UK
Phone: +44-(0)1904-412740
Fax: +44-(0)1904-426702
E-mail: admin@adauk.org.uk
Website: www.adauk.org.uk

Organising Committee: Jim Briggs, University of Portsmouth (Chairman); Rod Chapman, Praxis Critical Systems; Bill Taylor, Rational Software Ltd; Brian Tooby, BAE SYSTEMS Avionics Ltd; Helen Byard, Symposium Administrator.

Call for Participation

11th International Real-Time Applications Workshop IRTAW 11

9 -12 April 2002, Mont-Tremblant, Quebec, Canada

Over the last decade and 1/2, the International Real-Time Ada Workshops have provided a focus for identifying issues with Ada 83 and 95, proposing solutions for those problems and evaluating proposed language changes.

Since the standardization of Ada95, the International Real Time Ada workshops have assisted in the review of the real time portions of the Guidance in the Use of the Ada Programming Language for High Integrity Systems, and has developed the Ravenscar Tasking Profile.

With the advent of Java and the development of Real Time specifications for Java, the workshop has begun to consider the integration of embedded Ada and Java systems, and their interoperability.

The goals of the 11th IRTAW are to:

- examine and develop paradigms for using Ada 95 for real-time single processor, multiprocessor and distributed systems (including issues of hard and flexible scheduling);
- consider reports on experiences with using Ada 95 on actual real-time projects;
- identify the benefits and impacts of using object-oriented programming in multi tasking (potentially distributed) real-time systems;
- explore the use of Ada 95 in developing multi-tasking components which are resilient to software design errors and hardware failures;
- refine criteria for the use of Ada 95 in high integrity systems, especially those with real-time or embedded attributes;
- review the interactions between exceptions in Ada with those from other languages such as Java and C++;
- examine the issues around the interoperatility of Ada and real time Java embedded systems.

Participation at the Workshop is by invitation following the submission of a Position Paper addressing one or more of the above topics by 1 November 2001. Position papers should be between five and ten pages. All accepted papers will appear in the Workshop Proceedings which will be published as a special edition of Ada Letters.

How to submit:

Please submit your Position Paper to the Program Chair, Joyce Tokar, <tokar@attglobal.net> preferably in Word Perfect or Word format. Or mail four paper copies to

Dr. Joyce L. Tokar
PO Box 1352
Phoenix AZ 85001-1352
USA

All the papers must be written in English. The top of the first page of the paper should include the title of the paper, the author(s)'s name(s), position, organization, address, telephone number(s), fax number(s), and the email address for the author responsible for correspondence.

Schedule:

Receipt of Position Paper	1 November 2001
Notification of Acceptance	15 December 2001
Final Copy of Position Paper	1 February 2002
Workshop Date	9-12 April 2002

CALL FOR PAPERS

7th International Conference on Reliable Software Technologies – Ada-Europe 2002

17 – 21 June 2002, Vienna, Austria

http://www.ada-europe.org/conference2002.html

Conference Chair

Gerhard H. Schildt
Technical University Vienna
Dept. of Computer-Aided Automation
Treitlstr. 1-3
A-1040 Vienna, Austria
Schildt@auto.tuwien.ac.at

Program Co-Chairs

Johann Blieberger
Technical University Vienna
Dept. of Computer-Aided Automation
Blieberger@auto.tuwien.ac.at

Alfred Strohmeier
Swiss Fed. Inst. of Technology Lausanne
Software Engineering Lab
CH-1015 Lausanne EPFL, Switzerland
Alfred.Strohmeier@epfl.ch

Tutorial Chair

Helge Hagenauer
University of Salzburg
Dept. Comp. Science & System Analysis
hagenau@cosy.sbg.ac.at

Exhibition Chair

Thomas Gruber
Austrian Research Centers Seibersdorf
Phone +43-5-0550-4106
Fax +43-5-0550-4199
thomas.gruber@arcs.ac.at

Publicity Chair

Dirk Craeynest
OFFIS nv/sa & K.U.Leuven
Weiveldlaan 41/32
B-1930 Zaventem, Belgium
Dirk.Craeynest@cs.kuleuven.ac.be

Local Organization Chair

Bernd Burgstaller
Technical University Vienna
Dept. of Computer-Aided Automation
Treitlstr. 1-3
A-1040 Vienna, Austria
Burgstaller@auto.tuwien.ac.at

In cooperation with acm SIGAda (approval pending)

TU
TECHNISCHE UNIVERSITÄT WIEN

General Information

The 7th International Conference on Reliable Software Technologies (Ada- Europe 2002) will take place in the year 2002 in Vienna, Austria. 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.

Schedule

31 October 2000	Submission of papers, extended abstracts and proposals for tutorials and workshops
10 January 2002	Notification to authors
10 February 2002	Full papers required for accepted extended abstracts
10 March 2002	Final papers (camera-ready) required
17-21 June 2002	Conference

Topics

The conference will provide an international forum for researchers, developers and users of reliable software technologies. Presentations and discussions will cover applied and theoretical work currently conducted to support the development and maintenance of software systems. Participants will include practitioners and researchers from industry, academia and government. There will be a special session on embedded systems, including the use of Ada in this realm.

For papers, tutorials, and workshop proposals, the topics of interest include, but are not limited to:

- **Embedded Systems** (special session).
- **Management of Software Development and Maintenance**: Methods, Techniques and Tools.
- **Software Quality**: Quality Management and Assurance, Risk Analysis, Program Analysis, Verification, Validation, Testing of Software Systems.
- **Software Development Methods and Techniques**: Requirements Engineering, Object-Oriented Technologies, Formal Methods, Software Management Issues, Re-engineering and Reverse Engineering, Reuse.
- **Software Architectures**: Patterns for Software Design and Composition, Frameworks, Architecture-Centered Development, Component and Class Libraries, Component Design.
- **Tools**: CASE Tools, Software Development Environments, Compilers, Browsers, Debuggers.
- **Kinds of Systems**: Real-Time Systems, Distributed Systems, Fault-Tolerant Systems, Information Systems, Safety-Critical and Secure Systems.
- **Applications** in Multimedia and Communications, Manufacturing, Robotics, Avionics, Space, Health Care, Transportation, Industry.
- **Ada Language and Tools**: Programming Techniques, Object-Oriented
- **Ada Experience Reports**: Experience Reports from Projects using Ada, Management Approaches, Metrics, Comparisons with past or parallel Experiences in non-Ada Projects.
- **Education and Training.**
- **Case Studies and Experiments.**

Program Committee

Ángel Álvarez, Technical University of Madrid, Spain
Lars Asplund, Uppsala University, Sweden
Neil Audsley, University of York, UK
Stéphane Barbey, Paranor AG, Switzerland
John Barnes, UK
Guillem Bernat, University of York, UK
Ben Brosgol, Ada Core Technologies, USA
Bernd Burgstaller, Technical University Vienna, Austria
Ulf Cederling, Växsjö University, Sweden
Roderick Chapman, Praxis Critical Systems Limited, UK
Paolo Coppola, INTECS HRT, Italy
Dirk Craeynest, OFFIS nv/sa & K.U.Leuven, Belgium
Alfons Crespo, Universidad Politécnica de Valencia, Spain
Peter Dencker, Aonix GmbH, Germany
Raymond Devillers, Université Libre de Bruxelles, Belgium
Jesús M. González-Barahona, Universidad Rey Juan Carlos, Spain
Michael González Harbour, Universidad de Cantabria, Spain
Helge Hagenauer, University of Salzburg, Austria
Günter Hommel, Technische Universität Berlin, Germany
Jan van Katwijk, Delft University of Technology, the Netherlands
Hubert B. Keller, Forschungszentrum Karlsruhe, Germany
Yvon Kermarrec, ENST Bretagne, France
Jörg Kienzle, Swiss Fed. Inst.of Technology Lausanne, Switzerland
Fabrice Kordon, Université P.& M. Curie, France
Albert Llamosí, Universitat de les Illes Balears, Spain
Kristina Lundqvist, MIT, USA
Franco Mazzanti, Ist. di Elaborazione della Informazione, Italy
John W. McCormick, University of Northern Iowa, USA
Pierre Morere, Aonix, France
Laurent Pautet, ENST Paris University, France
Erhard Plödereder, University of Stuttgart, Germany
Juan A. de la Puente, Universidad Politécnica de Madrid, Spain
Gerhard Rabe, TÜV Nord e.V., Hamburg, Germany
Jean-Marie Rigaud, Université Paul Sabatier, France
Jean-Pierre Rosen, Adalog, France
Gerhard H. Schildt, Technical University Vienna, Austria
Bernhard Scholz, Technical University Vienna, Austria
Edmond Schonberg, New York University & ACT, USA
Tullio Vardanega, European Space Research and Technology Center, the Netherlands
Stef Van Vlierberghe, Eurocontrol CFMU, Belgium
Andy Wellings, University of York, UK
Ian Wild, Eurocontrol CFMU, Belgium
Jürgen Winkler, Friedrich-Schiller-Universität, Germany
Thomas Wolf, Paranor AG, Switzerland

9 May 2001, BLIEB/AS, 08/A4

Submissions

Authors are invited to submit original contributions. Submissions should be in English. An extended abstract (4-6 pages) or, preferably, the full paper (up to 12 pages) should be sent using the Web submission form. For more information please see the conference Web page. Submissions should be in PDF, Postscript or ASCII format, and follow the LNCS instructions (see 'Proceedings' below). Submissions by other electronic formats, such as a word processor source file, or by fax are not accepted. The Web submission form is the preferred procedure. However, if you don't have access to the Internet, or you don't have an appropriate Web browser, you may send your extended abstract or paper by e-mail to the Program Co-Chair Johann Blieberger. If electronic submission is not available, please send five paper copies. For e-mail or paper submissions, the body of the electronic message or the first page should identify the submission as a paper or extended abstract, and should include in plain text: the title; name, current affiliation, postal address, e-mail address, telephone and fax of each author; the name of the designated contact person; a short abstract; and a list of keywords ordered by relevance, including, whenever possible, topics and subtopics taken from the list of topics of the conference. If the paper is submitted in Postscript, please be sure to select the option "optimize for portability" in your printer driver. To enable publication of the accepted papers before the conference, strict adherence to the dates of the schedule is essential. Authors of extended abstracts must be prepared to submit a full version of their paper before February 10, 2002.

Proceedings

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. See the LNCS Authors Instructions page for guidelines (http://www.springer.de/comp/lncs/authors.html).

Awards

There will be honorary awards for the best paper and the best presentation.

Call for Tutorials

A tutorial should address any of the topics of the theme of the conference. A tutorial will last a half or full day. The proposals should include a title, an abstract, a description of the topic, a detailed outline of the presentation, a description of the presenter's teaching experience in general and with the proposed topic, duration (half day or full day), level of the tutorial (introductory, intermediate, or advanced), expected audience experience and background. Proposals should be submitted by e-mail to the Tutorial Chair.

Call for Workshops

Half- and full-day workshops can be held to address timely issues or to initiate a longer term effort on a topic of interest. Proposals should be submitted by e-mail to the Program Co-Chair Alfred Strohmeier.

Exhibition

The conference will be accompanied by a three-day commercial exhibition on June 18, 19 and 20. Vendors of software products and services should contact the Exhibition Chair at their earliest convenience for further information and to ensure their inclusion.

The 2001 Ada-Europe General Assembly – An Informal Report

John Barnes

Ada-Europe

Abstract

The 2001 General Assembly of Ada-Europe took place on Tuesday, May 15th during the annual conference which was held at Leuven, Belgium. Formal minutes will be distributed to members in due course, but meanwhile here is an informal report on the assembly.

Preliminaries

The meeting started, as is customary, with a check by the Secretary (Erhard Plödereder) on the number of members present and their entitlement to vote. Seven associate members were present as was the case for the previous two years, namely Ada-Belgium, Ada Germany, Ada France, Ada-Spain, Ada in Sweden, Ada in Switzerland, and Ada UK. There was also one direct member, Currie Colket. The total number of members was 343 of which 332 were represented at the meeting.

It was noted by the Secretary that the number of members from all countries was now stable.

The agenda was accepted unanimously. The minutes of the previous meeting (in Potsdam) were approved without discussion. And there were no matters arising.

President’s Report

John Barnes opened by expressing the Board's deep regret at the untimely death of Björn Källberg. Björn had been our Treasurer for many years. He had served both the Board and the Ada community well and would be very much missed both as a friend and as a colleague.

The President then reported on the various conferences which were now going well. The conference last year in Potsdam had been a resounding success both in financial terms and as a technical and social event. The number of attendees was up on previous years and it had made a useful profit against a budgeted loss of 7000 Euros. The President thanked the organizers, Peter Dencker and Hubert Keller for their efforts and the members applauded.

The 2001 conference here in Leuven was also clearly a success and was going to make a significant profit as against the neutral budget. This was going to allow us to make a Surprise Christmas Gift to all members.

Plans for the 2002 conference were progressing well. It will be held in Vienna (Austria and not Virginia as someone incorrectly stated!) and the organizer is Johann Blieberger. The date is the week commencing 17th June. The budget has already been approved and the contract signed.

In 2003, the conference will be in Toulouse, France under the guidance of Jean-Pierre Rosen and Agusti Canals.

The Journal was again jointly produced with Ada UK. At the General Assembly last year, it had been noted that it was proving to be rather expensive partly because of the shift in pound/euro exchange rate and also because the layout meant that the number of pages was more than originally planned.

The agreement with Ada UK has been renegotiated and printing has recently been moved to Spain. There is also a new Editor (Neil Audsley of York) and the formatting has been redesigned both in order to save space and also to permit authors to submit correctly formatted text electronically. The overall savings should be substantial with annual costs below 7000 Euros as opposed to previous budgets of 10000 Euros. The opportunity has also been taken to redesign the cover. Michael Gonzalez provided us with three choices, one was admirably acceptable, one was a bit boring and the third was just too garish. We chose the acceptable one.

So the costs are down and the layout is improved but unfortunately the first issue under the new arrangements has been delayed for a number of reasons. The Board will make every effort to get things back on schedule as soon as possible.

John Barnes concluded his discussion of the Journal by thanking Michael Gonzalez for all his efforts in obtaining the new cover design and establishing the new production mechanism. The members applauded. Although Michael was leaving the Board this year, it was good to know that he would continue to be involved with the Journal and to serve on the Editorial Board.

The President then turned to finances and in the absence of a Treasurer, presented the Treasurer's report.

Treasurer’s Report for 2000

The key point was that the final outcome was excellent. A budgeted deficit of 22500 Euros had been replaced by an actual deficit of less than 5000 Euros. The main reason for this was the financial success of the Potsdam conference plus late revenues from the Santander conference. In addition, costs of the Board and Secretariat were down. On the other hand financial costs were higher than anticipated because of extra work on taxation relating to the old Edinburgh account; in addition there was a late expenditure item from the 1998 conference caused by an overdue expense item. But these two negative items were clearly exceptional and would not occur in future.

The President concluded by pointing out the strong contrast with the situation a few years ago when it was clear that the then annual deficit (if continued) would result in disaster within a few years whereas the current position was sustainable for a long time with the substantial reserves we still had.

The President's and Treasurer's reports were approved.

Workplan, Budget and Fees for 2002

The President gave a brief overview of the Workplan for 2002. It was generally as before but with some important special items.

In view of the excellent results from 2000 and the anticipated profits from Leuven it was proposed that the planned fee increase for 2002 of 5 Euros be postponed for a year.

Moreover, as mentioned above, it was also proposed to give a Surprise Gift to all members. This would be a copy of the Springer reprint of the Ada95 Reference Manual incorporating the changes made by the Technical Corrigendum that was approved by ISO earlier this year. In order to allow this gift to be made before Christmas a revised budget for 2001 was presented.

The workplan also noted that we could make substantial savings by electing auditors from among the members and completing the tax return ourselves.

The President then presented the budget for 2002. This showed an overall deficit of 3500 Euros. However, it was cautious and still included provision for accountancy fees of 2500. Moreover, the provision for the Journal of 8000 was generous in view of our estimates from the first issue under the new arrangements that it ought to be less than 7000. Taking these two items into account the budget would be neutral.

The revised budget for 2001 and the workplan and budget for 2002 were then approved.

Election of Auditors

The question of auditing was then discussed in detail. The proposed model was that each year one of the associate members take responsibility for carrying out the audit for one year only and presenting their findings at the next General Assembly.

Ada Spain expressed willingness to carry out the audit on the books for 2001. The President thanked them for volunteering and the General Assembly then passed an appropriate motion.

Election of Board Members

There were three board members whose two-year period had expired. They were Michael Gonzalez, Erhard Plödereder and Dirk Craeynest. Of these, Erhard and Dirk were willing to be reelected, but Michael was unwilling because of the pressure of other work. Moreover, the sad death of Björn Källberg meant that there was a further vacancy to be filled as well.

The President pointed out that, as indicated in his covering letter distributed with the agenda, he recommended that the Board have seven members next year in view of the fact that a number of roles would be changed. The members accepted this recommendation and so there were three vacancies to be filled by new board members.

By good fortune three new candidates had been nominated. They were Alejandro Alonso (Spain), Janet Barnes (UK), and Kristina Lundqvist (Sweden). There was thus no need for a formal vote and so all the candidates were elected by acclamation.

Election of President

Last year, John Barnes had indicated that he wished to serve as President for only one more year and so his term as President had now expired. Accordingly, it was necessary to elect a new President. He stated that he had much enjoyed his nine years as President and thanked the other members of the Board for their support.

Erhard Plödereder had been nominated by both Ada UK and Ada Germany. There being no other candidates, he was elected by acclamation. John Barnes then handed over the chair to the new President, Erhard Plödereder, and wished him every luck for the future.

Erhard thanked the General Assembly for their confidence in electing him. He then asked the members to pass a motion thanking John Barnes for his many years of service as President. The members approved and applauded.

Other Business

There was no other business and so Erhard Plödereder, the President, closed the meeting.

The Contribution of the Ada Language to System Development: A Market Survey

Ian Gilchrist

IPL Information Processing Ltd., Eveleigh House, Grove St., Bath BA1 5LR UK
Tel.: +44-1225-475114; Fax.: +44-1225-444400; Email: iang@iplbath.com

1. The Survey

This small study, commissioned by Ada (UK), was prompted by questions raised at the 1999 AdaUK Technology Update. A previous Ada (UK) study - "Attitudes to Ada" was well received, but the delegates wished to know if the Ada language really made a significant contribution to the efficiency of the software development process.

With this aim in mind, it was decided that a number of establishments involved in the development of High Integrity Software be approached. All these development departments have kept some level of statistics about their development process and the objective was to analyse the project data and attempt to reach some conclusion about the particular contribution of the Ada language. It would have been nice to compare the performance of Ada with the only other worthwhile development languages, i.e. C/C++. However, all the projects offered for study by the developers were written in Ada and there was no opportunity to test the performance of C as a development language. Maybe this could be the subject of some future study.

The questionnaires are recorded in Appendix A and as with the previous study various comments about Ada and C are recorded unattributed in Appendix B.

2. The Sample

2.1 Companies Interviewed - Question A1

Nine interviews were conducted altogether; seven with defence companies, one with a civil aviation company and one with the Ministry of Defence. Metrics from twelve projects were collected and contribute to the study’s findings. The interview conducted with the MoD Procurement Division did not involve specific projects but served to provide a useful check and balance on the statistics being provided by the application developers.

For reasons of security and company confidentiality the project statistics provided were understandably somewhat limited and a number of answers to the questions posed by the questionnaire were deduced during and subsequent to the interview rather than spelt out in the numbers provided. Nevertheless, considering the difficulties involved, all the companies were very forthcoming and helpful.

2.2 Types of Application - Question A2

Without exception all the applications considered were Real Time Embedded Systems so it is not surprising that Ada dominated the proceedings. In the case of the defence contractors the language was mandated, of course, and although the mandate no longer applies most of the follow up work stems from these mandated projects (the European Fighter Aircraft contracts are a prime example). So Ada remains the development language of defence and avionics. Although at the time, the mandate was much resented, it will be seen later that not one of the contractors, with the benefit of hindsight, would have used an alternative language.

In terms of safety and mission criticality the developers were asked to state the SIL Level. As can be seen in Table 1 only 3 of the projects were non-critical and 9 were to some extent mission critical and higher.

The survey also asked the Ada standard used (Ada 83 or Ada 95) and if the Ada SPARK subset was used.

Project Identity	Application Type	SIL	Ada * = SPARK
cm	Electronic Surveillance and Counter Measures	0	Ada 83
ps	Missile Guidance	0	Ada 83
ad	Air Defence Command and Information	0	Ada 83
pm	Missile Guidance	0 & 3	Ada 83 *
tr	Armoured Vehicle Fire Control	2	Ada 95
ls	Airborne Early Warning	2	Ada 95
sm	Avionics - Stores management	2 & 3	Ada 83
es	Avionics - Stores Management	2 & 4	Ada 83
sc	Sensor Control Firmware	3	Ada 83
mc	Missile Control Card	3	Ada 83
fa	Avionics - Engine Control	3	Ada 83 *
sp	Avionics - Fuel System	3	Ada 83 *

Table 1

Application Size

One of the problems of measuring code size in Ada is determining exactly what is a line of Ada code. Consider the following code.

```
declare
 days_in_month: integer;
begin
 case month is
  when apr | jun | sep | nov =>
    days_in_month := 30;
  when feb =>
   if year mod 4 = 0 then
    days_in_month := 29;
   else
    days_in_month := 28;
   end if;
  when others =>
   days_in_month := 31;
 end case;
end;
```

How many lines of code is this? There are 16 rows (carriage returns) and 8 semicolons, twice as many non-comment Ada source lines as semicolons. For the purposes of this survey lines of code are specified as the number of semi-colons. It would be interesting to know how many equivalent C source lines would be counted to perform the equivalent calculation.

The code size of the various applications varied considerably from the smallest of 13,000 LoC to the largest which was a massive 353,000 LoC.

Table 2 shows the spread.

2.3 Development Phasing - Questions A3 to A5

Predictably, all the developments were monitored using a waterfall overlapping model (sometimes known as the sashimi model). In all cases the complexity of the application typically led to evolving requirements which cascaded down the model forcing iterations between design and coding, coding and testing. Because of the inevitable changing war scenarios, the Ministry of Defence accept that the hardware and software is bound to change causing "requirements creep". The civil engine software builders suffer the same problem for different reasons and report that the engine is being designed as the software is built, inevitably leading to a cyclic process of redesigning, re-coding and re-testing, a cycle that is repeated 20 times in this instance.

Project Identity	Application Type	Ada LoC
tr	Armoured Vehicle Fire Control	13,000
mc	Missile Control Card	13,700
sc	Sensor Control Firmware	22,100
sp	Avionics - Fuel System	26,893
ps	Missile Guidance	35,900
fa	Avionics - Engine Control	45,000
pm	Missile Guidance	55,000
sm	Avionics - Stores management	70,500
es	Avionics - Stores Management	76,500
ls	Airborne Early Warning	101,000
ad	Air Defence Command and Information	219,000
cm	Electronic Surveillance and Counter Measures	353,000

Table 2

There is a marked inconsistency between the companies when it came to quoting percentage resources used during each stage of the development process. In one case requirements were captured and analysed elsewhere and in another, independent testing was carried out outside the department. Some companies specified their chargeable management time as a separate entity while others included the management time within the phase of the process being managed. Two companies were sharing the work with another site so that there were increased overheads in management time.

It was difficult to draw conclusions from this input. However, an attempt was made to take eight of the projects and normalise the data to arrive at approximate percentage man-hours for the four basic development stage (see Table 3).

Project Identity	Ada LoC	Requirements	Design	Code/Unit Test	Integration/Acceptance
tr	13,000	20%	40%	25%	15%
sp	26,893	0%	22%	28%	50%
fa	45,000	18%	18%	32%	32%
pm	55,000	12%	8%	40%	40%
sm	70,500	13%	14%	20%	53%
ls	101,000	6%	23%	36%	24%
ad	219,000	10%	42%	25%	23%
cm	353,000	14%	22%	18%	46%
Average		12%	26%	25%	37%

Table 3

Project Identity	Ada LoC	Staff Costs (£ m)	H/W Costs (£ m)	S/W Costs (£ m)	Other Costs (£ m)	Total Costs (£ m)
tr	13,000	0.700	0.039	0.050	0.020	0.809
mc	13,700	0.528	0.009	0.038		0.575
sc	22,100	0.792	0.650	0.151		1.008
sp	26,893	3.752	0.150	0.100		4.002
ps	35,900	3.000	0.225	0.105		3.330
fa	45,000	1.600	0.151	0.100		1.851
pm	55,000	2.750	0.600	0.100		3.450
sm	70,500	3.500	0.135	0.348		3.983
es	76,500	2.800	0.095	0.314		3.209
ls	101,000	4.650	0.500	0.300	0.034	5.840
ad	219,000	7.400	0.290	0.200		7.890
cm	353,000	35.000	0.800	0.800		36.600

Table 4

There is very little that can be drawn from this data other than that coding/testing uses between 18% and 40% of the man hours of a project.

2.4 Project Costs - Questions B1 to B4

Table 4 shows the costs of each project broken down into staff, hardware, software and other costs. The hardware and software costs vary considerably. Some companies do not cost in the basic network servers because they are included in the company IT budget and not associated with a particular project. Some companies use hardware rigs and software packages supplied free by the client.

Note the low percentage costs of the hardware and software, somewhere at 20% of total cost on some of the smaller projects and down to as low as 5% on the larger. There are hardware (rigs) and software (test code) that are specific to projects but a large proportion of these costs could be amortised over future projects and are therefore, in reality, even lower.

Project Identity	S/W Costs (£ m)	Total Costs (£ m)	% Tools Costs
mc	0.038	0.575	6.61%
tr	0.050	0.809	6.18%
sc	0.151	1.008	14.98%
fa	0.100	1.851	5.40%
es	0.314	3.209	9.78%
ps	0.105	3.330	3.15%
pm	0.100	3.450	2.90%
sm	0.348	3.983	8.74%
sp	0.100	4.002	2.50%
ls	0.300	5.840	5.47%
ad	0.200	7.890	2.53%
cm	0.800	36.600	2.19%
Total	2.606	72.547	3.59%

Table 5

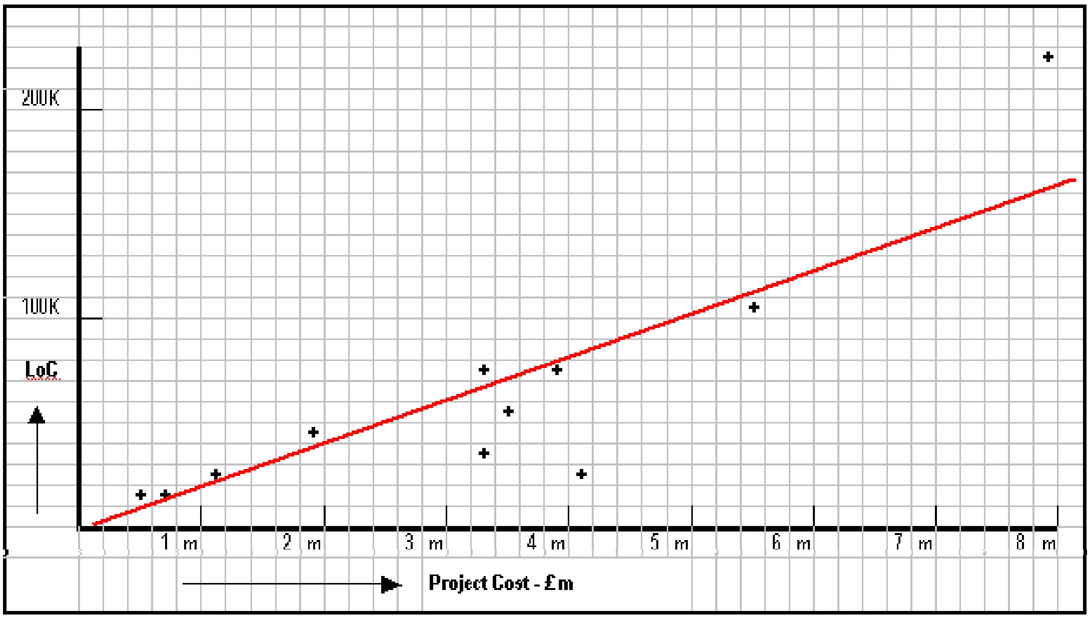

Table 6

The cost of "tooling up" a project is comparatively small. Just to emphasise this point, the Table 5 is supplied. Sorted into ascending size of project the table shows that among the smaller projects a percentage of around 6% is common, although there are exceptions including one project spending almost 15% on tools, and among the largest projects this percentage drops to around 3% and less.

If the percentage costs of software tooling is so low why then the concern by the industry of these costs? The reason may lie in the way defence projects are funded. The Ministry of Defence, for good reasons, tends to fund man-hours but not tools, therefore, as a general rule, the cost of tools is set against the company profits not the total costs.

Returning to the cost of Ada LoC, the data in Table 4 can be plotted, Total Cost of Project vs Lines of Code. Leaving aside the large project which will be off the graph, the remaining projects are represented graphically in Table 6. The red line is the "least squares fit" and shows, for what it's worth, that on average, 1000 lines of Ada code cost £48,000 throughout the entire life of the project.

2.5 Lines of Code per Day (LoC) Question B5

The number of Ada lines of code that could be developed per engineer day over the whole life of the project was also specified at each interview and the results are listed in Table 7.

Project Identity	Application Type	Ada LoC	SIL	LoC per day
sp	Avionics - Fuel System	26,893	3	2.00
ps	Missile Guidance	35,900	0	3.00
pm	Missile Guidance	55,000	0/3	3.40
cm	Electronic Surveillance and Counter Measures	353,000	0	4.00
sm	Avionics - Stores Management	70,500	2/3	4.10
tr	Armoured Vehicle Fire Control	13,000	2	6.00
ls	Airborne Early Warning	101,000	2	6.56
mc	Missile Control Card	13,700	3	7.78
fa	Avionics - Engine Control	45,000	3	8.00
es	Avionics - Stores Management	76,500	2/4	8.00
sc	Sensor Control Firmware	22,100	3	8.37
ad	Air Defence Command and Information	219,000	0	9.60

Table 7

These statistics speak for themselves. Firstly, over the life of a project, they show how few Ada instructions are actually engineered per day. Secondly, they show what a huge discrepancy exists between the best and the worst performing projects, five times, in fact. The Ministry of Defence independently arrived at the same spread and quoted that LoC per engineer day was 2 to 9.

Development Performance related to Project Size

On the evidence, this large spread is clearly unrelated to project size, which is a little surprising. One would expect large amounts of code to incur a great deal of overhead cost due to integration problems. One is tempted to infer that maybe Ada keeps integration costs down, although only three companies specifically mentioned the benefit of Ada in this respect.

Development Performance related to SIL

Nor is there any evidence, from this sample at least, that critical systems (SIL levels > 0) are more labour intensive than non-critical systems (SIL levels = 0). In fact, by removing the first and last entries in Table 7, where the project performance can be explained in other ways, it appears that the higher the SIL level the more code per man-day is achieved. Why this should be so is not easy to explain from the data. It could be conjectured that critical systems can only be developed by the highest quality software engineers who are indeed the best performing coders.

Using all the entries in Table 7 gives the following average performance by SIL:

SIL 0	4.76 LoC per day
SIL 2	5.31 LoC per day
SIL 3	4.58 LoC per day

Development Performance related to Experience

It is difficult to glean from these statistics why exactly one company can only write 2 LoC per day and another 9 LoC per day. Looking at the lowest performing projects there are two things that may be significant. In one case, the company had great difficulty employing graduate programmers and had to resort to training numerate non-programmers in Ada. In the other case, the company was obliged to share the application development with their French counterparts who were using a different development regime (e.g. C + different testing tools).

A closer examination of the importance of experienced Ada programmers reveals that four companies had evidence that their code performance improved with the experience of their programmers. In particular, one company reported an incremental improvement in performance starting with 4 LoC per day on the first project, rising to 6.6 LoC on the next and 9 on the next. Another reported that initially they were only achieving 2 LoC but this rose to 9 LoC in three years. One company went so far as to say they could eventually achieve 20 LoC per day.

Development Performance relating to Management

Three of the lowest four performing projects were developments split over two sites. No judgement is being made on the wisdom of such decisions; there are good managerial and political reasons for sharing sites. However, the increased burden on management, which is an expensive resource, inevitably brings down overall development performance. On the evidence of one company, sharing developments operating to different development regimes would appear only to exacerbate the problem.

2.6 The Costliest Lifecycle Stages – Questions B6 and B7

The projects that specified their most expensive stages fell into basically two equal camps. In one were those that experienced volatile requirements and therefore lots of design changes. There were four in this category. The other camp was those that rated elements subsequently to the design phase, the coding/testing/integration group. Again there were four projects in this category.

Having identified the most expensive stages the projects were asked to specify where costs were unacceptably high. Two companies recognised that requirements inevitably changed but they were concerned that there was insufficient control and management of these changes. Three companies concentrated more on the testing stages, but their problems were very different. One needed an additional hardware rig, one lacked experienced safety critical test engineers and one found the safety critical SPARK analyses just unacceptable expensive. Finally one company found the integration phase too expensive due to the shortage of one hardware rig. These results are given in Table 8.

2.7 Choice of Language - Question 8

Finally the companies were asked to speculate whether, with the benefit of hindsight, they would have chosen a different language to develop the application.

With possibly one dissenting opinion, the unanimous answer was that they would not have changed the language. One project was not complete but this company was critical of C so it is unlikely that they would change in the future. The dissenting opinion speculated that using C might have reduced costs but insignificantly. Two companies were reviewing their language choice for future projects.

3. Conclusion

In answer to the key question posed in the introduction of this study, that is does Ada make a significant contribution to the efficiency of the software development process, there is no evidence that it does.

The cost of designing, coding, testing, integrating and delivering 1000 lines of Ada code varies from £36,000 to £150,000. Developing safe real time embedded systems is one expensive exercise (!) and Ada is regarded as the only candidate language.

On the other hand, there is no evidence either that the only viable alternative language, which is C, would do a better job of improving the efficiency of development. This opinion is only based on rhetorical evidence since this limited study did not encounter equivalent C based metrics. No company thought there would be a significant reduction in development costs using C instead of Ada.

Project Identity	Application Type	Costliest Stage	Unacceptably Expensive Stages
sp	Avionics - Fuel System	Reqs./Design	Requirements
ps	Missile Guidance	Unspecified	
pm	Missile Guidance	Unspecified	
cm	Electronic Surveillance and Counter Measures	Requirements	Requirements
sm	Avionics - Stores Management	Unspecified	
tr	Armoured Vehicle Fire Control	Testing	Testing
ls	Airborne Early Warning	Testing	SIL 3 Testing
mc	Missile Control Card	Code Review	SPARK Analyses
fa	Avionics - Engine Control	Unspecified	
es	Avionics - Stores Management	Detailed Design	None
sc	Sensor Control Firmware	Requirements	Maintenance/ Enhancement
ad	Air Defence Command and Information	Integration	Integration and Final Testing

Table 8

Areas considered unacceptably expensive were:

a) *Control of change due to "requirements creep"*
There are tools to do this although it is outside of the scope of this study to comment on their efficacy. Ada's contribution can only be marginal.

b) *Real Time testing.*
A number of companies said that the longer they put off the coding/testing stage the cheaper were the overall project costs. Design reviews, Fagan walk-throughs, more emphasis on hardware availability, in other words good management, were cited as ways to reduce testing expense. Ada was said to support delayed testing and eliminate the necessity for unit testing, C was criticised for encouraging the opposite.

c) *Integration*
Developing large systems across different sites to different standards is expensive and the development language makes little difference.

d) *Maintenance and Enhancement.*
Some anecdotal evidence in this study that maintenance and enhancement is cheaper with Ada but no metrics to prove this contention were provided. The trouble is, a system is almost never programmed twice in different languages.

The cost of software tools was not mentioned as unacceptable. On the contrary, the statistics tend to show that it is a relatively low percentage of total project costs.

Generally speaking, the choice of Ada and the ongoing preference for the language has less to do with improving project performance but more to do with its facility to support large, real time safety critical systems. The evidence that there is, indicates a general belief that C is not up to the job and anyway, won't improve development performance to any significant extent.

In the end, it would appear, that development performance very much comes down to the quality and experience of the programmers.

Appendix A: Completed Questionnaires

The full text of Appendix A is available on the Ada UK website:
http://www.adauk.org.uk
Click on "publications" in the left hand menu, and then look under surveys in the main window.

Appendix B: Ada vs C Comments

C, C++ don't have strong typing, many memory leaks. Easier to code but integration and maintenance harder. C++ leads to earlier generation of code. Cheaper to use Ada. C++ needs code analyser. More difficult to recruit Ada programmers. Programmers prefer "wizzy" languages.

Ada eliminates 90% of low level coding. Easier to build stub hammocks in C and easier to test. Ada slow to compile.

SPARK Ada is the de facto standard for high integrity systems difficult to audit C code. Used C on avionics display systems.

Walk-throughs can replace unit testing. Ada programmers improve with each project.

Portability Host to Target a key advantage of Ada. Ada is well defined once compiled tends to work.

Test Rigs far outweigh costs of compilers. C ok for small projects but only Ada for large projects. Ada easier for code review. Ada expensive but not too expensive. Benefits of Ada compilation outweigh C.

Ada not cause of poor productivity but next project will be in visual C++.

Entirely an Ada shop because of safety criticality. Not impressed with C standards, inefficient and incorrect. The French C legacy code should have been rewritten in Ada.

Supporting 19 targets could not have been done without Ada.

It's a fallacy that Ada is too expensive although compilers are expensive. Benefits of Ada compilation outweigh C. C ok for small projects but only Ada for large ones. Ada much easier for code review.

Do You Use Generic Arrays and Vectors? – ISO/IEC 13813: 1998

John Barnes

11 Albert Road, Caversham, Reading RG4 7AN.; Tel: +44 118 947 4125

Abstract

This short note explains the background to the standard for complex types, arrays and vectors which is based on Ada 83 but also contains information regarding how it can be used with Ada 95. This standard will soon be due for review and this note invites comment on how it should be revised.

Introduction

The Numerics Working Group of WG9 met many times during the period when Ada 95 was being designed and produced a number of standards. They were faced with the problem of whether to produce standards based on Ada 83 (87 in ISO terms) or whether to base them on Ada 95 or subsume them into Ada 95. One dilemma was of course that although Ada 95 was on the way nevertheless use of Ada 83 was expected to continue in use for many years.

The standards are

11430: Generic package of elementary functions for Ada.

11729: Generic package of primitive functions for Ada.

13813: Generic packages of real and complex type declarations and basic operations for Ada (including vector and matrix types).

13814: Generic package of complex elementary functions for Ada.

I mention 11430 and 11729 for completeness. They were published in 1994. They were based entirely on Ada 83 and their facilities are provided in the Ada 95 core language. The elementary functions, 11430, became the package Ada.Numerics.Generic_Elementary_Functions and the primitive functions, 11729, became the various attributes such as 'Floor and 'Ceiling, and 'Exponent and 'Fraction. These two standards were withdrawn recently and I will mention them no more.

Whither 13813 and 13814?

The other two standards, 13813 and 13814, were published in 1998 and will soon be up for review at the end of their five year period. Three possible fates can befall a standard when it is reviewed. It can be withdrawn, revised or confirmed.

These standards were discussed at a BSI meeting in April and at the recent WG9 meeting held in Leuven in May.

In the case of 13814, the functionality is all incorporated into the Numerics Annex of Ada 95 as the package Ada.Numerics.Generic_Complex_Elementary_Functions. There are a few changes in presentation because the Ada 95 package uses the generic package parameter feature which of course did not exist in Ada 83. Nevertheless there seems little point in continuing with 13814 and so at the Leuven meeting it was agreed to recommend that it be withdrawn.

However, the situation regarding 13813 is not so clear. Some of its functionality is included in Ada 95 but quite a lot is not. The topics covered are (1) a complex types package including various complex arithmetic operations, (2) a real arrays package covering both vectors and matrices, (3) a complex arrays package covering both vectors and matrices, (4) a complex input-output package.

The complex types package (1) became the package Ada.Numerics.Generic_Complex_Types and the input-output package (4) became Ada.Text_IO.Complex_IO. However, the array packages, both real and complex, were not incorporated into the Ada 95 standard.

At the Leuven meeting, it was agreed that 13813 should not be withdrawn without further study. Moreover, I spoke too much and consequently was given homework in the sense that the UK (the BSI) was asked to study whether small or large changes are required in 13813 and to report back at the next meeting (Minneapolis in October 2001). The Ada Rapporteur Group would then decide whether the functionality should be included in a future revision or amendment to Ada 95.

Any advice on this from the user community would be very welcome. In particular we would like to hear from anyone who has used this standard. It seems that no vendor has implemented it but it's pretty easy to implement and maybe users have implemented it or something similar for themselves.

If you do not have a copy of the standard but wish to comment on it in order to allow me, as the UK Head of Delegation to WG9, to make comments on your behalf then please contact me on jgpb@jbinfo.demon.co.uk.

References

ISO/IEC 13813: 1998: Information technology - programming languages - Generic packages of real and complex type declarations and basic operations for Ada (including vector and matrix types).

Book Review: *"Real-Time Systems and Programming Languages" by Alan Burns and Andy Wellings*

Pat Rogers

Software Arts and Sciences; Tel.: +1-281-648-3165; Email: progers@classwide.com URL: www.classwide.com

Real-Time Systems and Programming Languages by Alan Burns and Andy Wellings promises to "present a comprehensive description and assessment of the programming languages and operating systems used" in real-time/embedded systems. Those two subjects are large parts of a very large domain but the correspondingly substantial book fulfills its promise and does so in a manner suitable for professionals as well as advanced students. Considerable depth is achieved in each topic but the text is nonetheless highly readable.

In this third edition, the Java language joins Ada 95, C/POSIX, and occam2 as the principle programming languages assessed. Java replaces Modula-2 from earlier editions because of the appearance of the real-time Java specifications and the resulting expectation for increased use in real-time applications. The real-time specifications for Java are thus covered in detail. Other languages are also discussed when they provide facilities not available in the core languages.

Instead of comparing the languages directly to each other, though, each is examined in terms of the primitives they provide for development of reliable, deadline-driven software. As such, a reader interested in any one of the core languages will find ample material available. However, a principle benefit of the book is the insight made possible by examining several languages in light of common requirements, highlighting relative strengths and weaknesses in each.

The book starts with an introduction to real-time systems and includes overviews of each of the core languages, such that the material can stand on its own and be read in fragments as the interests of the reader may dictate. The extensive material on meeting deadlines via schedulability analysis is perhaps the most valuable individual part of the book – coming as it does from acknowledged leaders in the field – but to read only those parts would be to miss the all-inclusive nature of the material.

The book provides detailed coverage of the kind of low-level programming ubiquitous in embedded systems. There is considerable treatment of reliability and fault tolerance, requirements often left unstated but essential in many real-time systems. Each of these topics is discussed in terms of language support, as are the POSIX real-time extensions. Distributed real-time systems are included because they are becoming more common, and indeed, the real-time CORBA facilities are explored.

Much more than a "scheduling book", the text offers an extensive, detailed, and thorough exploration of the programming language issues involved in the production of reliable real-time systems. Although scheduling is certainly covered completely, other critical aspects, especially concurrency, reliability, and fault tolerance, are also explored extensively and in detail.

We in the Ada community generally believe Ada to be the best language for this domain, perhaps at the expense of learning what other languages have to offer. *Real-Time Systems and Programming Languages* provides a highly-recommended resource for learning about other languages and techniques, but it is quite likely that the Ada-oriented reader will learn something new about their favorite language as well. Readers familiar with the authors' other books will recognize that as characteristic of all their work.

Publication Details:

Real-Time Systems and Programming Languages (Third Edition)
Alan Burns and Andy Wellings
Hardback – 611 pages
March 2001
Addison Wesley Longman
ISBN: 0201729881

Ada-Europe Associate Members (National Ada Organizations)

Ada-Belgium

attn. Dirk Craeynest
c/o Offis nv/sa
Weiveldlaan 41/B32
B-1930 Zaventem
Belgium

Phone:	+32-2-725-40-25
Fax:	+32-2-725-40-12
Email:	Dirk.Craeynest@offis.be
URL:	www.cs.kuleuven.ac.be/~dirk/ada-belgium

Ada in Denmark

attn. Jorgen Bundgaard
c/o DDC-I
Gl. Lundtoftevej 1B
DK-2800 Lyngby
Denmark

Phone:	+45-45-871144
Fax:	+45-45-872217
Email:	jb@ddci.dk

Ada-Deutschland

attn. Dr. Peter Dencker
Aonix GmbH
Durlacher Allee 95
D-76137 Karlsruhe
Deutschland

Phone:	+49-721-98653-22
Fax:	+49-721-98653-98
Email:	dencker@aonix.de
URL:	ada-deutschland.de

Ada-France

chez Fabrice Kordon
48 rue Vergniaud
F-75013 Paris
France

Phone:	+33-1-44 27 61 89
Fax:	+33-1-44 27 62 86
Email:	bureau@ada.eu.org
URL:	www.ada-france.org

Ada-Spain

attn. Francisco Perez-Zarza
P.O. Box 50.403
E-28080 Madrid
Spain

Phone:	+34-1-627-8247
Fax:	+34-1-309-3685
Email:	fperez@ceselsa.es
URL:	www.adaspain.org

Ada in Sweden

Ada I Sverige
c/o Mariadata
Box 1085
SE-141 22 Huddinge
Sweden

Phone:	+46-08-779-88-30
Fax:	+46-08-774-37-93
Email:	info@ada-i-sverige.se
URL:	www.ada-i-sverige.se

Ada –Switzerland

attn. Henri Roethlisberger
EINEV, Ecole d’Ingenieurs de l’Etat de Vaud
CP 587
CH-1401 Yverdon-les-Bains
Switzerland

Phone:	+41-24-423-22-63
Fax:	+41-24-425-00-50
Email:	roethlisberger@einev.ch

Ada Language UK

attn. Helen Byard
P.O. Box 322
York YO10 3GY
UK

Phone:	+44-(0)1904-412740
Fax:	+44-(0)1904-426702
Email:	admin@adauk.org.uk
URL:	www.adauk.org.uk

Ada UK 2001 Sponsors

ACT Europe
Contact: Franco Gasperoni
8, Rue de Milan, 75009, Paris, France
Tel: +33-1-49-70-67-16 Fax: +33-1-49-70-05-52
Email: sales@act-europe.fr URL: www.act-europe.fr

Alenia Marconi Systems
Contact: Don Harvey
Eastwood House, Glebe Rd., Chelmsford, Essex, CM1 1QW, UK
Tel: +44-(0)1276-696901 Fax: +44-(0)1491-571866
Email: info@aonix.co.uk URL: www.aonix.co.uk

Aonix Europe Ltd
Contact: Neil Michniak
Partridge House, Newtown Rd., Henley on Thames, Oxon, RG9 1HG, UK
Tel: +44-(0)14941-415000 Fax: +44-(0)14941-571866
Email info@aonix.co.uk URL: www.aonix.com

ARTiSAN Software Tools
Contact: Peter Kibble
Stamford House, Regent St., Cheltenham, Glos., GL50 1HN, UK
Tel: +44-(0)1242-229320 Fax: +44-(0)1242-229301
Email: peterk@artisansw.com URL: www.artisansw.com

BAE SYSTEMS
Contact: Paul McCormack
Warwick House, PO Box 87, Farnborough Aerospace Centre, Farnborough, Hants, GU14 6YU, UK
Email: Paul.McCormack@baesystems.com URL: www.baesystems.com

Data Systems and Solutions
Contact: Dave Woodhall
SEAS Building, Sinfin Lane, Derby, DE24 8BJ, UK
Tel: +44-(0)1332-771700 Fax: +44-(0)1332-770921
Email: info@ds-s.com URL: www.ds-s.com

EDS
Contact: Lee Edwards
Hartley House, 15 Bartley Wood Business Park, Bartley Way, Hook, Hants., RG27 9XA, UK
Tel: +44-(0)1256-741122 Fax: +44-(0)1256-741132
Email: swep.sales@eds.com

First Matrix Ltd
Contact: Alan Barker
Old Lion Court, High St,. Marlborough, Wilts., SN8 1HQ., UK
Tel: +44-(0)1672-515510 Fax: +44-(0)1672-515514
Email: arb@ftmx.com

Green Hills Software Ltd
Contact: Jon Williams
Chancery Court, Lincoln Rd., High Wycombe, Bucks., HP12 3RE., UK
Tel: +44-(0)1494-429336 Fax: +44-(0)1494-429339
Email: sales-uk@ghs.com URL: www.ghs.com

IPL Information Processing Ltd
Contact: Ian Gilchrist
Eveleigh House, Grove St., Bath, BA1 5R., UK
Tel: +44-(0)1225-475114 Fax: +44-(0)1225-444400
Email: ipl@iplbath.com URL: www.iplbath.com

LDRA Ltd
Contact: Jim Kelly
24 Newtown Rd., Newbury, Berks., RG14 7BN, UK
Tel: +44-(0)635-528828 Fax: +44-(0)635-528657
Email: sales@ldra.com URL: www.ldra.com

Objektum
Contact: Derek Russell or Ahmed Amin
Units 2/3 Cranleigh Works, The Common, Cranleigh, GU6 8SB, UK
Tel: +44-(0)1483-278178 Fax: +44-(0)1483-275384
Email: info@objektum.com URL: www.objektum.com

Praxis Critial Systems Ltd
Contact: Peter Amey
20 Manvers St., Bath, BA1 1PX, UK
Tel: +44-(0)1225-469991 Fax: +44-(0)1225-466006
Email: sparkinfo@praxis-cs.co.uk URL: www.praxis-cs.co.uk

Rational Software Ltd
Contact: Roger Bowser
Kingswood, Kings Ride, Ascot, Berks., SL5 8AJ, UK
Tel: +44-(0)1344-295000 Fax: +44-(0)1344-295001
Email: info@rational.com URL: www.rational.com

John Robinson & Associates
Contact: John Robinson
2 Currer St., Oakenshaw, Bradford, W. Yorks., BD12 7DP, UK
Tel: +44-(0)1274-691935 Fax: +44-(0)8700-558750
Email: John@jr-and-assoc.demon.co.uk URL: www.jr-and-assoc.demon.co.uk

Telelogic UK Ltd
Contact:
Chancery House, 8 Edward St., Birmingham, B1 2RX, UK
Tel: +44-(0)121-2346600 Fax: +44-(0)121-2346611
Email: info@telelogic.com URL: www.telelogic.com

TNI Europe Ltd
Contact: Tony Elliston
58a Mill St., Congleton, Cheshire, CW12 1AG, UK
Tel: +44-(0)1260-291449 Fax: +44-(0)1260-291449
Email: info@tni-europe.com URL: www.tni-europe.com

Wind River Systems UK Ltd
Contact: David Bew
Unit 5 & 6, 1st Floor, Ashted Lock Way, Aston Science Park, Birmingham, B7 4AZ, UK
Tel: +44-(0)121-3590999 Fax: +44-(0)121-3804444
Email: inquiries-uk@windriver.com URL: www.windriver.com