

# ADA USER JOURNAL

Volume 28

Number 2

June 2007

---

## Contents

	<i>Page</i>
Editorial Policy for <i>Ada User Journal</i>	66
Editorial	67
News	69
Conference Calendar	102
Forthcoming Events	111
Articles	
J A de la Puente “13 <sup>th</sup> International Real-Time Ada Workshop (IRTAW-13)”	117
W Glaister “Template Driven Design in UML and Ada 95”	122
Ada-Europe 2007 Sponsors	128
Ada-Europe Associate Members (National Ada Organizations)	Inside Back Cover

# Editorial Policy for Ada User Journal

## Publication

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

## Aims

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

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

The Journal publishes the following types of material:

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

Further details on our approach to these are given below.

## Original Papers

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

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

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

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

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

## News and Product Announcements

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

## Reprinted Articles

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

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

## Commentaries

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

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

## Announcements and Reports

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

## Reviews

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

## Submission Guidelines

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

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

# Editorial

If you would allow me a grand, legendary quotation, I should like to title this editorial: “*The times they are a-changin*” (Bob Dylan, 1964). Yes indeed they are, though in less prophetic ways in our little world of the Ada-Europe organization than in that historical song. We have held the production of this June issue up past the week of the Ada-Europe conference because a number of changes were in preparation that involved the attention of several of us. Now that the 12<sup>th</sup> International Conference on Reliable Software Technologies – Ada Europe 2007 has been celebrated, and with considerable success, we ought to say, thanks to the dedicated effort of the organizing team at the University of Applied Sciences in Geneva, Western Switzerland, we can just begin to see the extent of the changes. We will talk about them at length in the next issue of the journal. For now, suffice it to say that the Ada User Journal will have a new editor-in-chief in the person of Luís Miguel Pinho of Portugal, to whom I address my best wishes and encouragement for a taxing but also rewarding job. This is therefore the last editorial I am signing after 5 years of service, which amount to 5 volumes and 20 issues in journal terms: almost a lifetime by the sound of it! I had the luck of being editor through the beautiful time of the Ada language amendment process, which fed the journal with lot of value to process and to read. That much I will remember as a treat to me and, hopefully, to the readership as well. We (meaning the editorial team and I) have tried hard to keep the journal production timely, which we have managed to achieve but only at times, for which I must apologize to the readers. When I took over the editorship of the Ada User Journal the backlog was thin and we had to go ways to stimulate the production of articles of value. In handing over the baton to Miguel, I am happy to leave behind a robust backlog of worthwhile material. I look forward to enjoying reading the articles in store (and the new ones that will surely come along) in future journal issues. Let me now come to illustrate the contents of this issue. You may remember that in the previous editorial I talked about IRTAW-13 (the 13<sup>th</sup> edition of the International Real-Time Ada Workshop, which took place last April in the mountains of Vermont). Well, I am happy that the workshop chair, Juan Antonio de la Puente, kept his promise to deliver his workshop report in time for publication in this issue. Since I had the pleasure to attend that event I can safely recommend you to read that report and have some taste of the insightful discussions that it hosted. I am also happy to inform you that we are taking steps in these days to obtain permission to publish the proceedings of IRTAW-13 in a future issue of the Ada User Journal. With the second technical article in this issue we also commence the staggered publication of the proceedings of the Industrial Track of the Ada-Europe 2007 conference. The first article to appear is by Wain Glaister from AgustaWestland, UK, which talks about UML design patterns, templates, and foundation code frameworks in Ada. Let me conclude this editorial by thanking the News editor, Santiago Urueña, and the Calendar editor, Dirk Craeynest, for the continued support (often as large as half issues) that they provided to me and to the journal in the years of my tenure. Thanks again lads. *Proficiat* to the new editor-in-chief and long live the Ada User Journal.

*Tullio Vardanega*

*Padova*

*June 2007*

*Email: tullio.vardanega@math.unipd.it*

# News

**Santiago Uruña**

*Technical University of Madrid (UPM). Email: Santiago.Uruena@upm.es*

---

## Contents

Ada-related Organizations	69
Ada-related Events	70
Ada-related Resources	73
Ada-related Tools	73
Ada-related Products	77
Ada and GNU/Linux	85
Ada and Macintosh	87
References to Publications	87
Ada Inside	87
Ada in Context	90

---

## Ada-related Organizations

### ARA — Ada 2005 Becomes Official ISO Standard

*From: AdaIC*

*Date: March 9, 2007*

*Subject: Ada 2005 Becomes Official ISO Standard*

*URL: <http://www.adaic.org/news/Ada05-final.html>*

Oakton, VA (March 9, 2007) — The Ada Resource Association today announced the formal completion of the process to revise the Ada 95 language, with the publication of the Ada 2005 standard by ISO (the International Organization for Standardization) in Geneva, Switzerland. Publication of the Ada 2005 standard — officially named ISO/IEC

8652:1995/Amd 1:2007 — culminates a collaborative international effort under ISO's Ada Working Group (WG9) to enhance the 1995 version of the Ada language. The effort was sponsored in part by the Ada Resource Association, which helped support the work of the project editor, Mr. Randall Brukardt.

Ada 2005 offers significant enhancements in several areas. Improvements in the language's Object-Oriented Programming features include the addition of Java-like interfaces and traditional "object.operation" syntax. More flexible program structuring allows mutually dependent package specifications and makes it easier to interface with languages such as Java. Real-time system support includes additional task dispatching policies, execution-time clocks, and handlers for task termination. The concurrency and object-oriented features are successfully unified through a new interface feature that allows

implementation through either a sequential or concurrent type. Support for safety and security is enhanced with the inclusion of the Ravenscar Profile (a tasking subset that is amenable to safety certification), syntax that avoids some common Object-Oriented Programming errors with inheritance, and a mechanism for defining language profiles. Other enhancements increase the language's general expressiveness, for example by allowing nested subprograms to be passed as run-time parameters, and by extending the predefined environment with new functionality such as a Containers library.

The new Ada standard advances the state of the art in language design and is expected to strengthen Ada's role as a language of choice for systems with stringent safety and/or security requirements. Copies of the new Ada Reference Manual and the supporting Rationale are available on the Ada Information Clearinghouse website at <http://www.adaic.org/standards/ada05.html>.

About the Ada Resource Association

The Ada Resource Association (ARA), an international Ada advocacy trade group, is committed to ensuring the continued success and expanded usage of Ada-related technology. The ARA financially supports the maintenance of the Ada language standard and maintains the Ada Information Clearinghouse website. Current ARA members are AdaCore, IBM Rational Software, Praxis High Integrity Systems, and SofCheck.

### ARA — Development snapshot for ACATS 3.0

*From: AdaIC*

*Date: March 14, 2007*

*Subject: Update — Ada Conformity Assessment Test Suite*

*URL: <http://www.adaic.com/whatsnew.html>*

The last Ada 95 Ada Conformity Assessment Test Suite, ACATS 2.6, has been posted, along with an associated ACATS Modification List 2.6A and its associated test files.

[See also "Updated Ada Conformity Assessment Test Suite" in AUJ 23-2 (Jun 2002), p.70. —su]

*From: AdaIC*

*Date: March 30, 2007*

*Subject: Update — Ada Conformity Assessment Test Suite*

*URL: <http://www.adaic.com/whatsnew.html>*

ACATS Modification List 2.6B and the associated test files have been posted.

Also, the first development snapshot for ACATS 3.0 was posted.

<http://www.adaic.com/compilers/testing.html>

### Ada-Belgium — Upcoming SC22/WG9 meeting

*From: Dirk Craeynest*

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

*To: [ada-belgium-info@cs.kuleuven.be](mailto:ada-belgium-info@cs.kuleuven.be)*

*Date: 03-may-2007 22:21*

*Subject: Upcoming SC22/WG9 meeting and review of Ada Issues*

As you may know, there is an upcoming meeting of ISO's Ada language working group (ISO/IEC JTC1/SC22/WG9) scheduled after the Ada-Europe 2007 conference end of June in Geneva, Switzerland.

The Chairman of the Ada Rapporteur Group (ARG) of WG9 informed the Heads of Delegation that a number of Ada Issues (AIs) have entered Editorial Review, and are intended to be submitted to WG9 for approval at the above mentioned meeting.

The AIs can be found online at <http://www.ada-auth.org/AI-SUMMARY.HTML> and <http://www.ada-auth.org/AI05-SUMMARY.HTML>.

Ada 95 AI:

AI95-00447-01/02 2006-06-16 — Null\_exclusions allowed in Ada 95

Ada 2005 AIs:

AI05-0007-1/03 2006-12-13 — Stream 'Read and private scalar types

AI05-0013-1/03 2006-12-13 — No\_Nested\_Finalization is difficult to enforce

AI05-0014-1/02 2006-06-20 — Accessibility of designated objects

AI05-0015-1/03 2006-12-13 — Constant return objects

AI05-0016-1/02 2006-12-13 — Others => <> can be used in place of null record

AI05-0020-1/02 2006-12-13 — Universal operators of fixed point and access types

AI05-0021-1/02 2006-12-13 — Issues with containers

AI05-0025-1/02 2006-12-13 — Missing legality rules for formal\_package\_association

Those AIs are now being circulated within the Ada community for review, with the intention to return comments to the ARG in time to properly answer them before the WG9 meeting.

## Ada-related Events

[To give an idea about the many Ada-related events organized by local groups, some information is included here. If you are organizing such an event feel free to inform us as soon as possible. If you attended one please consider writing a small report for the Ada User Journal. —su]

### Mar 27 — Ada-Spain Technical Day Report

From: J. Javier Gutiérrez  
<gutierjj@unican.es>

To: Santiago Uruena  
<Santiago.Uruena@upm.es>

Subject: Meeting Report of the XV Technical Day of Ada-Spain

Date: Fri, June 15, 2007 11:43 am

The Technical Day of Ada-Spain is a yearly meeting devoted to presenting and discussing the results of research and development projects related to the Ada language and reliable software technologies.

This year, the meeting was held on March 27th at the Telecommunications Engineering School of the Technical University of Madrid (UPM), and it consisted of two invited talks and six technical presentations.

The invited talks were:

- *PolyORB: Leveraging Ada to build Distributed Real-Time and Embedded Systems*, by Jérôme Hugues, from the Department of Computer Science ENST (Paris), an expert in distributed systems, real-time systems and the usage of formal methods and modeling applied to complex systems. He has been involved in the PolyORB project since its beginning in 2002, becoming one of the head designers. Jérôme gave a very interesting talk that revealed the status of PolyORB and how it works. PolyORB currently supports basic RT-CORBA for statically scheduled systems. It was announced that in a short period of time support for Ada's Distributed Systems Annex will also be completely operative.

- *Indra-ATM: Air Traffic Control and Ada*, by Juan López Campos, from Altran SDB working for Indra-ATM, and expert in distributed systems and real-time systems. He has been elaborating the documentation of SACTA, an automatic system for air traffic control, and is currently working on the development of a new communications middleware for ATM environments (iMAS). Juan briefly introduced the main activities of INDRA-

ATM which is a very big company, and gave a specific view of the wide range of simulators they build. The audience was pleased to hear how an air traffic control system works, how the air space is divided in Europe, and which are the physical and logical mechanisms they use to make the control. Finally, he talked about the impact of the Ada language in the projects in which the company is involved.

The technical programme had the following presentations:

- “Including precise garbage collection in Ada”, by Francisco García Rodríguez, from the University of Las Palmas de Gran Canaria (ULPGC)
- “QoS in real-time systems with dynamic memory management”, by Alfons Crespo, from the Technical University of Valencia (UPV)
- “Partitioned real-time systems with Ada 2005”, by José A. Pulido, from the Technical University of Madrid (UPM)
- “Towards Ravenscar for distributed systems”, by Santiago Uruena, from the Technical University of Madrid (UPM)
- “Implementation of new Ada 2005 real-time services in MaRTE OS and GNAT”, by Mario Aldea, from the University of Cantabria
- “Integrating Ada and RT-CORBA into a real-time platform — a first approach”, by Héctor Pérez Tijero, from the University of Cantabria

Right after the finalization of the Technical Day, the General Assembly of Ada-Spain was celebrated. A few days after the assembly, the winners of the yearly Ada-Spain Award to the best academic project developed in Ada were announced. This year, two projects have been awarded with and shared the first prize: one was titled “Library for the creation of graphical interfaces for controlling embedded systems, and application to a telemanipulated arm”, authored by Ángel Muñoz Cantera, and the other one was “Device drivers for analog and digital input/output and for the I2C serial bus”, authored by Daniel Sangorrín López, both from the University of Cantabria.

### Jun 12 — Ada-Belgium

From: Dirk Craeynest

<Dirk.Craeynest@cs.kuleuven.be>

Subject: Ada at Barco Avionics, Tue 12 Jun 19:45, Ada-Belgium

Date: Tue, 29 May 2007 20:49:17

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

Newsgroups:

comp.lang.ada.fr.comp.lang.ada.be.com  
p.programming.nl.comp.programmeren

Ada – Belgium Special Evening Event  
A technical presentation by

Ludovic Brenta, Barco Avionics, Belgium  
*Ada at Barco Avionics:  
history, coding standards, and products*  
Tuesday, June 12, 2007, 19:45–21:45  
U.L.B., Department of Computer Science  
Boulevard du Triomphe / Triomflaan,  
B-1050 Brussels  
(after the Ada-Belgium 2007 General  
Assembly at 18:45)  
<<http://www.cs.kuleuven.ac.be/~dirk/ada-belgium/events/local.html>>

Announcement

Ada-Belgium will hold its 14th annual General Assembly on Tuesday, June 12, 2007, at the U.L.B., Department of Computer Science, Boulevard du Triomphe / Triomflaan, B-1050 Brussels, at 18:45. The official convocation is available, also in PDF format, and is sent via postal mail to all members.

At 19:45 the General Assembly will be followed by a technical presentation on “Ada at Barco Avionics: history, coding standards, and products”, by Ludovic Brenta from Barco Avionics, Belgium.

Abstract

The presentation will cover:

- an explanation of what Barco Avionics's products are; this can be in the form of flyers and slides, or Ludovic might even bring a couple of actual avionics units to show them;
- Barco Avionics's history with Ada;
- the Ada coding standards that Ludovic wrote in 2005 taking into account the unique constraints imposed by avionics certification requirements.

Speaker

Ludovic Brenta has been programming computers on various operating systems and languages since 1990. He graduated from INSA Lyon in 1996 with a diploma in Production Engineering. Having written too many bugs in other languages, he became interested in Ada in 2001 and started learning the language by himself. In 2003, he volunteered to maintain the Ada packages in Debian GNU/Linux, and became an official Debian Developer in 2006. Currently, his day job is at Barco Avionics, where he participates in the development of safety-critical embedded software, using Ada. Ludovic has been living in Brussels, Belgium, since 1998.

Participation

Everyone interested is welcome at either or both parts of this meeting. As usual, the event is free and presentations are in English.

If you plan to attend the General Assembly or the technical presentation, we would appreciate it if you could inform us by e-mail. Although no formal registration is required, this helps our preparations.

Looking forward to meet many of you in Brussels!

Dirk Craeynest  
President Ada-Belgium  
Dirk.Craeynest@cs.kuleuven.be

Acknowledgements

We would like to thank our sponsors for their continued support of our activities: AdaCore, Katholieke Universiteit Leuven (K.U.Leuven), Offis nv/sa — Aubay Group, and Université Libre de Bruxelles (U.L.B.).

## Jun 25–29 — Ada-Europe 2007

From: Dirk Craeynest

<Dirk.Craeynest@cs.kuleuven.be>

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

Date: Tue, 1 May 2007 08:09:29

Subject: 12th Int. Conf. on Reliable Software Technologies, Ada-Europe 2007

Summary: Advance Program is online. Registration has started.

Keywords: Conference, tutorials, reliable software, Ada, industry, LNCS, Geneva, ISO

Newsgroups:

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

Call for Participation

12th International Conference on Reliable Software Technologies — Ada-Europe 2007

25 – 29 June 2007, Geneva, Switzerland  
<http://www.ada-europe.org/conference2007.html>

Organized, on behalf of Ada-Europe, by Ecole d'Ingénieurs de Genève in cooperation with ACM SIGAda

Ada-Europe organises annual international conferences since the early 80's. This is the 12th event in the Reliable Software Technologies series, previous ones being held at Montreux, Switzerland ('96), London, UK ('97), Uppsala, Sweden ('98), Santander, Spain ('99), Potsdam, Germany ('00), Leuven, Belgium ('01), Vienna, Austria ('02), Toulouse, France ('03), Palma de Mallorca, Spain ('04), York, UK ('05), and Porto, Portugal ('06).

The 16-page Advance Program brochure with full information is available on the conference web site; both the AP and the web site contain the list of accepted papers and industrial presentations, as well as detailed descriptions of tutorials and keynote presentations. To download the AP, use the "Program" button on the conference home page and then the "Pdf Program" link.

Direct URLs

- Advance Program (large file, 535 Kb)
- conference registration form (small file, updated version)
- accommodation and travel information

Quick overview

- Mon 25 & Fri 29: tutorials
- Tue 26 – Thu 28: paper, industrial & vendor presentations, exhibition

Proceedings

- published by Springer-Verlag as volume 4498 in Lecture Notes in Computer Science series (LNCS); will be available at conference

Program co-chairs

- Nabil Abdennadher, University of Applied Sciences Geneva, Switzerland, [nabil.abdennadher@hesge.ch](mailto:nabil.abdennadher@hesge.ch)

- Fabrice Kordon, University Pierre & Marie Curie, France, [fabrice.kordon@lip6.fr](mailto:fabrice.kordon@lip6.fr)

Invited speakers

- Klaus D. Mueller-Glaser, University of Karlsruhe, Germany, "Challenges for Reliable Software Design in Automotive Electronic Control Units"

- Gerard Berry, Esterel Technologies, France, "Synchronous Techniques for Embedded Systems"

- Ali Mili, New Jersey Institute of Technology, USA, "Perspectives on Next Generation Software Engineering"

- Liviu Iftode, Rutgers University, USA, "Observation Rooms for Program Execution Monitoring"

Tutorials (full day)

- "Security by Construction", Rod Chapman, Praxis HIS, UK

- "Situational Method Engineering: Towards a Specific Method for each System Development Project", Jolita Ralyté, University of Geneva, Switzerland
- Tutorials (half day)

- "An Overview of Model Driven Engineering", William Bail, The MITRE Corporation, USA

- "Correctness by Construction: a UML2 Profile Enforcing the Ravenscar Computational Model", Tullio Vardanega, University of Padua, Italy

- "Object-Oriented Programming in Ada 2005", Matthew Heaney, On2 Technologies, USA

- "Verification and Validation for Reliable Software Systems", William Bail, The MITRE Corporation, USA

- "Synchronous Design of Embedded Systems: the Esterel/Scade approach", Gerard Berry, Esterel Technologies, France

- "Building Interoperable Distributed Applications with PolyORB", Thomas Quinot and Jérôme Hugues, AdaCore and ENST, France

Papers and Presentations

- 19 refereed technical papers on Ada 2005 and real-time systems, verification and validation, testing, MDA and applications, MDA and real-time,

Ada 2005 general purpose programming, distributed systems

- 9 industrial presentations, on current practice and challenges — authors from 11 countries: Australia, Austria, China, France, Germany, Italy, Republic of Korea, Spain, Tunisia, UK, and USA

Exhibition

- 6 exhibitors already committed: AdaCore, Aonix, Ellidiss Software (TNI-Europe), Green Hills Software, Telelogic, and Praxis; others expressed interest

- vendor presentation track for exhibitors

Social evening events

- Tuesday: visit of, and reception at, the building of the World Meteorological Organization (WMO, agency of the United Nations)

- Wednesday: aperitif in the History of Sciences Museum; banquet at the restaurant La Perle du Lac close to the Lemman Lake

For more info and latest updates see the conference web site at <<http://www.ada-europe.org/conference2007.html>>.

## Nov 4–8 — SIGAda 2007

From: Michael Feldman

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

Subject: Call for Participation:

SIGAda 2007 International Conference

Date: Sun, 11 Mar 2007 22:35:53

Organization: The George Washington University

Newsgroups: comp.lang.ada

SIGAda 2007

Call for Participation — SIGAda 2007 ACM SIGAda Annual International Conference

Submission Due Date: 16 May 2007  
Conference on Software Development for Safety, Security, and High Reliability Systems

November 4–8, 2007

Washington, DC, USA

Sponsored by ACM SIGAda, the ACM Special Interest Group on Ada

[www.acm.org/sigada/conf/sigada2007](http://www.acm.org/sigada/conf/sigada2007)  
(Approval pending by ACM)

SUMMARY

Constructing high reliability software is an engineering challenge that can now be met in many domains. The application of software engineering methods, tools, and languages interrelate to make the challenge easier or more difficult.

This conference focuses on safety, security and high reliability systems and the issues related to their development. Topics such as applied software engineering principles, conforming to specific safety or security standards, testing philosophies, programming language selection, etc. will be discussed.

The conference will gather industrial experts, educators, software engineers,

and researchers interested in developing, analyzing, and certifying reliable, cost-effective software. Technical or theoretical papers as well as experience reports with a focus on, or comparison with, Ada are solicited. Possible topics include but are not limited to:

- Safety, security and high integrity development issues
- Language selection for a high reliability system: Ada, C, C++, Java, C#, or others
- Use of high reliability subsets or dialects: Java HIP, MISRA C, Ravenscar, SPARK, etc.
- High reliability standards and their issues: DO-178B, EIC 61508, FDA, SAE, CC, EAL, etc.
- Process and quality metrics
- Analysis, Testing, and Validation
- Use of ASIS for new Ada tool development
- Mixed-language development
- Quality Assurance
- Performance Analysis
- High reliability software engineering education
- High reliability development experience reports
- Real-time networking/quality of service guarantees
- Fault tolerance and recovery
- Distributed system load balancing
- Static and dynamic code analysis
- Debugging complex systems
- Integrating COTS software components
- System Architecture & Design
- Information Assurance in the age of terrorism
- Use of new Ada 2005 features and capabilities
- Ada products evaluated per Common Criteria, Protection Profiles, or Security Targets

#### HOW YOU CAN CONTRIBUTE

SIGAda 2007 is interested in receiving contributions in six major categories: Technical Articles, Extended Abstracts, Experience Reports, Workshops, Panel Sessions, and Tutorials. Contributions from students are actively solicited.

- **TECHNICAL ARTICLES** present significant results in research, practice, or education. These papers will be double-blind refereed and published in the Conference Proceedings and in Ada Letters. Articles will also be considered for poster presentations.
- **EXTENDED ABSTRACTS** discuss current work for which early submission of a full paper may be premature. If your abstract is accepted, you will be expected to produce a full paper, which will appear in the proceedings. Extended abstracts will be double-blind refereed. Clearly state the contribution of the work being described, its relationship with previous work by you and others (with bibliographic references), results to date, and future directions.

- **EXPERIENCE REPORTS** present timely results on the application of Ada and related technologies to the design and implementation of applications such as the following: avionics, aerospace, automobile, command and control, consumer electronics, process control, transportation, trading systems, energy, medical systems, simulation, telecommunications, etc. Such reports will be selected on the basis of the interest of the experience presented to the community of Ada practitioners. You are invited to submit a 1–2 page description of the project and the key points of interest of project experiences. Descriptions will be published in the final program or proceedings, but a paper will not be required.

- **WORKSHOPS** 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. A list of planned workshops and requirements for participation will be published in the SIGAda 2007 Advance Program. Workshop proposals will be evaluated by the Program Committee and selected based on their applicability to the conference and potential for attracting participants. Proposals should state the problem or issue to be addressed, the coordinator(s), and criteria for participant selection.

- **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. Panel proposals should be 1–2 pages in length, identifying the topic, coordinator, and potential panellists.

- **TUTORIALS** offer the flexibility to address a broad spectrum of topics relevant to Ada, and those enabling technologies which make the engineering of Ada applications more effective. Submissions will be evaluated based on relevance, suitability for presentation in tutorial format, presenter's expertise, and past performance. Tutorial proposals should include the expected level of experience of participants, an abstract or outline, the qualifications of the instructor(s), and the length of the tutorial.

Final acceptance will be contingent on at least one co-author registering for and presenting the contribution at the Conference.

- **OUTSTANDING STUDENT PAPER AWARD.** An award will be given to the student author(s) of the paper selected by the program committee as the outstanding student contribution to the conference.

- **VENDORS.** Please contact S. Ron Oliver (SROliver at CSC.CalPoly.Edu) for information about participation at SIGAda 2007.

Please submit any questions on the conference to the Conference Chair, Alok Srivastava (Alok.Srivastava at AUATAC.Com).

#### IMPORTANT VISA INFORMATION FOR NON-US SUBMITTERS

##### General Visa Information

The sites <http://www.UnitedStatesVisas.gov> and <http://travel.state.gov> have information about obtaining a visa for those traveling to the United States. Both sites have links to websites for U.S. embassies and consulates worldwide. The embassy and consulate websites have very helpful information about procedures, timelines, communities served, required documentation, and fees.

##### Letters from ACM

International registrants should be particularly aware and careful about visa requirements, and should plan travel well in advance. All visa inquiries must be handled by ACM Headquarters. Please send your request for a letter in support of a visa application to Ashley Cozzi (acozzi at acm.org), and include your name, mailing address, and fax number, as well as the name of the conference you are attending. (Authors of papers/posters should also include the title). Please note that ACM does not issue formal "letters of invitation" to any of its conferences.

---

## Ada and Education

### Ada featured wikibook

*From: Martin Krischik*  
*<krischik@users.sourceforge.net>*  
*Subject: Ada Programming now a featured book*  
*Date: Thu, 26 Apr 2007 08:38:33*  
*Newsgroups: comp.lang.ada*

Ada Programming is now an Ada featured book on Wikibooks. The Ada book is the only book on programming languages which reached that state ☺.

Now anybody here with marketing experience [is encouraged to] to update our advertising template:

[http://en.wikibooks.org/wiki/Wikibooks:Featured\\_Books/Book2](http://en.wikibooks.org/wiki/Wikibooks:Featured_Books/Book2)

[See also "Ada wikibook to be published" in AUJ 28-1 (Mar 2007), p.9. —su]

### Ada 95 training course

*From: bex <andy.bissell@objektum.com>*  
*Newsgroups: comp.lang.ada*  
*Subject: Ada 95 training in the UK —*  
*23/07/07 — small group, limited space*  
*Date: 24 May 2007 09:02:39*

We are running a public Ada 95 training course 23rd – 26th July 2007 at our offices in Croydon (20 mins from

London Bridge), UK. This is an excellent hands-on course with limited space. Please see the link below for more details:

<http://www.objektum.com/objektum/indexcourse.asp?id=451>

[See also same topic in AUJ 28-1 (Mar 2007), p.9. —su]

## Praxis HIS — Q3 2007 Courses

*Public Course Dates for 2007 — UK*

Course 1 — “Software Engineering with SPARK”

10th – 13th September 2007 at the Praxis Offices in Bath.

Course 2 — “Black-Belt SPARK”

18th – 20th September 2007 at the Praxis Offices in Bath.

[See also same topic in AUJ 28-1 (Mar 2007), p.9. —su]

## AdaCore — InSight GNATbench webinar

*From: AdaCore Developer Center*

*Date: Wednesday May 16, 2007*

*Subject: InSight GNATbench webinar*

*RSS:*

[www.adacore.com/category/developers-center/development-log/feed/](http://www.adacore.com/category/developers-center/development-log/feed/)

Introducing the GNAT Pro InSight webinar series. AdaCore is launching a series of informative webinars dedicated to tools that make up the GNAT Pro development environment. 30 minutes long, each webinar will consist of a presentation, a demo, and a question and answer session conducted by an AdaCore technical expert.

The first technology in the series is GNATbench, the GNAT Pro Ada plug-in for Eclipse that supports both native (standard) Eclipse and Wind River’s Eclipse-based Workbench software development environment. The webinar will take place on Tuesday, June 5, 2007.

## Ada Distilled

*From: AdaCore Developer Center*

*Date: Monday June 4, 2007*

*Subject: Ada Distilled: An Introduction to Ada Programming for Experienced Computer Programmers*

*RSS:*

[www.adacore.com/category/developers-center/development-log/feed/](http://www.adacore.com/category/developers-center/development-log/feed/)

This book is aimed at experienced programmers who want to learn Ada at the programming level. It is not a “...for dummies” book, nor is it intended as a program design book. Instead, we highlight some key features of the Ada language, with coded examples, that are

essential for getting started as an Ada programmer.

This book emphasizes syntax, control structures, subprogram rules, and how-to coding issues rather than design issues. There are other really fine books available that deal with design. Also, this is not a comprehensive treatment of the Ada language. The bibliography lists some books that can help you once you have finished the material in this book. Think of this a quick-start book, one that enables you, the experienced programmer, to get into the Ada language quickly and easily. The examples use text-oriented programs for simplicity of learning. If you prefer to do the exercises for a GUI environment, check out the sites listed for compilers and tools so you can download bindings for programming in a variety of windowing environments, including Wintel, Linux, and Unix. Of particular interest to those wanting to explore Window style programming are GtkAda, CLAW, GWindows, and JEWL.

Copyright 2002, AdaWorks Software Engineering

License: Public Edition. Permission to copy if AdaWorks is acknowledged in copies.

In addition to downloading the ebook in pdf format, you can also download the sample programming files in the accompanying .zip file

## Ada-related Resources

### Ada IRC channel on Freenode

*From: Genro Kane Gupta*

*<genro@niestu.com>*

*Date: Sun, 04 Mar 2007 03:13:54*

*Subject: #Ada IRC channel on Freenode*

*Newsgroups: comp.lang.ada*

This is the semi-annual reminder of the existence of the #Ada channel on the Freenode IRC network. Now over five years old, the channel is open to all discussions related to the Ada language and its use. We welcome beginners and pros alike, and do our best to maintain a friendly, productive, and informative atmosphere. The channel has been growing steadily; we now average about 30 members and have active conversations daily.

Point your IRC client to [irc.freenode.net](http://irc.freenode.net) and join the #Ada channel. Come one, come all!

[See also same topic in AUJ 26-1 (Mar 2005), p.9. —su]

*From: Genro Kane Gupta*

*<genro@niestu.com>*

*Date: Sun, 04 Mar 2007 13:03:22*

*Subject: Re: #Ada IRC channel on Freenode*

*Newsgroups: comp.lang.ada*

> Are the discussions archived and searchable?

There's been some talk about doing that. Right now, the channel infobot (written in Ada, of course) logs the channel traffic, originally as a debugging feature. I guess since this question has been asked more than once, by more than one person, there's enough of a demand that I should set up a mechanism to format those logs in a useful way and put them on the web.

So I guess the answer to your question is, not yet, but soon! [...] Once the archives (what I have myself, and what I can scrounge from others) are up, I'll announce their availability here.

## Ada-related Tools

### Simple components

*From: Dmitry A. Kazakov*

*<mailbox@dmitry-kazakov.de>*

*Date: Sat, 19 May 2007 14:41:08*

*Subject: ANN: Simple components 2.5*

*Newsgroups: comp.lang.ada*

<http://www.dmitry-kazakov.de/ada/components.htm>

The library provides:

- Doubly-linked webs and lists
- Objects and handles to
- Parsers
- Persistent objects and handles to
- Persistent storage and handles to
- Pools
- Sets and maps
- Stacks
- Strings editing
- Tables (containers of strings)
- Unbounded arrays

Changes to version 2.4:

- Functions `Is_In` were added for doubly-linked webs and lists;
- For GNAT users GPS project files were added.

[See also same topic in AUJ 27-4 (Dec 2006), p.199. —su]

### Units of measurement for Ada

*From: Dmitry A. Kazakov*

*<mailbox@dmitry-kazakov.de>*

*Date: Sat, 19 May 2007 14:37:32*

*Subject: ANN: Units of measurement for Ada 2.3*

*Newsgroups: comp.lang.ada*

<http://www.dmitry-kazakov.de/ada/units.htm>

The library was designed for handling dimensioned values. It supports:

- Mixed unit arithmetic. Values in SI (*Le Système International d'Unités*) units can be mixed with irregular units (such as foot and yard) as long as the result is defined;
- Shifted unit support (a widely used



instance of a shifted unit is degree of Celsius);

- The type Measure is generic, parameterized by a floating-point type. A non-generic version based on the type Float is also provided;
- String to Measure conversion supports as wide set of irregular units as possible;
- GTK+ widgets for comfortable visual measurement unit selection.

Changes to version 2.2:

- For GNAT users GPS project files were added;
- Compiled with GNAT GPL 2007.

[See also same topic in AUJ 27-4 (Dec 2006), p.9. —su]

## Strings Edit

*From: Dmitry A. Kazakov  
<mailbox@dmitry-kazakov.de>  
Date: Sat, 19 May 2007 14:31:54  
Subject: ANN: Strings edit 1.9  
Newsgroups: comp.lang.ada*

[http://www.dmitry-kazakov.de/ada/strings\\_edit.htm](http://www.dmitry-kazakov.de/ada/strings_edit.htm)

The library provides an alternative approach to formatting strings. It supports:

- Integer numbers (generic, package Integer\_Edit);
- Integer sub- and superscript numbers;
- Floating-point numbers (generic, package Float\_Edit);
- Roman numbers (the type Roman);
- Strings;
- Ada-style quoted strings;
- Handling UTF-8 encoded strings;
- Generic scales formatting for plotting purposes.

Changes to version 1.8:

- For GNAT users GPS project files were added;
- Strings\_Edit.Generic\_Scale bugfix.

[See also same topic in AUJ 26-3 (Sep 2005), p.152. —su]

## Tables for Ada

*From: Dmitry A. Kazakov  
<mailbox@dmitry-kazakov.de>  
Date: Sat, 14 Apr 2007 21:01:24  
Subject: ANN: Tables for Ada v1.6  
Newsgroups: comp.lang.ada*

The library provides an implementation of tables indexed by strings. Binary search is used for names of known length. It is also possible to search a table for names of unknown length, i.e. to parse a string using some table. Table elements can be of any private type. Key-insensitive tables are supported.

Additions: Procedure Replace was added for replacing tokens by their offsets. A variant of the procedure Add now returns the offset of the inserted item.

<http://www.dmitry-kazakov.de/ada/tables.htm>

*From: Dmitry A. Kazakov  
<mailbox@dmitry-kazakov.de>  
Date: Sat, 19 May 2007 14:27:33  
Subject: ANN: Tables 1.7  
Newsgroups: comp.lang.ada*

<http://www.dmitry-kazakov.de/ada/tables.htm>

Tables provide string maps. Case insensitive and coalesced blank characters matches are supported. The string source can be matched against the table selecting the longest token in.

Changes to version 1.6:

- For GNAT users GPS project files were added.

[See also same topic in AUJ 26-3 (Sep 2005), p.153. —su]

## ASCL — Application Standard Components Library

*From: ASCL Project Account  
<michael.erdmann@snaflu.de>  
Date: Tue, 10 Apr 2007 20:38:48  
Subject: Announce: ASCL version 0.2.0  
open for comments  
Newsgroups: comp.lang.ada*

While writing applications I have always come across the same problems:

- I find components which might be reusable in later projects; fine, where to put them; or where did I put them?
- How to handle documentation; how to create reasonably formatted documents.
- Oops, I found a bug in a reused component; isn't it relevant for project XYZ I have done in the past? What about quality assurance ...?
- For each project I reinvent the configuration management procedures.

So I decided to provide a development framework for Linux which resolves the above issues.

For further information please refer to:

<http://home.gna.org/ascl/>

<https://gna.org/projects/ascl>

The home page is still under construction and the source code is not yet stable but it gives you an good idea about what I am intending to do!

## GNAT GPL 2007 Edition

*From: Nasser Abbasi <nma@12000.org>  
Date: Fri, 11 May 2007 19:41:15  
Subject: [AdaCore] GNAT GPL 2007 now available  
Newsgroups: comp.lang.ada*

We are pleased to announce the release of GNAT GPL 2007. This new edition includes all of the core features introduced in the recent language

revision, Ada 2005, including several features of the new real-time annex thanks to the new Linux-based Marte runtime developed in cooperation with the University of Cantabria. Ada 2005 features are described in the new Ada 2005 reference manual (now included in the GNAT GPL documentation).

GNAT GPL 2007 comes with version 4.0.2 of the GNAT Programming Studio IDE.

It is available on the Linux and Windows platforms.

GNAT GPL 2007 can be downloaded from the “Download GNAT GPL Edition” section on <https://libre.adacore.com>.

In addition, we have created a section on our website dedicated to technical information surrounding GNAT and Ada. The Developer's Center includes a developer's log giving updates on the GNAT technology, technical papers, code samples, and documentation.

[See also “GNAT GPL 2006 Edition” in AUJ 27-2 (Jun 2006), p.71. —su]

## GNU Ada Compiler

*From: Martin Krischik  
<krischik@users.sourceforge.net>  
Date: Sun, 20 May 2007 10:56:49  
Subject: [gnuada] R7 has been released  
Newsgroups: comp.lang.ada*

The GNU Ada Project [1] just released the 7<sup>th</sup>-generation Ada compiler distribution. This is a major release with lots of new functionality and extended Ada 2005 support.

From the release notes:

The R7 release is based on GCC 4.2.0 and GPL 2007 — both versions now offer Ada 2005 features.

The first version to be released is SuSE 10.2 x86\_64. 64-bit support is significantly improved and most of the special 64-bit patches are not needed anymore. Still AdaCore won't supply GNAT/GPL for 64 platforms.

ASIS is not available GNAT/GCC 4.2.0 as the current ASIS code is incompatible — we hope to get them in sync with GCC 4.3.0 again.

GPS is not available for GNAT/GPL 2007 as the new GPS 4.1.0 is not compatible with GNAT/GPL 2007. We might provide an older version later.

GNAT/GCC features all main GCC languages for convenient multi language development. GNAT/GPL has been reduced to just Ada and C — which reduces the download and installation footprint.

All tools have all been updated and Z-Shell has been added to the list tools. Z-Shell is more advanced than bash and a

lot of the tricky hacks needed for bash could be replaced with simpler and easier to understand techniques. We also work on a GNAT aware command line completion for Z-Shell.

[See also same topic in AUJ 27-4 (Dec 2006), p.200. —su]

*From: Martin Krischik  
<krischik@users.sourceforge.net>  
Date: Thu, 22 Mar 2007 21:06:29  
Subject: [gnuada] Solaris 2.9 release  
Newsgroups: comp.lang.ada*

We have a new release for Solaris 2.9 (I guess that Solaris 9 in marketing talk). OK, no Ada library available right now, but just the compiler, which is the most important thing. And tons of GNU tools. The tools Solaris comes bundled with are just horrible.

[1] <http://gnuada.sf.net>

## LLVM Compiler Infrastructure

*From: Duncan Sands <baldrick@free.fr>  
Date: Thu, 10 May 2007 22:01:02  
Subject: Re: Implementing an Ada compiler and libraries.  
Newsgroups: comp.lang.ada*

> Why not play on a C generator for the Ada language ?

I'm porting the GNAT Ada [front-end] to LLVM (<http://llvm.org/>), which replaces the GCC code optimizers with those of LLVM. LLVM has a C backend which turns its IR into C. So you can put Ada in and get C out. Unfortunately the C is not always compilable. For example, if you compile an instance of GNAT.Table with Natural as the index type, in the C you end up with the declaration of a very big C array, corresponding to Big\_Table\_Type. This array is rejected by the GCC C compiler as too big. Still, it works most of the time.

*From: Duncan Sands <baldrick@free.fr>  
Date: Fri, 11 May 2007 13:04:24  
Subject: Re: Implementing an Ada compiler and libraries.  
Newsgroups: comp.lang.ada*

> Interesting, I'll probably have a look at some point. Thanks.

It will most likely be available with LLVM 2.1. I will post an announcement here when I release it.

*From: Georg Bauhaus  
<bauhaus@futureapps.de>  
Date: Fri, 11 May 2007 10:39:38  
Subject: Re: Implementing an Ada compiler and libraries.  
Newsgroups: comp.lang.ada*

Given the C language, there might be a few platform issues in case you'd have to deal with such things as MIN\_INT - 1 ☺

There is at least one compiler front end that generates (readable) C. Maybe someone having worked with AdaMagic

by SofCheck has a few hints? Also, several Eiffel compilers use C as the target language.

## Imaging in Ada

*From: Jacob Sparre Andersen  
<sparre@nbi.dk>  
Date: Mon, 07 May 2007 13:30:19  
Subject: Re: Imaging in Ada  
Newsgroups: comp.lang.ada*

> What capabilities, in terms of libraries/packages/modules, does Ada have for doing basic image processing?

I am not sure whether there are that many published libraries. Some starting points might be:

<http://adaimgsvr.sourceforge.net/>  
<http://png-io.sourceforge.net/>

> For example, let's say you wanted to load a JPG or TIFF image and display it on the screen (...)

I would use the functions in GtkAda for this.

*From: Dmitry A. Kazakov  
<mailbox@dmitry-kazakov.de>  
Date: Mon, 7 May 2007 11:56:42  
Subject: Re: Imaging in Ada  
Newsgroups: comp.lang.ada*

In GtkAda that would be Gtk\_Image (or Gdk\_Pixbuf for a lower level handling):

```
procedure Gtk_New (Image : out
  Gtk_Image; Filename : String);
which internally uses libjpeg, I guess.
```

*From: Patrice <frett27@free.fr>  
Date: 8 May 2007 00:52:58  
Subject: basic imaging with ada  
Newsgroups: comp.lang.ada*

I got several packages for working with images on my web site SimpleJpegLib, permitting loading and saving jpeg files vision, an image processing package (resize, get pixels, set pixels ... )

<http://pfreydiere.free.fr>  
*From: Chip Orange  
<Chip.Orange@psc.state.fl.us>  
Date: Wed, 9 May 2007 16:59:36  
Subject: Re: Imaging in Ada  
Newsgroups: comp.lang.ada*

In addition to the others, if it's helpful there's a partial Ada binding for ImageMagick at: <http://home.gna.org/g2f/>

## GTKAda contributions

*From: Dmitry A. Kazakov  
<mailbox@dmitry-kazakov.de>  
Date: Sat, 7 Apr 2007 14:18:56  
Subject: ANN: GtkAda contributions 1.5  
Newsgroups: comp.lang.ada*

The software is a contribution to GtkAda, an Ada binding to GTK+. It deals with the following issues:

- Tasking support;
- Custom models for tree view widget;

- Custom cell renderers for tree view widget;
- Resource styles;
- Embeddable images;
- Some missing subprograms and bug fixes;
- Measurement unit selection widget and dialogs;

This new version adds:

- GLib.Values.Handling for dealing with GLib values in a generic way;
- Gdk.Color.IHLS package with the improved HLS color model;
- Gtk.Generic\_Style\_Button for creation customizable buttons controlled by style properties;
- Editable custom cell renderers.

[http://www.dmitry-kazakov.de/ada/gtkada\\_contributions.htm](http://www.dmitry-kazakov.de/ada/gtkada_contributions.htm)

*From: Dmitry A. Kazakov  
<mailbox@dmitry-kazakov.de>  
Date: Sat, 19 May 2007 14:09:35  
Subject: ANN: GTKAda contributions v1.6  
Newsgroups: comp.lang.ada*

The packages described here are proposed as a contribution to GtkAda, an Ada bindings to GTK+. It deals with the following issues:

- Tasking support;
- Custom models for tree view widget;
- Custom cell renderers for tree view widget;
- Resource styles;
- Capturing resources of a widget;
- Embeddable images;
- Some missing subprograms and bug fixes;
- Measurement unit selection widget and dialogs;
- Improved hue-luminance-saturation color model;
- Simplified image buttons and buttons customizable by style properties;
- Controlled Ada types for GTK+ strong and weak references;
- Simplified means to create lists of strings.

The new release is compatible with GNAT GPL 2007. Changes since the version 1.5:

- Move\_After and Mover\_Before missing in Gtk.Tree\_Store were added to Gtk.Missed;
- Set\_Expand missing in Gtk.Tree\_View\_Column was added to Gtk.Missed;
- Is\_A missing in GLib was added to Gtk.Missed;
- Gtk.Enums.String\_Lists was added for simplified creation of constant lists of strings;
- GLib.Object.Strong\_References and GLib.Object.Weak\_References were added to provide controlled wrappers around GTK+ strong and weak references;

- Gtk.Handlers.References was added to provide references to signal handlers which disconnect the handler's callback upon finalization;
- Class\_List\_Properties was added to the package Gtk.Widget.Styles;
- Gtk.Widget.Styles.Store was added for capturing property styles of a widget;
- For GNAT users GPS project files were added;
- Bug fix in editable renderers which caused exception propagation upon keyboard initiated editing;
- Bug fix in Gtk.Widget.Styles which prevented reading Gdk\_Color properly.

[http://www.dmitry-kazakov.de/ada/gtkada\\_contributions.htm](http://www.dmitry-kazakov.de/ada/gtkada_contributions.htm)

[See also same topic in AUJ 28-1 (Mar 2007), p.10. —su]

## QtAda binding

*From: Vadim Godunko*  
*<vgodunko@rostel.ru>*  
*Date: 12 Apr 2007 04:43:00*  
*Subject: ANN: QtAda preview available*  
*Newsgroups: comp.lang.ada*

QtAda is an Ada 2005 language binding to Qt 4.2. It allows to easily create a powerful graphical user interface in Ada. QtAda uses a native thread safe signal/slot mechanism, provides access to more than 70 Qt classes, provides an Ada-aware meta object compiler, supports the development of custom widgets and Qt Designer's custom widget plugins, supports loading at runtime of GUI forms from Qt Designer's UI files and so on.

The current snapshot is available at:  
<http://sourceforge.net/projects/qtada/>

*From: Vadim Godunko*  
*<vgodunko@rostel.ru>*  
*Date: 12 Apr 2007 06:51:56*  
*Subject: Re: ANN: QtAda preview available*  
*Newsgroups: comp.lang.ada*

> Qt 4.2 — is that the one for the upcoming KDE4?

Yes, you are right.

*From: Vadim Godunko*  
*<vgodunko@rostel.ru>*  
*Date: 11 May 2007 03:50:08*  
*Subject: QtAda: successful build with GNAT GPL 2007*  
*Newsgroups: comp.lang.ada*

I have successfully built a current QtAda snapshot with the GNAT GPL 2007 compiler on a x86/Linux box.

## Cheddar — Real-Time Scheduling Simulator

*From: Frank Singhoff*  
*<frank.singhoff@orange.fr>*  
*Date: Thu, 8 Mar 2007 16:24:25*

*Subject: Ann. of the annual release of Cheddar, a free real-time scheduling analyzer*

*Newsgroups: comp.lang.ada*

We're pleased to announce a new release of Cheddar. Cheddar is a free real-time scheduling tool. Cheddar is designed for checking task temporal constraints and buffer sizes of a real-time application/system. It can also help you for quick prototyping of real-time schedulers. Finally, it can be used for educational purposes. Cheddar is developed and maintained by the LISYC Team, University of Brest. Cheddar is written in Ada and should run on Solaris, Linux and win32 boxes.

Thanks to A. Plantec, P. Dissaux, G. Laurens, J. Hugues, T. Vergnaud, S. Edzang, M. Strauss, M. Baert, V. Hugel, C. Valle, F. R. Rocha and M. Zanconi for their help on this new release (tests, bug reports or new features)

The current release is now 2.0. A part of the source code of this new release is now automatically generated by platypus (see <http://cassoulet.univ-brest.fr/mme/>). If you are a regular Cheddar's user, we strongly advice you to switch to the 2.0 release due to the fixed bugs. The files "ChangesLog.txt" and "FIXED\_BUGS.txt" described the new features and the fixed bugs.

Cheddar is distributed under the GNU GPL license. It's a free software, and you are welcome to redistribute it under certain conditions; See the GNU General Public License for details.

Source code, binaries and documentations can be freely downloaded from <http://beru.univ-brest.fr/~singhoff/cheddar>

1) Summary of features:

Cheddar is composed of two independent parts: an editor used to describe a real-time application/system, and a framework. The editor allows you to describe systems composed of several processors which own tasks, shared resources, buffers and which exchange messages. Such a systems specification can be expressed with AADL, the architecture and design language from the SAE. The framework includes many feasibility tests and simulation tools.

Feasibility tests can be applied to check that task response times are met and that buffers have bounded size. When feasibility tests can not be applied, the studied application can be analyzed with scheduling and buffer simulations. Cheddar provides a way to quickly define "user-defined schedulers" to model scheduling of ad-hoc applications/systems (ex: ARINC 653).

The most important analysis tools are the following:

- Do scheduling simulations with classical real-time schedulers (Rate Monotonic,

Deadline Monotonic, Least Laxity First, Earliest Deadline First, POSIX queuing policies: SCHED\_OTHERS, SCHED\_FIFO and SCHED\_RR) with different type of tasks (aperiodic, periodic, task activated with a poisson process law, ...)

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

- Apply feasibility tests on tasks and buffers (without scheduling simulation):

- + Compute task response time bounds.
- + Apply processor utilization tests.
- + Compute bound on buffer size (when buffers are shared by periodic tasks)

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

- Tools to express and do simulations/feasibility tests with task precedences:

- + Schedule tasks according to task precedences
- + Compute Tindell end to end response time.
- + Apply Chetto and Blazewicz algorithms.

- Tools to run scheduling simulation in the case of multiprocessors systems

- Do simulation when tasks are randomly activated.

- Can run scheduling simulation on user-defined scheduler and task arrival patterns.

- Run user-defined analysis on scheduling simulation.

- ...

[See original posting for a detailed enumeration of new features and bug fixes provided by the 2.0 Cheddar release —su]

4) Work in progress:

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

- Improve Hierarchical schedulers support.
- Improve the user-defined schedulers services.
- Update the graphical editor in order to take into account GtkAda 2.2
- Improvement of message scheduling with
  - + End to end response time with message scheduling.
  - + Allowing message sending at any time of a task capacity
  - + Providing a way to user-defined message delay communication by specification of user-defined message scheduling (as user-defined scheduler)
- Provide a global memory analysis (task stack, text/data segments ...)

[See also same topic in AUJ 26-4 (Dec 2005), p.238. —su]

## Umbrello UML Modeller

*From: Oliver M. Kellogg  
<okellogg@freenet.de>  
Date: 14 Apr 2007 15:41:27  
Subject: Generating Ada from UML on  
Linux  
Newsgroups: comp.lang.ada*

Umbrello UML Modeller version 1.5.7-beta1 was released today. The main focus of this release was to improve the stability of the tool.

On the basis of bug reports from the Ada community, the Ada 95 code import and generation have seen many improvements.

For details, see  
<http://uml.sourceforge.net/feature.php>

[See also same topic in AUJ 25-2 (Jun 2004), p.50. —su]

## UnZip-Ada

*From: Gautier de Montmollin  
<gdemont@hotmail.com>  
Date: Fri, 16 Mar 2007 14:14:58  
Subject: Ann: UnZip-Ada v.20  
Newsgroups: comp.lang.ada*

After years of “maintenance” or “cosmetic” releases of UnZip-Ada, here is a more important one...

UnZip-Ada is a library in “pure” Ada (not a binding, no dependency to compiler, external library or operating system) for decompressing data from Zip files into files or data streams.

UnZip-Ada is free software.

Some features:

- \*new\*: task-safe
- \*new\*: parts not concerning decompression are now in a ‘Zip’ library; possibility of manipulating, searching or building Zip archives too
- \*new\*: Zip\_info's dictionary tree is now (at last!) balanced: optimally fast access to entries, much faster loading too
- decompression to file or to a data stream
- decompresses all Zip methods up to and including Deflate (v. 2.x); this includes files made by PKZip, Winzip, Info-Zip's Zip, Java's JARs, OpenDocument files, and many others
- decryption (v. 2.x)
- possibility to read the Zip directory once into a Zip\_info variable and use it to search quickly and decompress selected archived files, or traverse completely the directory to do some specific task
- options: junk\_directories, test\_only

Other information to be found in UnZipAda.txt (archived in unzada20.zip)  
<http://homepage.sunrise.ch/mysunrise/gdm/unzipada.htm>

## Ada-related Products

### AdaCore — GNAT Tracker version 2

*From: Jamie Ayre <ayre@adacore.com>  
Date: Tue, March 20, 2007 5:30 pm  
Subject: [AdaCore] Announcing the  
immediate release of GNAT Tracker  
version 2  
To: announce@adacore.com*

AdaCore is pleased to announce the deployment of GNAT Tracker 2, the GNAT Pro customer web server.

GNAT Tracker 2 enhancements include:

- An improved, user-friendly design providing easier information retrieval.
- A redesigned ‘dashboard’ providing a better overview and access to tickets, wavefronts, and technology downloads.
- A streamlined process for reporting technical issues.
- Easier access to useful information concerning your GNAT Pro subscription.
- Access to the Getting Started Guide.
- Access to AdaCore’s Developer Center directly via GNAT Tracker.

As always, for questions, or to inform us of issues that you encounter, please let us know through the GNAT Tracker report facility or by email to the usual [report@adacore.com](mailto:report@adacore.com) address.

AdaCore is also pleased to announce a new electronic mailing list dedicated to providing company news. Content includes recently published press releases and articles, product releases, upcoming events, and technical tips on Ada and GNAT Pro. To sign up to this mailing list, please visit:

[http://www.adacore.com/home/mail\\_list](http://www.adacore.com/home/mail_list)  
The AdaCore team

### AdaCore — GPS 4.1.0

*From: Arnaud Charlet  
<charlet@adacore.com>  
Date: Tue, March 27, 2007 6:02 pm  
Subject: [AdaCore] Announcing the  
immediate availability of GPS 4.1.0  
To: announce@adacore.com*

AdaCore is pleased to announce the immediate availability of GPS 4.1.0 for the following hosts:

```
ia64-sgi_linux
ia64-hp_linux
pa-hpux
sparc-solaris
sparc64-solaris
x86-linux
x86-solaris
x86-windows
x86_64-linux
```

This is a major release and includes many new improvements, notably:

- an improved Outline View, with new design and features
- python and pygtk available on all platforms allowing powerful scripting and customized dialog
- enhanced support for plug-ins with many new plug-ins available
- a more powerful smart completion engine and GUI
- local VCS file changes history
- keyboard macros
- automatic fixing of many new compiler messages
- an improved Emacs emulation
- a new quick search interface
- support for AUnit 2.0

All new features are described in the release note section on GNAT Tracker.

All distributions can be downloaded as usual using GNAT Tracker. We encourage you to install and start using this latest version of the GPS tool suite. As always, for questions, or to inform us of issues that you encounter, please let us know through the GNAT Tracker report facility or by email to the usual [report@adacore.com](mailto:report@adacore.com) address.

[See also “AdaCore — GPS 3.1” in AUJ 27-1 (Mar 2006), p.10. —su]

### AdaCore — GNATbench Eclipse Plug-in

*Tuesday April 3, 2007*

AdaCore Announces GNATbench Version 2.0 for Eclipse 3.2

SAN JOSE, Calif. — April 3, 2007 — Embedded Systems Conference — AdaCore, provider of the highest quality Ada tools and support services, announced the future availability of GNATbench Version 2.0, a major enhancement to its GNATbench Eclipse-based plug-in, which supports native (standard) Eclipse and Wind River’s Eclipse-based Workbench software development environments. GNATbench Version 2.0 has been upgraded with new capabilities to support Eclipse 3.2, the latest version of the popular open source platform for integrating software tools for application development. The upgrade provides development teams using the Eclipse 3.2 framework with advanced Ada language support and a fully integrated GNAT Pro Ada toolset to facilitate multi-language development, sophisticated editing, browsing, debugging, and comprehensive compilation.

GNATbench adds these capabilities and features as a plug-in extension to the standard Eclipse framework, so that no special version of Eclipse is required. Some of the new GNATbench 2.0 features for standard Eclipse include:

- Full Ada perspective is available
- Project “Clean” command is now supported

- Builder Console displays build commands and resulting messages
- Automatic code completion after the dot in unit names or record object names
- Project Navigator automatically refreshes after a build so results can be seen (executables, etc.)

GNATbench also supports large, embedded, real-time applications for Wind River VxWorks platforms used for general commercial and high-reliability/safety-critical industries. GNATbench seamlessly integrates Ada into Wind River's Eclipse-based Workbench environment to extend the power of Workbench to both all-Ada and mixed-language development.

All GNATbench 2.0 versions for standard Eclipse and Wind River Workbench feature:

- Wizards for creating new projects for Ada
- Enhanced Ada Project Navigator menu entries
- New GNAT knowledgeable project file editor with syntax highlighting
- New toolbar for Ada element creation (new projects, new files)
- Enhanced Ada editor, allowing color selection for numeric literals
- Support to automatically generate package bodies from package declarations
- Enhanced Import wizard for creation of new Workbench projects from GNAT projects
- New enhanced User's Guide

"AdaCore continually strives to provide customers with the latest technology available," said Robert Dewar, President of AdaCore. "AdaCore supports the largest variety of platforms for both native and cross embedded Ada development available today. We have updated our GNATbench plug-in to be even more powerful than before, and upgraded it to Eclipse 3.2."

Another important enhancement for GNATbench 2.0 is that Ada-specific semantic analysis is no longer accomplished by a background process running GPS (GNAT Programming Studio). That functionality is now provided via a Shared Library, invoked directly from Eclipse and resulting in enhanced robustness and responsiveness in the user interface.

#### Pricing and Availability

GNATbench 2.0 will be available as part of the GNAT Pro subscription. Please contact AdaCore (sales@adacore.com) for the latest information on pricing and supported configurations.

We invite you to visit us at the Embedded Systems Conference at the McEnergy Convention Center in San Jose, CA, April 1-5, 2007. You can find us in Booth #1924. Our engineers will be available to demonstrate GNATbench 2.0, answer any questions you may have, and provide you

with information on all of AdaCore's products and support services.

#### About AdaCore

Founded in 1994, AdaCore is the leading provider of commercial software solutions for Ada, a modern programming language designed for large, long-lived applications where reliability, efficiency and safety are critical. AdaCore's flagship product is GNAT Pro, which comes with expert online support and is available on more platforms than any other Ada technology. AdaCore has customers worldwide; see <http://www.adacore.com/home/company/customers/> for more information.

Use of Ada and GNAT Pro continues to grow in high-integrity and safety-critical applications, including commercial and defense aircraft avionics, air traffic control, railroad systems, financial services and medical devices. AdaCore has North American headquarters in New York and European headquarters in Paris.

[www.adacore.com](http://www.adacore.com)

[See also same topic in AUJ 27-3 (Sep 2006), pp.141-142. —su]

*From: Jamie Ayre <ayre@adacore.com>*  
*Date: Tue, May 15, 2007 6:21 pm*  
*Subject: [AdaCore] Announcing the immediate availability of GNATbench 2.0*

*To: announce@adacore.com*

AdaCore is pleased to announce GNATbench 2.0, a major enhancement to the GNAT Pro Eclipse-based plug-in. GNATbench 2.0 supports:

- Standard Eclipse 3.2 (Windows and Linux)
- Wind River's Workbench 2.5 (Windows)
- Workbench 2.6 (Windows and Linux).

For both standard Eclipse and Wind River Workbench, GNATbench 2.0 includes the following shared features:

- Support for Ada 2005, Ada 95, and Ada 83
- Full Ada perspective
- Automatic code completion after the dot in unit names or record object names, for subprogram formal parameters, and simple identifiers
- Wizards for creating new Ada projects
- Enhanced Ada Project Explorer
- Increased responsiveness and reliability
- New GNAT Project File editor with syntax highlighting
- Enhanced Ada editor, allowing color selection for numeric literals
- Automatic generation of package bodies from package declarations
- Changes to GNAT project files are effective immediately without having to restart
- New extended and enhanced User's Guide, including tutorials for building systems from initial project creation through executable generation

GNATbench 2.0 for Wind River Workbench features all the above, plus:

- Workbench 2.6 support, in addition to Workbench 2.5
- Enhanced Import wizard for creation of new projects from existing GNAT projects, including configuration as Downloadable Kernel Module (DKM) and Real-Time Process (RTP) projects
- Debugging Ada tasks is supported
- Wizards for creating new projects warn if an unsupported build spec is selected
- Visible views added to the Application Development perspective, and view shortcuts for the views that are not automatically visible

GNATbench 2.0 for standard Eclipse features all the shared capabilities plus the following:

- Eclipse 3.2 support
- Support for both native- and cross-compilation
- Builder Console display
- Project "Clean", "Build", and "Rebuild" commands
- Automatic syntax and semantic analysis on Ada files
- User-defined builder commands
- New toolbar for Ada element creation (new projects, new files)
- Automatic refreshes after a command so tool results can be seen (executables, automatically generated package bodies, etc.)

You may be interested to learn that AdaCore is launching a series of informative webinars dedicated to tools that make up the GNAT Pro development environment. The first technology in the series is GNATbench. For more information on the webinars, please visit:

<http://www.adacore.com/home/gnatpro/webinars>

GNATbench 2.0 can be downloaded as usual using GNAT Tracker. As always, for questions, or to inform us of issues that you encounter, please let us know through the GNAT Tracker report facility or by email to the usual [report@adacore.com](mailto:report@adacore.com) address.

## AdaCore — GNAT GAP 2007

*From: Romain Berrendonner <berrendo@adacore.com>*  
*Organization: AdaCore*  
*Date: Thu, 10 May 2007 10:36:40*  
*Subject: [AdaCore] GNAT GPL 2007 now available for GAP members*  
*To: announce@adacore.com*

We are pleased to announce the release of GNAT GPL 2007 available for the GAP community. To facilitate the job of distributing GNAT, we have synchronized the technology distributed to the Academic and Free Software communities. Hence, on the most popular personal platforms, your students will be

able to download GNAT GPL 2007 directly from [libre.adacore.com](http://libre.adacore.com) in a few days.

It is immediately available on the x86 and x86\_64 GNU Linux, Windows, SPARC Solaris platforms. It is also available on the platforms using Leon as the target and Linux as the host.

GNAT GPL 2007 can be downloaded from the Download section on GNAT Tracker. Please note that, for your convenience, GNAT Tracker can now be accessed directly from AdaCore's academic page:

<http://www.adacore.com/home/academia/>

This new edition includes almost all of the new features introduced in the recent language revision, Ada 2005. These are described in the new Ada 2005 reference manual (now included in the GNAT GPL documentation).

The GNAT GPL 2007 package also includes a new runtime for the x86 GNU Linux platform based on MaRTE (thanks to a cooperative effort with the Santander University team). This new runtime will be ideal for real-time courses thanks to its full support for Ada 95 Annex D. It will also, in the future, be the base for all the new Ada 2005 annex D features.

## AdaCore Joins Eclipse Foundation

*From: AdaCore Press Center*

*Date: Monday May 14, 2007*

*Subject: AdaCore Brings Industry Leading Ada Expertise to the Eclipse Foundation*  
*RSS: [www.adacore.com/category/press-center/feed/](http://www.adacore.com/category/press-center/feed/)*

NEW YORK, May 14, 2007 — AdaCore, provider of the highest quality Ada tools and services, today announced that it has joined the Eclipse Foundation as an Add-In-Provider. This membership level allows AdaCore to participate in the development of the Eclipse ecosystem, and to offer the breadth of their Ada expertise to the Eclipse community as a whole. Eclipse is an important part of AdaCore's corporate and product strategy. The company tapped into the growing popularity of the Eclipse framework early on, adding an Eclipse plug-in to the company's comprehensive GNAT Pro Ada tool-suite last year. And the company continues to provide customers with the latest technology available with the recent major upgrade to its GNATbench plug-in, offering new capabilities to support Eclipse 3.2, the latest version of the popular open source platform for integrating software tools for application development.

"We feel that the Eclipse Foundation and its open source emphasis perfectly match the Free Software concept that AdaCore has embraced with our GNAT Pro Ada development environment," said Robert

Dewar, President of AdaCore. "By joining Eclipse, AdaCore is now able to contribute to another valuable open source community with our Ada-aware, Eclipse-based GNATbench product line."

"Eclipse welcomes new Add-In-Provider member AdaCore to the foundation," said Mike Milinkovich, Executive Director of the Eclipse Foundation. "AdaCore supports a wide range of native and embedded platforms for Ada language development. We look forward to contributions from AdaCore's industry-leading Ada experts to enhance the Eclipse effort and provide improved usability and power for all Ada language users."

"As a fellow Eclipse member, we are pleased to have AdaCore bring their Ada expertise to the Eclipse Foundation," said Robert Day, Vice President of Marketing for LynuxWorks, and Chairman of the Eclipse Embedded Workgroup.

"AdaCore's Eclipse-based GNATbench will add new Ada capabilities to the Eclipse framework in general and we will be validating it within our Luminosity product. The latter is an example of the immediate benefit to the software community when new capabilities are added to the powerful Eclipse framework."

"Wind River is an Eclipse Strategic Developer and leader of the Device Software Development Platform (DSDP) top-level project," said Rob Hoffman, Vice President and General Manager of Wind River's Aerospace and Defense Business. "As a Wind River strategic partner and fellow Eclipse Foundation member, we are extremely pleased to have AdaCore add their expertise and products to the Eclipse framework. This step will enhance Ada capabilities within the framework and is an excellent example of how the core principles behind the Eclipse framework can provide immediate benefit for software tools suppliers and software developers alike. With AdaCore now a member of the Eclipse foundation these capabilities will only grow more powerful for our common users and the Eclipse community at large."

About GNATbench

GNATbench extends the basic Eclipse text editor to support sophisticated Ada-aware editing and browsing, supports Ada-aware debugging with the Eclipse debugger, and adds code generation using GNAT Pro for building systems in Ada and any of the other languages supported by Eclipse. The editor provides a code browsing capability to facilitate program comprehension and development. Since GNATbench treats any name as a hyperlink it makes code traversal easier. A more global approach involves selecting a given entity so that all its references are displayed in a search

results window. Selecting one of these references then brings up the corresponding file.

GNATbench provides a full Ada perspective, automatic code completion (after the dot in unit names or record object names), Ada package body generation from existing specifications, Ada project support and many other capabilities. GNATbench provides full Ada language support with the common look and feel expected by Eclipse developers.

GNATbench is available for most AdaCore-supported GNAT Pro platforms. These include the Native, Embedded and High Integrity Edition safety critical product lines, providing the largest number of supported Ada platforms available from a single vendor today. These are now all now available with the powerful GNATbench Eclipse IDE.

About the Eclipse Foundation

Eclipse is an open source community, whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. A large, vibrant ecosystem of major technology vendors, innovative start-ups, universities and research institutions and individuals extend, complement and support the Eclipse Platform.

The Eclipse Foundation is a not-for-profit, member-supported corporation that hosts the Eclipse projects. Full details of Eclipse and the Eclipse Foundation are available at [www.eclipse.org](http://www.eclipse.org).

About AdaCore

Founded in 1994, AdaCore is the leading provider of commercial software solutions for Ada, a modern programming language designed for large, long-lived applications where reliability, efficiency and safety are critical. AdaCore's flagship product is GNAT Pro, which comes with expert online support and is available on more platforms than any other Ada technology. AdaCore has customers worldwide; see <http://www.adacore.com/home/company/customers/> for more information.

Use of Ada and GNAT Pro continues to grow, especially in high-integrity and safety-critical applications, including commercial and defense aircraft avionics, air traffic control, railroad systems, financial services and medical devices. AdaCore has North American headquarters in New York and European headquarters in Paris. [www.adacore.com](http://www.adacore.com) [See also "Hibachi — Eclipse Ada Development Tools" in this issue —su]

## Aonix Ada Commitment

*Title: Java's Great, But We Still Love Ada Too!*

*Author: Dave Wood (Aonix VP Marketing)*  
*URL: <http://www.aonixnews.com/may07/notesfromtheedge.htm>*

Aonix proudly has been in the Ada business for over a quarter century, since the original founding of Alslys by Ada's creator, the late Jean Ichbiah. Alslys later merged with another Ada giant, Telesoft, and also absorbed German Ada supplier SysTeam. Through the years, we have been Ada pioneers — the first to provide an off-the-shelf safety certifiable solution, the first to provide an affordable, fast, and user-friendly desktop Ada solution (ObjectAda for Windows), the first Ada supplier to support the Eclipse environment, and the first commercial adopter of the lightning fast compilation technology created by Ada 95 designer Tucker Taft.

Many of our more visible marketing activities these days focus on our emerging real-time Java technologies: PERC Ultra and PERC Pico. We're aware that this visibility sometimes can be a cause for consternation among our Ada clientele. I've enjoyed discussions recently with some prominent Ada advocates such as Dirk Craeynest, the co-founder of Ada-Belgium and vice president of Ada-Europe. Dirk eloquently points out that our claims for Java's abilities in real-time and embedded development are equally applicable for Ada, and have long been proven by Ada's impressive string of industry success stories.

Of course, Dirk is correct, but I'd like to take the opportunity to set the record straight about Aonix' views of Ada and Java.

First, it is important to understand that Aonix remains a successful and committed player in the Ada business. The Aonix product roadmap for 2007 includes no less than nine product releases. Core engineering activities currently underway include enhanced support for multi-partition kernels, Ada 2005 features, and the continuation of our best of breed Eclipse integration which we are now contributing to the open source community for the benefit of everyone, including users of GNAT. Aonix will continue to provide Ada technology and support as long as the market provides customers to consume it.

At the same time, Aonix is not an "Ada company" in the sense of being focused on Ada only, and we have not been one since the days of Ichbiah. In the early days of Ada, most of the Ada suppliers were companies focused only on Ada. But the mark of a healthy and growing company is that it offers multiple complementary product lines aimed with a clear market focus. In that spirit, Aonix offers multiple product lines today. In addition to Ada products, we offer technologies in real-time Java, the

TeleUSE line of Motif UIMS products, and the StP line of modeling tools. Our focus and specialization is in supplying tools and services for rigorous, mission-critical applications development. All of our products fit under that umbrella.

It is fair for some outside observers to wonder why we may sometimes seem to spend more visible energy promoting Java than Ada or our other products. To understand the reasons, one has to consider the current state of the markets for these products.

Ada is not a new technology. It has been in existence for close to a quarter century. It is well understood, with a deeply entrenched user base. We see the market for Ada as stable and healthy. There exists a large body of developers who, like us, believe that Ada is an elegant and powerful language supported by great products. Without a doubt, it is worthy of use, or at least fair consideration, on every major mission critical application.

By contrast, Java (at least in the embedded space, and especially in the real-time space) is a newcomer. Its value is very well understood by the enormous community of Java users, but it's applicability to our target audience is not as well understood by the market. Like Ada, Java has been shown to provide tremendous productivity, quality, and reliability advantages over C and C++. Further, our PERC technology has paved the way to show that Java not only is suitable, but indeed advantageous in the realm of complex real-time and embedded systems. We know that PERC is great, but the only way to communicate its value to our target audience is through visible marketing efforts. The target market already knows about the Ada value proposition.

Although significant advertising attention is paid to Java for the reasons cited above, I have to point out that we conduct substantial and continuous marketing of Ada as well. Marketing is far more than advertising. For one thing, we select trade shows that are attractive to both Ada and Java developers. Our newsletters always present both Ada and Java news. We do about half of our press releases on Ada topics. At public events we give Ada and Java equal treatment in our graphics and materials. Further, our R&D team is exactly evenly split between Ada and Java activities.

Truthfully, our Ada and Java solutions share a similar story: safe, predictable, reliable, scalable, modular, error-resistant, and highly productive tools. We won't bother to argue which language is better — we prefer to let the customer make that judgment. Our job is to inform our customers about our offerings, not to be language evangelists. Evangelism is best left to independent voices.

My message is this: Java is not the enemy of Ada. I think it can be agreed that C/C++ is the common enemy of everyone interested in sustainable growth in software complexity. C/C++ does for software what oxidation does for raw iron: makes it unsuitable for complex, reliable, and long-lived applications. We believe the industry wins every time a design choice is made for Ada or Java rather than C/C++, and that's what we're all about.

Email us at: [editor@aonixnews.com](mailto:editor@aonixnews.com)

## **Aonix — ObjectAda for Ardence ETS**

*Aonix's Latest ObjectAda Supports New ETS RTOS Version*

Aonix-Ardence integration enhances Ada real-time Windows environment  
 San Diego, CA, April 23, 2007

Aonix®, a provider of solutions for safety- and mission-critical applications, announced that ObjectAda V8.2 has been ported to the latest Ardence ETS real-time operating system (RTOS). This advanced version of ETS, Version 13, offers many new features, including kernel-level memory protection, a critical feature for Aonix's customers who need to bring real-time, secure memory applications into the Windows environment.

Ardence's Phar Lap ETS, a hard real-time RTOS designed for applications targeting execution on x86-based platforms, now boasts improved connectivity support. With this edition, ETS delivers TCP/IP multicast support and complete access to ETS TCP/IP APIs enabling development of network drivers and for support of higher layer protocols. Its 256 priority levels enable developers to meet the needs of real-time applications that are highly complex, guaranteeing that critical thread context switches and yields to threads of higher priority occur in the 500 nanosecond to less than 2 microsecond range. The ETS kernel-level memory protection safely isolates critical parts of the application from all other parts, ensuring that mission-critical components cannot be

"Aonix has been an ETS reseller for more than ten years with a great track record of success in supporting complex real-time systems," stated Stephen M. Woodard, VP, Product Line Executive, Embedded/OEM Division at Ardence, a Citrix company. "We're delighted they continue to deliver the value offered in our real-time solutions, and we look forward to jointly servicing the Ada community with them."

"The improved interrupt handling support and superior performance of ETS 13 is critical to our customers," said Gary Cato, Director of Strategic Alliances at Aonix. "Our latest ObjectAda technology

complements the improvements made with ETS, giving our customers the solution needed to create hard real-time applications for Intel-target applications.”

The Aonix ObjectAda for Windows brings the improvements of ObjectAda 8.2 to the Windows development platform. In integrating current Windows improvements with the Aonix Ada 95 compiler, Aonix has delivered enhancements to the object code and symbolic debugging information generation and provided full compatibility with the Microsoft Visual Studio .NET 2003 development tools. Recognizing the growing number of large-scale Ada projects, ObjectAda 8.2 for Windows targeting x86 processors offers dramatic performance improvements for developers linking executable files or initiating debugging sessions for large programs.

#### Shipping and Availability

ObjectAda for Windows targeting Ardence ETS is available immediately and is priced starting at \$15,499 (U.S. dollars) per seat with volume discounts available.

#### About Aonix

Aonix offers mission- and safety-critical solutions primarily to the military and aerospace, telecommunications and transportation industries. Aonix delivers the leading high-reliability, real-time embedded virtual machine solution for running Java™ programs deployed today and has the largest number of certified Ada applications at the highest level of criticality. Headquartered in San Diego, CA and Paris, France, Aonix operates sales offices throughout North America and Europe in addition to offering a network of international distributors. For more information, visit [www.aonix.com](http://www.aonix.com).

#### About Ardence, a Citrix Company

Ardence, a Citrix Company, develops software platforms for the on-demand world. The Ardence Software-Streaming Platform increases IT agility by delivering operating systems and applications on-demand from the network to desktops, servers and devices. The Ardence Embedded OEM Development Platform delivers market-leading operating system control capabilities that enable OEMs to increase system performance and manageability.

Citrix Systems Inc., (NASDAQ: CTXS) the global leader in application delivery infrastructure, acquired Ardence in January 2007. Ardence is headquartered in Waltham, MA and has a global distribution and reseller channel. Additional information about Ardence can be found at [www.ardence.com](http://www.ardence.com).

## Aonix — AonixADT goes open source

### *Aonix Leads Ada Eclipse Development Tools Project Initiative*

AonixADT goes open source, Offered as core technology for ADT project Embedded Systems Conference, San Jose, April 2, 2007

Aonix®, a provider of solutions for safety- and mission-critical applications, announced today that it is leading a new Eclipse-based initiative to create an Ada Development Toolkit (ADT) project and will collaborate with the Eclipse Foundation™ toward that end. With this new initiative, Aonix is offering its Eclipse™ plug-in technology to the open-source community and has proposed to the Eclipse Foundation that its AonixADT technology be used in the ADT project as the core baseline technology. Opening AonixADT to the user community through an authorized Eclipse project will provide the Ada community free access to the open and extensible AonixADT™ solution.

Since AonixADT can be used with a large set of existing plug-ins for third-party tools, including support for source-code configuration management, it is a powerful application-building enabler. Aonix, an early participant in the Eclipse™ open tools movement, has provided Ada Development Toolkit plug-ins for Aonix and GNAT compilers since 2004. With Eclipse Foundation sanctioned Ada plug-ins, the Ada community will enjoy the same benefits previously available for other languages.

By standardizing on a single IDE, companies that adopt Eclipse may eliminate the efforts traditionally required to learn and maintain multiple development environments. Within Eclipse, they can combine their extensive collection of in-house development tools with best-of-breed commercial products. This integration streamlines interactions within the development environment, simplifying the work process used by each developer on a daily basis, saving significant time and reducing overall development costs.

“We’re very pleased with the offer from Aonix to lead an ADT Project initiative,” said Doug Schaefer, QNX Software Systems, Eclipse CDT Project Lead. “Adding Ada to the Eclipse list of supported languages is further evidence of its broad acceptance as the best possible cross-language, cross-industry development IDE.”

The ADT project benefits any organization using popular real-time operating systems that support Eclipse in their development toolkits. Plug-ins built on a standard ADT project baseline would work together with other standard eclipse

plug-ins supported by products such as Luminosity from LynuxWorks™, Momentics® from QNX® Software Systems, and Workbench from Wind River® Systems to name a

“As a user of the Eclipse platform, the CDT project and partner of Aonix, we are happy that ADT is helping to bring these together”, said Robert Day, VP marketing at LynuxWorks and chairman of the Eclipse Embedded Workgroup. “This initiative also shows the openness of Eclipse platform architecture and how its widespread usage in the embedded industry is extending daily to meet embedded users’ needs.”

“As a Strategic Developer in Eclipse and lead of the Device Software Development Platform (DSDP) top-level project, Wind River has actively promoted Eclipse adoption, collaboration, and contribution in the Device Software Development space,” says Andrew Lyons, Director of Developer Tools at Wind River. “We strongly support Aonix decision to initiate and lead the ADT project in Eclipse and look forward to a growing Ada community in Eclipse. Ada customers of Wind River Workbench, an Eclipse-based integrated development suite that supports both VxWorks and Linux operating systems, can only benefit from broad access and enhancement of tools like ADT.”

AonixADT provides Ada-project awareness, an Ada-language sensitive editor, Ada-language compile and build capabilities, along with a complete Ada debugger interface. ADT project awareness allows full library hierarchy manipulation and Ada program units can be conveniently inserted or removed from Ada projects. The language-sensitive editor provides complete language awareness with syntax color coding and template completion. Symbolic debugging is integrated within the Ada-language sensitive editor. The build interface offers complete access to the Aonix ObjectAda compile and build capabilities.

#### Shipping and Availability

ObjectAda Real-Time Raven for Windows platforms targeting the PowerPC processor family is immediately available. Prices range from \$15,000 to \$30,000 in the U.S. depending on bundle options plus runtime license fees. Quantity discounts are available. DO-178B certification materials are available and priced based on board support package and other project-specific requirements.

[See also “Hibachi — Eclipse Ada Development Tools” in this issue —su]

## Hibachi — Eclipse Ada Development Tools

From: Tom Grosman <[grosman@aonix.fr](mailto:grosman@aonix.fr)>



*Organization: Eclipse Community*  
*Date: Thu, 5 Apr 2007 23:40:55*  
*Subject: [ANN / Req for Participation]*  
*Hibachi — The Eclipse Ada*  
*Development Tools Project*  
*Newsgroups: comp.lang.ada*

I have submitted a proposal to the Eclipse Management Organization for the creation of a vendor-neutral IDE for Ada development in Eclipse. Eclipse is an open source community whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle.

ADT, the Ada Development Tools project (codename Hibachi), a sub-project of the Tools top-level project analogous to JDT, the CDT and others, will follow the Eclipse development process and be made available under the Eclipse Public License (EPL).

As the Eclipse platform has grown in robustness, maturity and level of adoption, the Ada community has expressed the desire for a first-class Eclipse Ada development environment that can leverage the openness, extensibility and widespread availability of the exemplary tools found within the Eclipse framework. Various Ada vendors have announced the availability of Eclipse plugins for their Ada tools. There is necessarily much duplication of effort between these plugins, effort that could be better spent advancing the robustness and functionality of a common Eclipse IDE. Especially resource consuming is the integration of each proprietary environment with the myriad of additional tools that are often used in the large scale development environments in which Ada is frequently found. It is the goal of this project to marshal the resources of the Ada community of vendors, open source projects, third party tool providers, industrial and academic partners in order to create a vendor neutral, extensible, best-of-breed, Ada plugin for Eclipse.

Since it is the goal of the Hibachi project to create a framework that can be exploited by all Ada vendors, regardless of underlying technology, the participation of representatives of the various Ada and tools vendors is highly desired in order to ensure that the best engineering design decisions are being made in terms of openness and extensibility.

The ADT proposal can be seen at <http://www.eclipse.org/proposals/adt/>

Comments are welcome via the adt newsgroup — <news://news.eclipse.org/eclipse.tools.adt> newsgroup.

(The Eclipse newsgroups are password-protected to protect them from spam and trolls. Visit

<http://eclipse.org/newsgroups/index.html> and request a free password to access the newsgroups. The same password can be used to browse the Eclipse mailing list archives.)

Those interested in Hibachi are also invited to contact me directly.

I'd like to thank Aonix for contributing the sources to AonixADT as the baseline for Hibachi, and agreeing to contribute engineering and project management resources to the project. I'd also like to thank Doug Schaefer, CDT project lead and member of the Tools Project Management Committee for agreeing to act as a mentor, and for the guidance and enthusiasm he brings to the project.

For information on the Eclipse development process, refer to ([http://www.eclipse.org/projects/dev\\_process/development\\_process.php](http://www.eclipse.org/projects/dev_process/development_process.php))

I look forward to your comments, suggestions and participation.

*From: Tom's Hibachi musings*  
*Date: April 5 2007*  
*Subject: Going Live*  
*RSS: <http://hibachitom.blogspot.com/feeds/posts/default>*

The Hibachi (ADT) proposal is now online and open for comments at <http://www.eclipse.org/proposals/adt/> and the newsgroup <news://news.eclipse.org/eclipse.tools.adt> .

People will have surely noticed that the project is known by two names, ADT and Hibachi. ADT is the acronym for the official project name (Ada Development Tools). Hibachi is a codename. I noticed that Eclipse allows projects to be known by codenames (like Mylar). I thought that a codename would be a good way to pay tribute to Jean Ichbiah, the chief designer of Ada who had passed away just the month before. Also, although we (Aonix) tried to always refer to our plugin as AonixADT, to avoid confusion with an eventual Eclipse ADT project that we had envisioned, there was some namespace creep which I didn't think would be helpful when it came time to build a community. So while the project will be known as ADT, which is unavoidable for a sibling of CDT and JDT, I will try to use Hibachi whenever possible in memory of Jean Ichbiah and to avoid confusion with the Aonix product.

As far as the name Hibachi, until I hit on it, the leading candidate was Beaujolais, after the Beaujolais Effect. Others suggested "Green", Jedi, Kermit, and various more obscure references that have meaning for the old-timers here at Aonix, the company that Jean started to turn the Ada language he designed into a technology and products.

I only came up with one complete anagram for Ichbiah — Hibachi.

*From: Tom's Hibachi musings*

*Date: April 20 2007*  
*Subject: Growing interest*  
*RSS: <http://hibachitom.blogspot.com/feeds/posts/default>*

I've been encouraged by the response from the Ada vendors to the project proposal. The major active Ada vendors are on board and the combined level of commitment leads me to believe that we will have enough community involvement to create a viable project. It is interesting to see how the principles of cooperation and competition play out within the Eclipse open development environment.

I think the Ada vendors have all come to the same conclusion, that there's more to be gained by pooling resources and cooperating, then by each one going their own way. What I've been seeing is that our plugin (AonixADT) is more and more being compared with CDT and JDT, rather than Ada IDEs from other vendors. And with 10+ committers on a project like CDT, and even more on a project like JDT, it's pretty obvious that any single Ada company's technology, no matter how good it currently is, will fall further and further behind customer's expectations.

*URL: <http://www.eclipse.org/proposals/adt/>*  
 Ada Development Tools (ADT)

Proposal posted 2 months ago

Introduction

The Ada Development Tools is a proposed open source project under the Eclipse Tools Project.

This proposal is in the Project Proposal Phase (as defined in the Eclipse Development Process document) and is written to declare its intent and scope. This proposal is written to solicit additional participation and input from the Eclipse community. You are invited to comment on and/or join the project. Please send all feedback to the <news://news.eclipse.org/eclipse.tools.adt> newsgroup.

Background

The Ada language was designed by a team led by Jean Ichbiah at CII Honeywell Bull and became an ISO standard in 1983. Ada is a type safe procedural programming language with support for many object oriented features, as well as support for runtime checking, exceptions, generics and parallel processing. Subsequent revisions to the language, commonly known as Ada 95 and Ada 2005, were created to correct perceived shortcomings to the language and strengthen support for OO programming practices. Ada is frequently found in mission-critical and safety-critical systems such as those used in avionics, aeronautics and rail transportation.

Ada compilers and development environments are available as proprietary commercial products as well as open source projects. Different Ada environment providers often have their own IDE. In fact, due to the longevity of Ada, some providers offer multiple IDEs for their various Ada tools. The same challenges (editor functionality, integration with third party tools and CM systems, displaying debugging information, etc.) have been faced and solved in a myriad of ways, with varied results.

#### Scope

The objectives of the ADT project are to:

- Create a standard, vendor neutral Ada development environment for Eclipse. ADT should maintain as close as possible, the look and feel of the CDT and JDT, in order to mesh well with the Eclipse environment, as well as aid developers who need multi-language capabilities, or who need to transition between languages.
- Provide a standard Ada environment with which third party tool developers can easily work, regardless of the choice of underlying compiler technology. Such tools would include, but are not limited to, modeling tools, editor tools, static and dynamic code analysis tools (ex coverage, profiling), refactoring tools, configuration management tools.
- Act as an additional reference for the ongoing process of making the Eclipse framework more able to easily support a variety of native and embedded language development environments (through interaction with projects such as CDT, DSDP and Safari.)

#### Description

We propose to contribute the sources to AonixADT as a baseline for the Eclipse ADT project. AonixADT, modeled after the CDT and JDT, is a commercial plugin for Ada development. It has been in development since 2004 and in use by industrial customers since 2005. AonixADT currently provides supports for ObjectAda (Aonix's own Ada 95 offering) as well as GNAT "commercial" (GMGPL'ed) and non-commercial (GPL'ed) open source offerings. GNAT is the Ada component of GCC, the Gnu Compiler Collection. AonixADT v3.2.1 is available on Windows, Linux x86 and Sparc Solaris, and supports native as well as cross (i.e. embedded) development.

AonixADT's core features include Ada project configuration and navigation, Ada builder, customizable project creation wizards, coloring Ada 95 syntactic editor with semantic code assist and semantic browse/search capabilities, Ada debugger interface, an Ada source reformatter as well as support for various Eclipse/CDT basic functionalities (tasks, bookmarks, error navigation from the problem view, etc.)

AonixADT is organized into core features, and separate toolchain support. The primary focus of the ADT project will be on core functionalities. The integration of specific Ada toolchains will be done either as components within ADT (assuming sufficient community interest), or by interested parties (vendors, developers) outside of the context of the ADT project. It is the goal of ADT to provide a well documented manner to add toolchains, and to be structured and developed in such a way as to make common core functionality available to all toolchain components.

#### Organization

##### Initial committers and contributors

The initial committers will initially focus on re-architecting/extending the code base to provide an open, well documented API. Our agile development process will follow eclipse.org's standards for openness and transparency. Our goal is to provide the infrastructure and APIs needed to allow the integration of additional toolchains. We also plan to help improve the Eclipse platform by submitting patches and extension point suggestions. The initial team will consist of:

- Tom Grosman (Aonix): project lead (half-time, at first mostly devoted to project planning, administration, organization, documentation and community building. As those responsibilities begin to demand less time as the project matures, more time will be spent in a development role)
- Adam Haselhuhn (Aonix): contributor, lead developer (half-time)
- New Hire (Aonix—in progress): developer (half-time, partly or fully replacing Adam Haselhuhn)
- Equivalent to 1 or 2 full time committers drawn from the interested community

##### Interested parties

We have had discussions with various customers and members of the Ada community who have expressed interest in contributing to an open source ADT project. As this project proposal is publicized, we expect to be able to name specific organizations and people who will be interested in becoming committers, contributors, testers and users. Here are the types of people and organizations who would be interested:

- All Ada vendors (either with or without an existing Ada plugin product)
- The Gnu Ada open source project
- The Mac Ada open source project
- Makers of complementary Ada development tools (analysis tools, refactoring tools, etc.)
- Current users of AonixADT
- University faculty and students

#### Developer community

The developer community will be mostly drawn from the list above. Additionally, since a skillset combining Ada and Eclipse plugin development experience is almost certainly restricted to a handful of individuals at this point, developers with knowledge of the Eclipse API and Eclipse plugin development, but who may not be familiar with Ada (ex- JDT, CDT, Photran developers) are solicited.

We are expecting and will actively pursue during the proposal and incubation phases, active participation from Ada vendors. It is possible that they may not be able to budget a full-time committer, but perhaps each vendor can contribute (funding, or a partly funded developer) in order to create one or more full-time committers.

#### User community

The existing Ada developer community will be the primary user base. This includes an important presence in academia since Ada is frequently used in programming courses.

In addition, Aonix has made available a free download of AonixADT. On average 50 people a week download the product, leaving a valid email address. These 1000+ downloaders will be contacted to inform them of the proposal and solicit their participation and/or comments.

#### Tentative Plan

The details of the development/release plan will depend upon the level of participation, as well as the composition of the contributors. This initial plan is based on the equivalent of 2–3 full time committers with a favorable project review occurring in early June 2007. If these assumptions prove to be inaccurate, the schedule will need to be adjusted or shifted accordingly.

##### 1) ADT 0.5.0 — Initial release. 8/2007.

The goal of this release is to bring the initial code contribution under the Eclipse project framework and make sure that the resources, procedures and roles necessary to release an Eclipse project are understood, defined, put in place and functioning properly. To incite the user community to take this initial version of ADT out for a spin, 0.5.0 will include initial versions of ObjectAda and GNAT toolchain components. It is anticipated that the committers will be coming up to speed during the development of this release, so part of the role of the committers already familiar with ADT will be to help the other committers (and contributors) to obtain a working understanding of ADT internals.

##### 2) ADT 1.0.0 — Internals rearchitecture. 01/2008

The goal of this release is to re-architecture ADT to provide a stable,

documented API and extension points. This will need to be addressed on a subsystem-by-subsystem basis. No major new functionality is planned, however bugs discovered in 0.5.0 will be addressed. This release will include synching with the latest stable CDT delivery. It is probable that the learning curve of new committers will continue during this phase. The first “ready for prime time” version of online help will also be included.

### 3) ADT 1.1.0 Internal structures and algorithms 04/2008

This release will include a reviewing of the Ada Model, extending it as necessary to better support generics, designing and implementing an AST, reimplementing References searches to remove dependencies on ASIS.

### 4) ADT 1.2.0 Large, complex environment support. 07/2008

Extend the builder to better handle conditional and multilanguage builds (either via the current Ada Configuration hooks, or if possible by using/adapting the builder used by CDT). Integration / Compatibility with DSPS (TM and DD) for embedded Ada toolchains (RTOS and Bare).\*

\*Our understanding is that currently little or nothing needs to be done to make use of Target Management functionalities. Further study and analysis will need to be done to better understand the current state of DSDP/DD and how that will fit with ADT. However, we view robust support of embedded development to be critical in the acceptance and success of ADT, and so will need to closely monitor and interact with the relevant DSDP projects.

### 5) ADT 1.3.0 Added user functionality.

At this point, a year after project creation, ADT can be considered to be a robust stable platform for continued development, and further releases will focus less on internals and more on adding user functionality (like support for Ada 2005, code folding, “AdaDoc”, Ada Unit Testing, refactoring) and adding APIs and graphical interfaces to allow coupling of third party tools (such as coverage tools, profilers, call tree generators, with tree generators, coding standard enforcers, round-tripping modeling tools, etc.)

If experience is a guide, individual vendors may be urged by their users to implement particular functionality sooner rather than later. Of course, the ADT project will welcome all contributions and try to incorporate them into our releases provided that they do not entail major disruptions to the release schedule.

#### Codename

As a tribute to Jean Ichbiah the designer of Ada, we propose to give ADT the codename “Hibachi”. Hibachi is:

- a) A small portable charcoal grill (US)
- b) Slang for a basketball player “in the zone”
- c) An anagram for (Jean) Ichbiah, b. 25 March 1940 d. 26 January 2007

[See also “Aonix — AonixADT goes open source” and “AdaCore Joins Eclipse Foundation” in this issue —su]

## McKae Technologies — Avatox 1.7

*From: Marc A. Criley <mc@mckae.com>*

*Date: Wed, 25 Apr 2007 20:07:36*

*Subject: Announce: Avatox 1.7 is now available (with pre-alpha AXF2Ada)*

*Newsgroups: comp.lang.ada*

Avatox (Ada, Via Asis, To Xml) is an application that traverses Ada compilation units and outputs their ASIS representation(s) as XML document(s) in the Avatox Xml Format, AXF, file extension “axf”. The format of the XML in the document can be configured, and supplemental source annotations can be generated.

Changes since version 1.6:

- Added three attributes needed to accurately represent source code structure.

Also included in this release is an in-work “pre-alpha” version of axf2ada.xml, an XSLT stylesheet that converts AXF into Ada source code. It is a work in progress, and currently can regenerate the Avatox source code from its AXF representation. For more information about axf2ada, see the Avatox web page:

Avatox 1.7 is available at [www.mckae.com/avatox.html](http://www.mckae.com/avatox.html).

[See also “McKae Technologies — Avatox 1.4” in AUJ 28-1 (Mar 2007), p.16. —su]

## Praxis HIS — SPARK Toolset 7.5

*SPARK Release 7.5*

May 2007

Praxis High Integrity Systems is pleased to announce the immediate availability of Release 7.5 of the SPARK language and the SPARK toolset.

Full details of all language and tool changes can be found in the release notes for release 7.5.

Supported, professional customers will receive upgrade packages immediately.

Buyers of the “SPARK Book” by John Barnes can now download upgrade packages to bring their toolset and documentation up to release 7.5.

Release 7.5 includes many significant improvements, including:

Corrections to the VC Generator for exported parameters where the actual parameter is an array element or a record field, and for explicit modular subtype conversions.

Additional warning control file keywords for default loop assertion and real RTC warnings.

Reduced memory usage in the Examiner.

VC Generation for values of type `Ada.Real_Time.Time` when in RavenSPARK mode.

Improved static semantic checking of the arguments of pragmas.

The Simplifier now reports all uses of user-defined proof rules in the Simplifier Log (SLG) file.

POGS now collates and reports all uses of user-defined proof rules in its summary output.

New “Summary only” option in POGS.

Improved Simplifier tactics for proof of scalar inequalities where transitivity is involved.

The SPARK Team

Praxis High Integrity Systems

[See also “Praxis HIS — SPARK Toolset 7.4” in AUJ 28-1 (Mar 2007), p.16. —su]

## SYSGO — PikeOS for AMCC 440

*AMCC and SYSGO Strengthen Partnership*

PikeOS for AMCC 440 now Available — Joins ELinOS Industrial Grade Linux for AMCC 440 & 405

Mainz/Germany, March 26, 2007

SYSGO AG, the global supplier of highly reliable device software, has released a PikeOS port for the AMCC (Nasdaq: AMCC) 440 PowerPC family. PikeOS is SYSGO’s trend-setting paravirtualization real-time operating system. With the release of the AMCC 440 port, SYSGO now provides support for the entire AMCC 400 product line. The AMCC 400 family is an industry-leading set of processors that’s used widely in high-performance networking and storage applications, among many others.

“We are very pleased that PikeOS is now available for our customers”, said Charlie Ashton, AMCC’s Director of Software. “With its virtualization technology and high level of security, PikeOS offers the user a powerful, modern and more versatile operating system paradigm for real-time applications.”

PikeOS is a powerful real-time operating system that is based on a partition microkernel. It supports heterogeneous software architectures that contain multiple additional operating systems and runtime environments, such as Linux,

POSIX, ARINC 653, Java, VxWorks and Ada. With PikeOS, a relatively uncritical Linux-based part of an application, such as a graphical monitoring system, may run in one partition, with a high-throughput protocol stack running in another partition. With its support for multiple operating systems and run-time environments, PikeOS can support a combination of legacy software and new software, all running on one CPU; it is therefore very effective as an operating system for migration to new environments, while allowing the continued use of older software. PikeOS supports inter-partition communications, but one partition cannot corrupt another.

The PikeOS partitioning system, in conjunction with its internal protection mechanisms, provides the highest available level of safety and security.

“AMCC is a very important supplier of high-performance PowerPC processors for a broad variety of applications. With the availability of PikeOS for the AMCC 440 processor family, users will have an operating system to match the high performance levels of the 440 processors,” said Walter Grossgarten, Partner Manager at SYSGO. “We are happy to offer PikeOS for the AMCC 440 and to work with AMCC and its customers.”

#### About AMCC

AMCC is a global leader in network and embedded PowerPC processing, optical transport and storage solutions. AMCC products enable the development of converged IP-based networks offering high-speed secure data, high-definition video and high-quality voice for carrier, metropolitan, access and enterprise applications. AMCC provides networking equipment vendors with industry-leading network and communications processing, Ethernet, SONET and switch fabric solutions. AMCC is also the leading vendor of high-port count SATA RAID controllers enabling low-cost, high-performance, high-capacity storage. AMCC's corporate headquarters are located in Sunnyvale, California. Sales and engineering offices are located throughout the world. For further information regarding AMCC, please visit the company's web site at <http://www.amcc.com>.

#### About SYSGO AG

SYSGO is specialized in design, implementation and configuration of device software for the embedded market. Besides SYSGO's real-time operating system solution for safety-critical systems, PikeOS and the Embedded Linux development environment, ELinOS, SYSGO offers the development of device drivers, board support packages and firmware. SYSGO also supports its international customers with services for

Embedded Linux, real-time and certification for safety-critical applications. The target markets are Aerospace & Defense, Industrial Automation, Automotive, Consumer Electronic and Network Infrastructure. SYSGO's customers include DaimlerChrysler, EADS Airbus, EADS Military Air Systems, Honeywell, IBM, Raytheon, Rheinmetall, Rockwell-Collins as well as Rohde & Schwarz. SYSGO AG was founded in Mainz, Germany, in 1991 and was reincorporated as a joint stock company in October 2002. Today, the company has six facilities in Germany and Europe and provides a global distribution network.

## Ada and GNU/Linux

### Package Maintainers

*From: Ludovic Brenta <ludovic@ludovic-brenta.org>*

*Date: 22 May 2007 02:48:52*

*Subject: Re: R7 has been released*  
*Newsgroups: comp.lang.ada*

> Have you used the BUSH Business Shell which is built using Ada (95) syntax? It looks interesting, but the couple of times I've tried to install it on one of my Debian machines, there's been a dependency problem or something similar, and I got busy and never got it running. I'm curious to hear if anyone is using it and if it's as good as it looks.

It would seem a natural tool to make use of in an Ada distribution. The BUSH web site / download is at: <http://www.pegasoft.ca/bush.html>

I considered packaging BUSH for Debian but decided not to spread myself too thin (I already maintain ~18 packages, GCC and gnat-gps being the largest two). IIRC, BUSH depends on the Booch Components, which I also decided not to package for lack of time. Of course, if anyone wants to package the Booch Components, BUSH, or any other fine piece of Ada software, I'll be happy to answer questions and sponsor the packages into the official Debian archive.

### Choosing a Linux distribution

*From: Jeffrey Creem*

*<jeff@thecreems.com>*

*Date: Tue, 24 Apr 2007 22:24:52*

*Subject: Re: Which Linux is best on lab top*  
*Newsgroups: comp.lang.ada*

[...] If you care about Ada support under Linux, it sure seems as if Debian is the way to go. While the GNUAda project has many packages available for a few different Unix platforms, I still think the work that is being done in Debian is top notch.

I have been running CentOS 4 for a while and recently started migrating to CentOS 5. CentOS combined with rpmforge is a pretty compelling target in that you get the stability of the Red Hat enterprise packages and still have access to some bleeding edge items.

CentOS includes yum and is very easy to update.

*From: Alex R. Mosteo*

*<alejandro@mosteo.com>*

*Date: Wed, 25 Apr 2007 11:21:21*

*Subject: Re: Which Linux is best on lab top*  
*Newsgroups: comp.lang.ada*

If I'm not mistaken, Debian is the only distro with a dedicated Ada maintainer (thanks Ludovic). And Debian is a great distribution.

You have too the option to choose to be a bit more on the edge and use Ubuntu, since it inherits from Debian. I'm now writing from a Kubuntu laptop while my project compiles in the background. Just a point for your sample.

I have never installed Debian; Ubuntu family is really easy.

I've used Ada equally without problems in Debian (in embedded x86 in robots, where we use Debian stable) and in kubuntu (in my laptop and workstations), although I use mostly the libre site packages and not the ones from the distro.

Perhaps you should check which one supports better your laptop hardware, there may be some particularities.

*From: George Shapovalov*

*<george@gentoo.org>*

*Date: 2 May 2007 00:31:39*

*Subject: Re: Which Linux is best on lab top*  
*Newsgroups: comp.lang.ada*

> If I'm not mistaken, Debian is the only distro with a dedicated Ada maintainer (thanks Ludovic).

Well, you are, Gentoo is the other one. The main difference from Debian is the ability to install multiple Ada compilers in parallel (AdaCores's, FSF's and various "SLOTS" (that is, based on different major backend versions) thereof) and select the active one "on the fly" (similarly to what we have for gcc). The selection of packages differs somewhat, although it is mostly the same common stuff:

<http://packages.gentoo.org/packages/?category=dev-ada>

Of course then there is also an issue of how much control over the installation is desired. That is to say, one should be aware of other Gentoo distinctions if there is any thought of selecting that distribution.

*From: Ludovic Brenta <ludovic@ludovic-brenta.org>*

*Date: Wed, 25 Apr 2007 11:39:02*

*Subject: Re: Which Linux is best on lab top*  
*Newsgroups: comp.lang.ada*

Like others have said, it really boils down to Debian or Ubuntu.

People installing Ubuntu usually download a new 650-meg ISO image and reinstall from scratch every 6 months. Only a few of them know about the better alternative, which is...

People installing Debian usually install only once, and upgrade in place whenever they want. They choose between stable, testing and unstable and track that.

Neither Debian nor Ubuntu have auto-upgrade, because Debian doesn't do things behind people's back. You ask for an upgrade explicitly, when you choose to, and it is very easy because APT tracks dependencies and preserves your configuration files. I have a machine at home where I installed Debian 3.0 "Woody" in late 2002 and never reinstalled since (this machine doesn't boot from CD or network and its diskette drive died since, so I \*couldn't\* reinstall from scratch even if I wanted to). It now runs Debian 4.0 "Etch". The same holds for my two laptops (first installed in March 2004 and November 2006 respectively, and constantly upgraded to track Etch).

The Ada toolchain is not on the Ubuntu CD-ROMs; you have to install from the network.

The Ada toolchain is on the Debian CD-ROMs and DVD-ROMs; you can install either from there or from the network.

I do not receive bug reports filed against Ubuntu, or read the Ubuntu mailing lists or forums. But I do see the Debian bug reports and I am on Debian mailing lists (e.g. [debian-gcc@lists.debian.org](mailto:debian-gcc@lists.debian.org)).

So my advice is: go with Debian, the mother of all distributions. Why use a derivative when you can have the original? Of course you may think I'm biased. I'm not. I chose Debian in 2002 because it was the largest distribution and because it was possible for outsiders like me to contribute. If things changed, I would consider switching again, but so far Debian has kept its promises.

*From: Pascal Obry <pascal@obry.net>  
Date: Wed, 25 Apr 2007 11:47:55  
Subject: Re: Which Linux is best on lab top  
Newsgroups: comp.lang.ada*

> Neither Debian nor Ubuntu have auto-upgrade, because Debian doesn't do

Yes and no. GNOME provides an updatel applet which can be configured:

1. to download and install software automatically
2. download but ask before installing
3. notify when new software versions are available to download

I'm using option 3 on the testing branch. Stable enough (read no problem since 3 years) for what I'm doing.

## Debian GNU/Linux 4.0 released

*From: Ludovic Brenta <ludovic@ludovic-brenta.org>*

*Date: Sun, 08 Apr 2007 21:06:42*

*Subject: Announce: Debian GNU/Linux 4.0 released*

*Newsgroups: comp.lang.ada*

Debian users and developers rejoice!

See the official announcement at <http://www.debian.org/News/2007/20070408>

Like its predecessor, this release is, among many other things, a complete and integrated development and deployment platform for Ada programmers. The platform consists of compiler, debugger, development environments, and many libraries. The deployment platform consists in the many precompiled libraries and apt, Debian's renowned package management system. All packages will work together thanks to adherence to the Debian Policy for Ada (<http://www.ada-france.org/debian/debian-ada-policy.html>)

Here is a summary of what an Ada programmer can find in Debian GNU/Linux 4.0, code-named Etch:

- The GCC 4.1 compiler with support for Ada. The accompanying run-time library is licensed under the GNAT-Modified General Public License (GMGPL). Parts of GNAT are now split out into shared libraries for reusability, and are in fact being reused in the GNAT tools, ASIS, and GPS. The two shared libraries are libgnatvsn (licensed under GMGPL) containing some of GNAT's internals, and libgnatprj (licensed under pure GPL) containing GNAT's project manager (i.e. support for .gpr project files). The package 'gnat' is an empty package that depends on the default compiler. In Etch, the Ada compiler supports more platforms than in the previous version, Sarge: the new platforms supported are alpha, amd64, hppa, ia64, kfreebsd-i386, and s390. Platforms already supported in Sarge were i386, (32-bit) sparc and (32-bit) powerpc.

```
gnat
gnat-4.1
libgnat-4.1
libgnat-4.1-dbg
libgnatprj-dev
libgnatprj4.1
libgnatprj4.1-dbg
libgnatvsn-dev
libgnatvsn4.1
libgnatvsn4.1-dbg
```

- The Ada-aware debugger based on GDB 6.4:

```
gnat-gdb
```

- The GNAT Programming Studio, version 4:

```
gnat-gps
gnat-gps-doc
```

- The GNAT Library for Ada Distributed Execution (GLADE), the free implementation of Annex E:

```
gnat-glade
gnat-glade-doc
libgarlic-dev
libgarlic2006
```

(gnat-glade-doc is the only package not updated since Sarge, for lack of time and energy on my part, but the older version of the doc is still usable and relevant to GNADE 2006).

- The Ada Semantic Interface Specification (ASIS):

```
asis-programs
libasis-dev
libasis2005
```

- Additional development tools based on ASIS:

```
adabrowse
adacontrol
```

- GtkAda, version 2.8.1 (used in GPS):

```
libgnomeada-2.8
libgnomeada2-dbg
libgnomeada2-dev
libgtkada-2.8
libgtkada-gi-2.8
libgtkada-glade-2.8
libgtkada2-bin
libgtkada2-dbg
libgtkada2-dev
libgtkada2-doc
```

- The Ada Web Server, version 2.2:

```
libaws-bin
libaws-dev
libaws-doc
libaws2.2
```

- XML/Ada, version 2.2:

```
libxmlada2
libxmlada2-dev
```

- The GNU Ada Database Environment, version 1.6:

```
gnade-bin
libgnademysql-dev
libgnademysql1.6
libgnadeodbc-dev
libgnadeodbc1.6
libgnadepostgresql-dev
libgnadepostgresql1.6
libgnadesqlite-dev
libgnadesqlite1.6
```

- Several assorted libraries to complement the above:

```
adacgi
libadsockets-dev
libadsockets1
libaunit-dev
libaunit1.03
libflorist-dev
libflorist2006
libopentoken-dev
```

libopentoken3.0b  
 libtemplates-parser-dev  
 libtemplates-parser10  
 libtexttools-dev  
 libtexttools2.0

- And one end-user application written in Ada; I wish that future versions of Debian contain many more!

music123

## Plans for next Debian release

*From: Ludovic Brenta <ludovic@ludovic-brenta.org>*

*Date: Thu, 26 Apr 2007 23:51:41*

*Subject: Re: Which Linux is best on lab top*  
*Newsgroups: comp.lang.ada*

[...] We're looking at a release of Debian "Lenny" in October 2008. That will be my overriding goal. In the mean time, I expect GCC 4.2 to be released and a large-scale transition to take place. Then, GNAT GPL 2007 will probably cause another transition. If GCC 4.3 releases reasonably soon before the compiler freeze in mid-2008, there will be a third transition, and possibly a fourth transition if GNAT GPL 2008 releases soon enough.

During each transition, Ada in Debian unstable deserves its name: UNSTABLE, in big neon letters. In the Lenny time frame, Ubuntu will probably want to release 7.10, 8.04 and 8.10. Each time, it runs the risk of taking a snapshot of Debian unstable at an unfortunate point in time, and to ship broken Ada packages.

So far, one person has volunteered to monitor the Ada packages in Ubuntu and see to it that they are not too broken. This person is Matthias Klose, the principal maintainer of GCC, Python and other things in both Debian and Ubuntu. If you look at the lists of packages he maintains, you'll understand he may not be able to give Ada all the love and care she deserves. Of course, he keeps in touch with me to make Ubuntu releases relatively stable, but with so many transitions on the horizon, he might just not be able to.

If you want a stable and supported distribution, use Debian stable.

---

## Ada and Macintosh

### Ada Mac OS X FAQ

*From: Jerry <lanceboyle@qwest.net>*

*Date: 26 Apr 2007 20:46:37*

*Subject: New FAQ for Macintosh (OS X)*  
*GNAT users*

*Newsgroups: comp.lang.ada*

There is a new FAQ up for GNAT users on Macintosh OS X. It addresses several common issues that new users ask so I

hope it helps those who posted recently about GNAT on OS X.

[www.macada.org/macada/FAQ.html](http://www.macada.org/macada/FAQ.html)

---

## References to Publications

### AdaCore — Ada Gems series

*From: Jamie Ayre <ayre@adacore.com>*

*Date: 14 May 2007 08:53:56*

*Subject: Announcing the launch of the Ada*  
*Gems series*

*Newsgroups: comp.lang.ada*

AdaCore is pleased to announce the launch of the new Ada Gems series. The Gems will cover a wide-range of Ada programming topics and will often include executable source code and other related resources to help understand and discover the Ada programming language. The first Gem, written by Bob Duff entitled "Limited Aggregates in Ada 2005", is available at:

[www.adacore.com/home/ada\\_answers/gems](http://www.adacore.com/home/ada_answers/gems)

### AdaCore — Dr. Dobb's

*From: AdaCore Press Center*

*Date: Thursday May 10, 2007*

*Subject: Safety AND Security*

*RSS: [www.adacore.com/category/press-center/feed/](http://www.adacore.com/category/press-center/feed/)*

Dr. Dobb's

"Safety AND Security" by Robert BK Dewar

Historically, safety and security have been rather sharply separated

<http://www.ddj.com/dept/security/199300140>

### Praxis — Military Embedded Systems magazine

Military Embedded Systems magazine features SPARK in "Building secure software: Your language matters!" co-authored by SPARK team's Roderick Chapman and AdaCore's Robert Dewar.

<http://www.mil-embedded.com/articles/id/?2012>

### AdaCore — Abstract Interface Types in GNAT

*From: AdaCore Developer Center*

*Date: Wednesday March 21, 2007*

*Subject: New technical paper on Abstract*  
*Interface Types in GNAT*

*RSS:*

[www.adacore.com/category/developers-center/development-log/feed/](http://www.adacore.com/category/developers-center/development-log/feed/)

Ada 2005 Abstract Interface Types provide a limited and practical form of multiple inheritance of specifications. In this paper we cover the following aspects of their implementation in the GNAT compiler: interface type conversions, the layout of variable sized tagged objects with interface progenitors, and the use of the GNAT compiler for interfacing with C++ classes with compatible inheritance trees.

Authors: Javier Miranda and Edmond Schonberg

<http://www.adacore.com/2007/03/21/new-technical-paper-on-abstract-interface-types-in-gnat/>

### AdaCore — Wind River EMEA Aerospace and Defence Seminars

*From: AdaCore Press Center*

*Date: Monday May 14, 2007*

*Subject: Wind River EMEA Aerospace and*  
*Defence Seminars*

*RSS: [www.adacore.com/category/press-center/feed/](http://www.adacore.com/category/press-center/feed/)*

AdaCore will be exhibiting at the upcoming Wind River Aerospace and Defence events in Europe:

- June 5, Rome Italy
- June 7, Madrid Spain
- June 11, Manchester UK
- June 12, Farnborough UK
- June 14, Paris France

For more information [see <http://www.windriver.com/announces/aerospacedefenseforum-2007/> —su]

---

## Ada Inside

### DDC-I — KC-135 jet transport

*DDC-I Helps Northrop Grumman Migrate KC-135 Ada Tools from VAX to PC Host Environment*

PC Host Environment Reduces Maintenance Cost

Phoenix, AZ. March 12, 2007. DDC-I, a leading supplier of development tools for safety-critical applications, today announced that it has been selected by Northrop Grumman to help migrate the KC-135's Ada software development environment from a VAX/VMS to a PC/Windows XP host. The change in host environment will greatly reduce the cost associated with providing fielded support for the KC-135's Ada code.

"The TADS-i960 provides a very stable i960 Ada development environment for the KC-135," said Bob Morris, president and CEO of DDC-I. "But maintaining Ada code on aging and obsolete VAX systems becomes more expensive every year. With the KC-135 expected to

continue flying for at least another decade, a change in host systems was inevitable. After we completed a migration risk assessment, Northrop Grumman decided that it would be more cost effective to move to a Windows XP environment.”

The KC-135 is a swept-wing, high-altitude, high-speed jet transport whose primary mission is refueling strategic long-range bombers. The KC-135 can haul 83,000 pounds of cargo, airlift up to 80 passengers, or carry 202,800 pounds of JP-4 jet fuel, most of which is transferable for global refueling missions.

Northrop Grumman currently uses DDC-I's TADS-i960 Ada development environment to develop and maintain code for the KC-135's Fuel Management Advisory Computer Program (FMAC). The FMAC gauges fuel level, both inside and outside (for refueling) the aircraft. It also provides performance calculations for takeoffs, landings (such as center of gravity and optimal landing/takeoff speed) and navigation using a variety of data, such as environmental and runway conditions.

The FMAC is implemented using a custom module equipped with redundant i960 processors. The FMAC Ada application code that runs on each processor is deployed on a bare board Ada run-time system provided by DDC-I. The application code was developed using a TADS-i960 development environment and Ada compiler hosted on a VAX/VMS system.

Northrop Grumman, upon inheriting the original FMAC code and development tools, decided that they could simplify maintenance and reduce ongoing maintenance costs by moving the tools and code to a PC development host running Windows XP. Complicating this migration, however, was the fact that the code and set up were not well documented. So, at Northrop Grumman's request, DDC-I performed an on-site assessment of the Ada software and libraries. Ultimately, DDC-I was able to unravel the mystery behind many of the libraries, including their origin and how to re-build them.

With DDC-I's assistance, Northrop Grumman now has the capability to completely rebuild its libraries from scratch. DDC-I also helped Northrop Grumman complete an initial build of its Ada and assembler files and resolve many other issues that would have otherwise taken much longer to resolve. Ultimately, with DDC-I's assistance, Northrop Grumman was able to successfully port the software to the new development platform with minimal problems. The port went much smoother than originally anticipated due to DDC-I's support.

More on the TADS-i960

The TADS-i960 provides a comprehensive environment for developing, compiling and debugging Ada applications for the i960 processor. The TADS-i960 compiler is optimized for the i960 processor, leveraging special-purpose i960 instructions such as branch-and-link and instruction scheduling. The compiler also provides Ada-specific optimizations such as constraint and overflow check elimination, parameter binding, data packing, and static aggregates initialization.

The TADS-i960 debugger provides point-and-click, multi-window symbolic debugging at both the Ada source and assembly language level. The debugger displays key program information such as command execution, source code, program I/O, scoreboarding of automatically updated data, disassembled machine code, interspersed source/assembly code, and on-line help. It also features full tasking and exception support, and dynamic code patching for executing programs.

The TADSi960 run-time system provides real-time multitasking with priority-driven, pre-emptive scheduling. The run-time system also provides a rendezvous accelerator and supports multiple options for handling interrupts, including attachment to task entries, Ada procedures, and assembly routines.

About DDC-I, Inc.

DDC-I, Inc. is a global supplier of software development tools, custom software development services, and legacy software system modernization solutions, with a primary focus on safety-critical applications. DDC-I's customer base is an impressive “who's who” in the commercial, military, aerospace, and safety-critical industries. DDC-I offers compilers, integrated development environments and run-time systems for C, Embedded C++, Ada, JOVIAL and FORTRAN application development.

## Praxis HIS — iFACTS Air Traffic Control

07 March 2007

*Praxis appointed to work on the biggest change in air traffic control systems since the introduction of radar*

Praxis, a leading specialist in critical systems engineering has been appointed by National Air Traffic Services (NATS) to work on a £10 million Air Traffic Control (ATC) partnering contract. Praxis will write the specification and develop the software for a new ATC System called iFACTS which will trigger the biggest change in ATC since the introduction of radar.

The UK government predicts a doubling of air passenger numbers by 2020 and a trebling by 2030 so the introduction of

iFACTS is vital. This new technology, installed at the London Area Control Centre, Swanwick, will bring significant safety and capacity benefits.

iFACTS — Interim Future Area Control Tools Support — will not only provide Air Traffic Controllers with a set of advanced tools to increase capacity to meet the growing demand from the civil aviation industry, it will also alert Controllers to flights which are not following their flight plan and detect medium term conflicts, which will also enhance safety capability.

Praxis has written the specification and developed the new software for iFACTS. The company's work will build on the NATS-led design, and include the use of advanced software engineering. The contract includes training NATS staff in techniques from Correctness by Construction, Praxis' approach for developing very low defect software in critical applications.

Keith Williams, Praxis Managing Director, said “It is extremely exciting to be able to deploy our capability in critical software on the iFACTS project. Praxis is working with NATS at all levels to meet the challenges of the project and ensure its success.”

NATS has pioneered research and development of advanced air traffic control tools for several years from its simulator and research centre at Hurn. The iFACTS project will deliver a subset of these tools onto the system at the company's main en-route Control Centre at Swanwick in Hampshire.

The system monitors radar for the Controllers, and assesses the viability of various options available to them for maneuvering aircraft, as well as giving them more time to make decisions. Traditional paper flight information strips will be replaced with electronic data lines and more sophisticated split-screen displays. The system has been designed for robustness, and has built-in contingency.

Final trials of the iFACTS tools are currently under way at Hurn and a demonstration system is already installed in Swanwick's training unit for controllers to try out. Following full development, training, and installation of new workstations at Swanwick, iFACTS will be introduced into service.

Paul Barron, NATS' Chief Executive commented: “This is one of the most exciting developments in the aviation industry in decades and we're now close to introducing it.”

## MBDA Signs Agreement with AdaCore

Tuesday April 3, 2007

*MBDA Signs Global Software Licensing Agreement with AdaCore*

Leading missile company adopts AdaCore's GNAT Pro development environment as corporate standard

PARIS and SAN JOSE, Calif. — April 3, 2007 — Embedded Systems Conference — AdaCore today announced a global licensing agreement with world-leading missile systems company MBDA. The three-year agreement builds on MBDA's existing long-term commitment to the Ada programming language by adopting AdaCore's GNAT Pro development environment as its corporate Ada standard. The initial agreement covers 80 developers, and enables MBDA to add additional licences as needed to flexibly and cost-effectively meet its changing needs.

A key reason why MBDA chose AdaCore and GNAT Pro was the wide variety of platforms it supports, including native hosts, such as Microsoft Windows, Sun Solaris and GNU/Linux. GNAT Pro also supports embedded targets for bare board PowerPC and all WindRiver VxWorks platforms. The license covers Ada, C and add-on requirements for eight native and cross platforms. The agreement also allows MBDA to benefit from major cost efficiencies in terms of training, code portability and support.

MBDA is already using AdaCore products on a number of its high reliability and mission-critical projects. These include the Meteor Beyond Visual Range Air-to-Air Missile (BVRAAM), JERNAS ground-based defense system and the Principal Anti-Air Missile System (PAAMS). The Ada language enables MBDA to create maintainable, safety-related code for its products.

"Ada is becoming the de facto standard for developers of reliable, efficient and high integrity safety-related applications," said Franco Gasperoni, Managing Director, AdaCore. "This global contract demonstrates our leadership in the market and provides MBDA with flexible access to our software and support, enabling the company to leverage its investment in Ada across the widest range of development platforms."

About Ada / GNAT Pro

Ada is a modern, internationally standardized language that has proven its value for developing large, long-lived applications where safety and reliability are essential. It has consequently become a leading language in commercial and military avionics, air traffic control, high-speed rail systems and nuclear power systems.

AdaCore has been closely involved with the Ada language since its inception. Its GNAT Pro development environment combines market-leading technology, including Ada 2005, the latest version of

the language, with an expert support system to provide a natural solution for organizations creating efficient and reliable code.

At the heart of GNAT Pro is a full-featured, multi-language (Ada, C, C++) development environment complete with libraries, bindings and a range of supplementary tools. All its technology combines the flexibility and freedom associated with open source development and the assurance that comes from knowing that all tools go through a rigorous quality assurance process. It is based on the widely used GCC technology and is backed by rapid and expert support service.

About MBDA

With an annual turnover exceeding €3 billion, a forward order book of over €13 billion and over 70 customers around the world, MBDA is one of the world's leading missile systems companies. MBDA currently has 45 operational programs for missile systems and countermeasures, and has proven its leadership in the management of large multinational projects. MBDA is jointly owned by BAE SYSTEMS (37.5%), EADS (37.5%) and FINMECCANICA (25%).

About AdaCore

Founded in 1994, AdaCore is the leading provider of commercial software solutions for Ada, a modern programming language designed for large, long-lived applications where reliability, efficiency and safety are critical. AdaCore's flagship product is GNAT Pro, which comes with expert online support and is available on more platforms than any other Ada technology. AdaCore has customers worldwide; see <http://www.adacore.com/home/company/customers/> for more information.

Use of Ada and GNAT Pro continues to grow, especially in high-integrity and safety-critical applications, including commercial and defense aircraft avionics, air traffic control, railroad systems, financial services and medical devices. AdaCore has North American headquarters in New York and European headquarters in Paris. [www.adacore.com](http://www.adacore.com)

## Green Hills Software Selected by Saab Bofors Dynamics

Date: April 11, 2007

URL: [http://www.ghs.com/news/20070411\\_saab.html](http://www.ghs.com/news/20070411_saab.html)

Green Hills Software Selected by Saab Bofors Dynamics to Supply High-Reliability Solution for Advanced Missile System

INTEGRITY RTOS and MULTI Development Tools to Meet Rigorous Performance Requirements

Green Hills Software and Curtiss Wright SANTA BARBARA, CA — April 11, 2007

Green Hills Software, Inc., the technology leader in device software optimization (DSO) and real-time operating systems (RTOS), announced today that Saab Bofors Dynamics AB has selected the Green Hills Software INTEGRITY RTOS, networking middleware and AdaMULTI integrated development environment (IDE) for the next phase of its RBS15 project. The RBS15 is a next-generation fire-and-forget, multi-purpose surface-to-surface missile system.

"Reliability was crucial to the Saab Bofors Dynamics team and a major reason for choosing solutions from Green Hills Software," according to Pontus Björklund, manager Software Architecture and Platform at Saab. "Saab looks for rigorous conformance with all applicable certification schemes, as well as a firm commitment to long-term cooperation to support Pre-Planned Product Improvement (P3I) throughout the lifetime of its projects, including RBS15. Green Hills Software was uniquely able to meet Saab's requirements, both in terms of development tools and the RTOS for its chosen hardware platform."

Green Hills Software is also providing protocols for high-reliability networking to support control and monitoring of RBS15 units in the field. These form part of a complete development solution from Green Hills Software, providing project benefits, including lower costs and unified technical support, while also maximizing product performance and conformity with defence procurement requirements internationally.

"Green Hills has been able to make a valuable contribution to the RBS15 project, working closely with the team at Saab Bofors Dynamics," said Dan O'Dowd, founder and chief executive officer at Green Hills Software. "We are partnering in crucial aspects that simply must be right in order for defence contractors to deliver extremely high performance and reliability with the low life-cycle costs demanded by modern military procurement bodies."

The RBS15 project has unique requirements within the different computer systems. The combination of the INTEGRITY RTOS with an Ada 95 language-based AdaMULTI IDE for Bare Board provides a complete suite of software development tools for mission-critical embedded systems.

The Green Hills INTEGRITY RTOS is a true real-time operating system that carries reliability, availability, and serviceability credentials. It has been certified multiple times to the FAA's safety critical standard DO-178B Level A for flight-critical avionics and being used



in FDA/CDRH Class II and III life-critical medical devices. INTEGRITY uses hardware memory protection to maximize security without sacrificing real-time performance, and is designed to help embedded designers guarantee resource availability.

#### About Saab Bofors Dynamics

Saab Bofors Dynamics supplies high-performance, cost-effective, reliable defence systems that cover all levels of combat and can be used in all conditions from the cold of the Arctic to the arid heat of the desert and the humidity of the jungle. The majority of Saab Bofors Dynamics business is concentrated on research and technology. Operations are divided into the two primary business areas of Missiles and Support Weapons with worldwide well-known products like the anti-ship missile RBS 15, the air-defence missile RBS 70 and the multi-role weapon Carl-Gustaf. Saab Bofors Dynamics is headquartered in Karlskoga, Sweden.

Visit Saab Bofors Dynamics on the web at [www.saabgroup.com](http://www.saabgroup.com)

#### About Green Hills Software

Founded in 1982, Green Hills Software, Inc. is the technology leader in device software optimization (DSO) and real-time operating systems (RTOS) for 32- and 64-bit embedded systems. Our royalty-free INTEGRITY® and veOSity™ real-time operating systems, μ-veOSity™ microkernel, compilers, MULTI® and AdaMULTI™ integrated development environments and TimeMachine™ tool suite offer a complete development solution that addresses both deeply embedded and high-reliability applications. Green Hills Software is headquartered in Santa Barbara, CA, with European headquarters in the United Kingdom. Visit Green Hills Software on the web at [www.ghs.com](http://www.ghs.com).

Green Hills, the Green Hills logo, MULTI, INTEGRITY, μ-veOSity, veOSity, AdaMULTI, SuperTrace, TraceEdge and TimeMachine, are trademarks or registered trademarks of Green Hills Software, Inc. in the U.S. and/or internationally. All other trademarks are the property of their respective owners.

### National Ignition Facility laser

*From: John Woodruff  
<JPWoodruff@gmail.com>*

*Date: 18 May 2007 13:11:23*

*Subject: Re: is it safe do adopt CORBA?  
Newsgroups: comp.lang.ada*

> I'm diving into the development of a generic framework for build business applications in Ada.

I'd like to tell you about the Integrated Computer Control System that controls the National Ignition Facility laser at Lawrence Livermore National Lab. The laser, which is nearing completion, is used for research on inertial confinement nuclear fusion. This multi-billion dollar facility will be used for several decades.

The control system software is written in Ada and Java, and is distributed over some 600 computers using CORBA. The ICCS team developed a software framework of tools and patterns for building the large number of FEPs and supervisory systems. The system meets its requirements.

There are several published accounts of the software system, including my own papers in SigAda 1998 and 2002. I worked on the ICCS software framework from its inception until I retired 5 years ago. More recent reports have appeared in ICALEPCS conferences starting in 2001.

---

## Ada in Context

### About the ISO Ada Amendment 1

*From: Jeffrey D. Cherry*

*<jfej.yrrehc@stepbfav.moc>*

*Date: Mon, 02 Apr 2007 15:21:41 GMT*

*Subject: New Ada Standard*

*Newsgroups: comp.lang.ada*

I see that ISO and ANSI have a new listing for Ada, specifically:

ISO/IEC 8652/Amd1:2007  
Information technology — Programming languages — Ada — Amendment 1

I haven't purchased this document, yet. I just wanted to confirm that this is the new Ada 2005 standard that we've been waiting for. If it is, then should I be calling this the Ada 2007 standard? Also, will there be a reduced price version in the near future? (It is currently listed as US \$238 from ANSI for a 1.44MB PDF file.)

*From: Randy Brukardt*

*<randy@rrsoftware.com>*

*Date: Mon, 2 Apr 2007 19:01:31*

*Subject: Re: New Ada Standard*

*Newsgroups: comp.lang.ada*

> I believe you can find the same text at <http://www.adaic.org/standards/ada05.html> for somewhat less than \$238.

It should be noted that the ISO version is the Amendment itself, which is a list of changes to the Ada 95 standard. It's not very easy to use. It makes much more sense to use one of the consolidated versions; you can get those from the above site; and you can get printed versions from Springer.

*From: Stephen Leake*

*<stephen\_leake@stephe-leake.org>*

*Date: Tue, 03 Apr 2007 06:14:16*

*Subject: Re: New Ada Standard*

*Newsgroups: comp.lang.ada*

The ISO working group on Ada agreed that the common vernacular for the 2007 version should be Ada 2005. See the post in this newsgroup recently [...]

The rationale for this choice is that a lot of people have been calling it Ada 2005 for a while, and there's no real harm in continuing. The GNAT compiler has an option `-ada05`, for example, which is not going to change to `-ada07`.

*From: Randy Brukardt*

*<randy@rrsoftware.com>*

*Date: Tue, 3 Apr 2007 18:37:20*

*Subject: Re: New Ada Standard*

*Newsgroups: comp.lang.ada*

This is not an accurate summary of the WG 9 action. The WG 9 vote was intended to apply to the name during the development period (that is, up to standardization). After that, of course, the name of the language is Ada.

The only rationale was that we were hopelessly deadlocked on the issue. But there had to be some resolution (in that \*something\* had to go into the documents), and the people who were against using "Ada 2005" did not feel as strongly as some of those who wanted to stick with it. And in any case, the decision made there is not binding on anyone for purposes other than official WG 9 purposes (and even there, it should be avoided in favor of more precise identification of standards).

There are a lot of good arguments for changing the name now as was done with Ada 95 (especially that older articles which reference obsolete drafts of the language would not confuse readers with misinformation); there are also good arguments for staying the course (it doesn't make sense to dilute Ada's marketing).

In any case, the best choice is to call it "Ada"; it's now the standard and should be the default on new compilers (it certainly will be on Janus/Ada). If you need to reference the standard itself, that is "Amendment 1" (there is no standard document that includes the complete language; you have to merge three of them to get that, or use Ada Europe's consolidated standard — which isn't official anyway). You might need year numbers to talk about obsolete Ada versions like Ada 95, but not about the current one.

*From: Randy Brukardt*

*<randy@rrsoftware.com>*

*Date: Wed, 4 Apr 2007 17:44:15*

*Subject: Re: New Ada Standard*

*Newsgroups: comp.lang.ada*

> To me, the term "Ada" could refer to any version of the language. It is sometimes necessary to make it clear

that you are referring to the most recent version of the language, and a term such as “Ada 2005” is a convenient way of doing so.

True enough, but you can only use that informally...

> For example, if someone wanted to inquire whether a certain compiler supported all of the latest language features, he might ask whether that compiler supported the full Ada 2005 standard.

... and by using the word “standard”, you’ve left the realm of informal. Indeed, there is no such thing as “the full Ada 2005 standard”, and there never will be (even ignoring the year number issue).

The current Ada standard is made up of three ISO/IEC published documents: The International Standard “Programming Languages — Ada” (ISO/IEC 8652:1995); Technical Corrigendum 1 (ISO/IEC 8652:1995/COR.1:2001); and Amendment 1 (ISO/IEC 8652:1995/AMD.1:2007). Note that only the first is even called “International Standard”. The others aren’t “Standards” (even though they’re standardized, and go through similar approval processes as a standard). There is no single document that you can reference, two of them aren’t Standards (although it’s probably OK to reference Amendment 1 as a “standard”, note the capitalization difference), and none of them are published in 2005.

So it’s OK to reference “full Ada 2005” as an informal reference to a language; reference the Ada Europe consolidated reference manual (its cover says “Ada 2005 Reference Manual”); [but that is most definitely not a standard!], or reference the specific documents by their publication dates (“the Ada standards as of 2007” or “the 2007 Amendment to Ada”), but it isn’t OK to reference “standard” (singular) for Ada anymore, nor is it OK to reference the standards documents by years other than those they are published in. The possibility of confusion between “standard” and “Standard” is so severe that think it is best to avoid the word altogether.

Admittedly, Ada users have a long history of being confused about what is a standard and what is not (the RM you have on your desk or in your computer is almost certainly not a standard, no matter which version of Ada you are using, it just contains text fairly similar to that in the Standard [and other documents]).

If it was up to me, I call it Ada 2007, as that doesn’t lead to a confusion between the standards and the other documents. But even that isn’t ideal for a number of reasons. But we’re now in a situation that any name that you use will be either misleading or wrong in some contexts or will be loudly criticized by various self-appointed critics. I don’t want to go there,

so I don’t put any year number behind Ada unless I’m specifically talking about old, obsolete versions.

In any case, “Ada” officially means the most recent set of standards; older standards are “canceled and replaced”. Using “Ada” when you mean only “Ada 83” is just plain wrong and is far too common (especially amongst those who only used Ada in the 1980’s). I understand the ambiguity problem, but I prefer an combination of education (to get writers out of their bad habits) and direct references to the actual Amendment document when absolutely necessary.

[See also “Ada 2005 Published by ISO” in AUJ 25-2 (Jun 2004), p.43 and “Ada 2005 Becomes Official ISO Standard” in this issue. —su]

## Optimization of Floating-Point code

*From: David Smith*

*<david.smith@gmail.com>*

*Date: 16 May 2007 15:37:08*

*Subject: GNAT Optimization of Constant Expressions*

*Newsgroups: comp.lang.ada*

In running some benchmarks between Ada, C, and Fortran, I came across a bizarre situation with GNAT. The same algorithm ran 10x slower in Ada, so I surmised that the constant math expressions in the code below were being precomputed by the compiler in C and Fortran, but not in Ada. To test my hypothesis, I defined the commented-out constants and replaced the math functions with the appropriate constant. This worked, and the Ada code was then as fast as the others. The bizarreness came out when I put the math expressions back in but left in the constant declarations. The code was still fast! I’m not even using the constants, but somehow they are helping the compiler optimize the code.

In short, when I un-comment the constants, the code is super fast, but when commented out, the code is slow, even though I never use them.

Does anyone know why this could be? [...]

*From: Randy Brukardt*

*<randy@rrsoftware.com>*

*Date: Wed, 16 May 2007 23:50:01*

*Subject: Re: GNAT Optimization of Constant Expressions*

*Newsgroups: comp.lang.ada*

It doesn’t surprise me in the least; I’d expect Janus/Ada to work similarly, in fact.

First of all, Sin and the like are regular functions in Ada; they’re never built in. I suspect that is different in your typical Fortran compiler. The only thing that the compiler is likely to know about them in Ada is that they are Pure, which means

that the compiler can assume that two calls with the same arguments return the same result.

As such, the compiler has to do a loop-hoisting optimization in order to pre-evaluate these function calls. I suspect that from your results, Gnat doesn’t do such an optimization in this case. That doesn’t surprise me, a loop hoist can make the program run slower if the loop is never executed. I don’t think we do them at all. I would expect that Gnat does in some circumstances, but for some reason it doesn’t notice that the loops are going to execute far more than zero times.

Anyway, if you explicitly declare the constants, the situation changes. First of all, these sorts of constants will most likely be implemented much like initialized variables (the value being stored in memory). That’s one probably one of the reasons why Gnat doesn’t do the optimization (extra memory use). But if you explicitly declare them, there no longer is any extra memory.

Moreover, doing so changes the optimization from loop hoisting (which is done in limited circumstances) to common subexpression elimination (which is done whenever possible). [I’m speaking about these optimizations in Janus/Ada here, I have no knowledge of how Gnat/GCC divide up these optimizations; but I’d expect it to be similar.] Since these are Pure function calls, the compiler can replace the later uses by your explicitly declared constants, and thus get your observed speedup.

Without the constants, the compiler would have to effectively declare them itself, and there are a number of reasons that the compiler might not choose to do that (as previously mentioned). In any case, the best rule for a compiler optimizer is to implement what was written if you cannot be sure that the optimization will be faster/smaller. So I’m not surprised.

After all, your original code is written to make a bunch of very expensive calls (“Sin(0.5)”, for instance) many times. Expecting the compiler to remove those calls is asking a lot; it is much better to avoid making extra calls with appropriate constant declarations. (I’d have that advice for any language; why force the optimizer to work hard and possibly fail??)

*From: Randy Brukardt*

*<randy@rrsoftware.com>*

*Date: Thu, 17 May 2007 15:46:23*

*Subject: Re: GNAT Optimization of Constant Expressions*

*Newsgroups: comp.lang.ada*

> I agree for any function call, but in the special case of functions from `Generic_Elementary_Functions`, it is `_not_` asking a lot! A decent compiler should perform optimizations around these functions as well as around arithmetic operators.

It is reasonable for the fact that these are pure functions to be detected and used, but that is about it. (And that has nothing special to do with these functions, it is true for just about all pure functions.)

Keep in mind that floating point optimization is very hard; most of the typical optimizations that you do on integer types would change the result of a floating point expression. And, if those are carefully crafted numerical expressions, the “optimization” could actually cause the code to fail.

One example is:

```
Y := (X - 0.5) - 0.5;
```

You might think that replacing this with

```
Y := X - 1.0;
```

would be a good idea, but actually it reduces the accuracy of the result — which is probably the reason that the programmer wrote the above in the first place. Similarly, changing

```
X + (Y + Z) into (X + Y) + Z
```

can change the accuracy a lot.

The net effect is, with a truly good compiler, you have to write the floating point expression that you mean and expect limited optimizations. Either that or decide you don't care about accuracy (which is how C compilers tend to handle this). I know that Gnat worries about these issues a lot (I've heard Robert Dewar talk about them on many occasions).

There are optimizations that you can do safely, like common subexpression evaluation, but there are many fewer of them than you have with integer optimizations. Luckily, most of the code in a program (any program, once you take address calculations like array indexing into account) is integer code. It is very important to do a good job on integer expressions, much less valuable of float ones (because of the limited things that you can do).

> Obviously, there are compilers where the optimizer is working hard for the programmer, and other ones where the programmer has to work hard to make the optimizer work!

But that wasn't the point. The point was that a careful compiler *must* not do much optimization on float expressions, because doing so would change the results and break carefully crafted numeric code. So it is necessary to *write* what you mean, since the compiler can only help you if it is sloppy (and thus will cause trouble later).

*From: Randy Brukardt  
<randy@rrsoftware.com>  
Date: Thu, 17 May 2007 18:38:34  
Subject: Re: GNAT Optimization of  
Constant Expressions  
Newsgroups: comp.lang.ada*

> The purpose in leaving them in was to test this aspect of the optimization process. This is a very common expression to see in legacy scientific codes written by novices, and I wanted to know if GNAT can handle naive structures as well as Fortran compilers do. If it can't, then many scientists and potential Ada converts will be turned off because they'll write a naive bit of code for a benchmark and find that Ada is much slower than Fortran without trying to understand why.

Fair enough, but this seems to be the wrong question to ask. The question is whether it is fast enough for the application, not whether it is as fast as some other language that doesn't have the same requirements. Anything else is premature optimization. And, in addition, can the code be made fast if it actually matters. (I have serious doubts about the value of “naive” numerical code, but I'll leave that rant for some other time...)

> I understand that the FP ops are not associative, but doesn't the gcc -O3 imply that the user chooses speed over accuracy? And especially if the user passes -ffast-math the compiler should get the hint.

I don't agree. Ada has fairly strict requirements on accuracy that C and Fortran don't have. An optimizer shouldn't be providing the *wrong* answer in order to make code fast; so I doubt that -O3 would have that effect on an Ada program (where it might for C).. I realize that Ada 95 made those accuracy requirements optional, but I think that most Ada compilers (which have Ada 83 roots) make the strict mode their default. And since the strict mode necessarily incorporates any relaxed mode, it's quite likely that they never deviate from it. I don't know Gnat well enough to comment on whether it has a separate relaxed mode.

> [Floating Point operations] are very expensive, so trimming those can potentially save more than you might think. In this example, the common expression elimination speeds the code up by a factor of 10, roughly.

No, FP isn't very expensive. “Elementary function” library calls that may execute hundreds of lines of code are expensive! (Keep in mind that Ada's accuracy rules prevent Ada compilers from using the hardware support for Sin/Cos/Tan — it's not accurate enough.) You can write a lot of interesting FP code (statistics, for instance) without using any elementary functions other than Sqrt. And the use of constants as arguments to those expensive routines is pretty silly. (Besides, good Ada practice would be to name your constants, and when you did that, you got essentially the same code as with Fortran.)

None of which is to say that common subs shouldn't be handled, but there is a big difference between hoisting expressions and simply avoiding duplicate evaluations. The former can make code larger and slower if done inappropriately. It may simply not be worth doing, if it doesn't help most programs. Of course, you can always write a program that it does help (like yours), but one has to allocate ones resources based on the value to all customers and all programs, not just one.

(And please remember I know nothing at all about Gnat's actual optimization technology. This might just be a bug in something that is supposed to work.)

*From: Georg Bauhaus  
<bauhaus@futureapps.de>  
Date: Fri, 18 May 2007 13:27:45  
Subject: Re: GNAT Optimization of  
Constant Expressions  
Newsgroups: comp.lang.ada*

> Ada has fairly strict requirements on accuracy that C and Fortran don't have. An optimizer shouldn't be providing the *wrong* answer in order to make code fast;

An Apple employee once recommended using the then new FPT optimizations and to see whether the optimized program will meet the expectations. (“You can always sell speed”.) So maybe accurate computation is a far reaching requirement for some software markets.

*From: Randy Brukardt  
<randy@rrsoftware.com>  
Date: Fri, 18 May 2007 12:28:04  
Subject: Re: GNAT Optimization of  
Constant Expressions  
Newsgroups: comp.lang.ada*

I think it is generally agreed these days that the Ada accuracy requirements were a mistake, because they put Ada at a competitive disadvantage compared to other languages for floating point math. Indeed, that is why they were moved into “strict mode” in Ada 95. The problem is, faced with the need to implement strict mode anyway, implementers have tended to avoid implementing relaxed modes, because extra modes simply doubles the implementation and testing effort — the fewer modes the better. And the argument is “who wants less accuracy anyway?”.

We talked briefly about dropping the accuracy requirements altogether this time, but its obvious that would screw anyone that was depending on them. And that didn't seem like a good idea. So Ada is rather stuck with its slower but predictably accurate math. (In practice, this really only matters in programs with intensive FP, like the OP's example.)

*From: Duncan Sands <baldrick@free.fr>  
Date: Fri, 18 May 2007 19:08:54  
Subject: Re: GNAT Optimization of  
Constant Expressions*

> Can you please post your C equivalent. As expected, the `sin` etc function calls have been replaced with explicit constants before the real GCC optimizers even run! “Fold” is responsible. The difference with Ada is that the Ada `sin` etc functions are not recognized by fold.

*From: Duncan Sands <baldrick@free.fr>  
Date: Fri, 18 May 2007 19:32:57  
Subject: Re: GNAT Optimization of Constant Expressions  
Newsgroups: comp.lang.ada*

PS: If you disable fold, then the function calls are hoisted out of the loop if you mark `sin` etc with attribute “const”, but not if you mark them as “pure”. Now why aren't they being marked “const” in the Ada case? The GCC notion “const” corresponds pretty much to Ada's notion of Pure. In `decl.c`, there is logic to mark a function as “const” if `Is_Pure` (an internal Ada attribute) is true. However `Is_Pure` is returning false for the math functions. Why? I will investigate — but whatever the reason it's basically a front-end bug.

## GUI & Multilanguage development

*From: Alex R. Mosteo <alejandromosteo.com>  
Date: Tue, 08 May 2007 21:28:04  
Subject: Re: Newbie Needs Ada Advice  
Newsgroups: comp.lang.ada*

> One idea I have, and I'd love to hear advice or comments, is to use Ada as the “engine” for my program, and somehow use Java to create the GUI and deal with images. Does this seem like a possibility?

I've done this, or something similar. My core was a stand-alone Ada program, with SOAP web services (again, thanks to AWS) for feedback/command. Any compliant SOAP client could interact with my core, and in fact I had a Java one.

If you want to have a single executable, I don't know how to link with Java... unless you use GCJ I suppose (or JGNAT, but I understand is heavily outdated and unmaintained). But then, you can easily go the Ada+C/C++ way. Calling Ada from C/C++ is easy; C from Ada likewise. C++ is a bigger problem that I usually avoid with a simple C gluing layer.

Then, all the myriads of libraries in C/C++ are there for you, and you can program the part you deem necessary in Ada (if any, of course).

As an example: my current project uses a core in Ada, that interfaces with a robotic library for both control and simulation that has a C API. Furthermore, I must exchange data using YARP (a C++ library) with some modules programmed by colleagues, and I do this with just a

simple C++ file with extern “C” exports for calling from Ada.

In truth, there's some overhead in going mixed-language. But once you define interfaces and stick to them, the rest is piece of cake.

But if your non-Ada concern is a good GUI, maybe you should take a look at the Ada bindings first. Maybe `GtkAda` is your deal (though if you're an Ada newcomer I don't think so ☺ or one of the others is...

I don't really know how people build the amazing GUIs I see in my KDE desktop. Which tools are out there? `KDevelop`? Some Qt builder? `Glade` doesn't impress me much, I think I'm also missing something in the `Gtk+` side.

As the bottom line of my personal current opinion: if I were to build a good GUI for my Ada code, I'd use whatever toolkit I liked best, in its native language, and glue it to Ada via C exports.

*From: Ludovic Brenta <ludovic@ludovic-brenta.org>  
Date: Tue, 08 May 2007 20:20:35  
Subject: Re: Newbie Needs Ada Advice  
Newsgroups: comp.lang.ada*

At one point you asked whether writing the “engine” in Ada and the GUI in another language was a option; the answer is “yes, there are several ways”. But OTOH you seem reluctant to learn more than one language, and you also seem to lack experience with GUI programming, so you're unbiased as to which toolkit you should use.

Personally I used Java Swing about 7 years ago, and wrote a quick toy GUI with `GtkAda` a couple of years ago in one afternoon. I used [`GNOME`] `Glade`, generated Ada code from the XML file and, wanting to learn more by myself, rewrote the Ada by hand in what I thought was a cleaner way. I think both Java/Swing and Ada/`GtkAda` are equally difficult, or equally easy, to use. If you like you can browse the sources for this toy program here:

<http://www.ada-france.org:8081/branch/changes/org.ludovic-brenta.meuse>

(the license is PURE GPL [...]). You'll notice that the “engine” and GUI are in separate packages, and the program runs equally well on the command line as with a GUI (i.e. with a command line parameter, it doesn't start the GUI).

So, if I were you I'd go for `GtkAda` and avoid learning another language just for the GUI. But rest assured that no matter what language and toolkit you use, GUI programs are many times larger and more complex than command-line programs, and be prepared to spend most of your time doing “boring” stuff like building dialog boxes. Also be prepared for a steep learning curve with any toolkit you choose. Sounds like a lot of fun, doesn't it?

*From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>  
Date: Tue, 8 May 2007 22:02:06  
Subject: Re: Newbie Needs Ada Advice  
Newsgroups: comp.lang.ada*

> One idea that now makes some sense (having thought about it for all of a couple of hours), is that the GUI/interface could be a separate application that simply sets up the experiment (i.e. microscope configuration). For example, creating an XML document with appropriate hardware parameters, and then feeding this document to the “engine”, which actually controls the microscope. The GUI could then be written using something even more high-level (like my favorite language `MATLAB!`).

You can call Ada from Matlab, same as you would call C. But, Matlab is not GPL, it is rather a lot of money. And it is extremely slow. (For our customers we often design hardware-in-the-loop systems with Matlab/Simulink mounted on the system's top. Of course the engine is not Matlab, as you suggested, otherwise it would never work. But the only reason why Matlab is used, is not fancy GUI, but modeling our customers (mechanical engineers) wanted to do in Simulink. These are open-end systems. It does not look like your case.)

> The engine could be written in Ada. I could also write another application that would graphically monitor the ongoing experiment (i.e. display images). This monitor could be web based, and simply interact with the data that is acquired by the microscope controller engine.

Simply interact means what? I bet a subprogram call is the simplest thing one can imagine. A separate GUI process would mean some communication middleman between the data acquisition/control engine and the GUI. TCP/IP sockets, RPC, OPC? And all this just in order to display an image, a couple of buttons and combo boxes? It looks like an architectural mess. From what you have described, GUI is about half man-week programming in `GtkAda`.

> The only issue that still bugs me is that most of the hardware drivers for the major microscope manufacturers (Leica, Nikon, Olympus, Zeiss) are in C/C++, so there will still be a need to interface Ada/C/C++ to some extent.

I would not consider it as a problem at all. The real problem with hardware interface libraries from third firms is that they are \*normally\* non-portable, non-functioning and have documentation describing something quite different from what you get. Add here, that the devices are customary malfunctioning. If you send the command A followed by B, the device software crashes. When you make a full-

duplex communication, it crashes again. When A is sent more frequently than each 100ms it crashes once more, and so on and so forth. Usually GUI is your least problem.

*From: Chip and Allie Orange  
<acorange@comcast.net>  
Date: Thu, 10 May 2007 14:14:59  
Subject: Re: Newbie Needs Ada Advice  
Newsgroups: comp.lang.ada*

I can't remember your development host requirements, but if it's only Windows, you've always got the option of using A#, and therefore all the visual studio GUI capabilities.

I'm in a similar position to you; being a newbie to Ada I'm about to start my own first large Ada project, and I think A# is what I'll try first. Not for GUI reasons, but because of it's availability of some development tools specific to my project.

If I had a need for a platform-independent GUI, I'd certainly choose HTML (via AWS) over Java.

*From: Richard Riehle  
<adaworks@sbcglobal.net>  
Date: Sun, 27 May 2007 05:50:18  
Subject: Re: Newbie Needs Ada Advice  
Newsgroups: comp.lang.ada*

I learned recently that AdaCore is planning a resurrected modification of JGNAT that will allow Ada to interface to JNI libraries. If this works, it should allow you to do what you just described.

## Advantages of tagged types

*From: Randy Brukardt  
<randy@rrsoftware.com>  
Date: Tue, 15 May 2007 19:27:31  
Subject: Re: Generic Collection  
Newsgroups: comp.lang.ada*

> If your implementation guidelines say that everything should be tagged I'd have to ask why? (I don't believe that programming-language inheritance is necessarily a good way of implementing application area specialization/generalization).

Not to speak for others, but there are a lot of reasons:

- (1) Most complex types need to be controlled in order to properly manage their memory. Making them visibly tagged thus has no overhead.
- (2) Maximum flexibility: allowing future clients the ability to extend the type without modifying the base type likely reduces future maintenance (can't break something that you don't change). I don't care much for or about inheritance, but extension is a big win.
- (3) Tagged types "work right" in Ada: "=" composes properly, user-defined operations are used in generics (not true for untagged types, where the predefined ones "reemerge"), parameter passing is

consistent; 'Access can be used in the implementation without forcing users to declare everything in sight aliased, you can use prefix calls on the objects, and there may be more that I've forgotten.

Note that none of these have anything to do with classwide programming, dynamic dispatching, or inheritance, or in fact the tag itself. (3) was an important enough issue that we actually considered adding an "aliased" type to the Amendment which would provide those benefits without requiring a tag. But in the end it was considered to be not enough of an improvement.

Conclusion: (almost) all new ADTs in Ada should be visibly tagged and possibly visibly derived from Controlled.

*From: Benjamin Place  
<benjamin.place@sbcglobal.net>  
Date: Wed, 16 May 2007 09:27:49  
Subject: Re: Generic Collection  
Newsgroups: comp.lang.ada*

That echoes Prof. Dewar's recommendation in the graduate programming languages (IIRC) class at NYU.

Always nice to see the gurus in agreement.

## Ada and Multi-core CPUs

*From: Gerd <GerdM.O@t-online.de>  
Date: 29 Jan 2007 03:57:20  
Subject: Does Ada tasking profit from multi-core CPUs?  
Newsgroups: comp.lang.ada*

Does anyone have experience with Ada tasking (especially GNAT) on multi-core systems?

Show programs with several working tasks a performance boost on dual-core or quad-core CPUs?

*From: karl@grebyn.com  
Date: 9 Feb 2007 02:18:07  
Subject: Re: Does Ada tasking profit from multi-core CPUs?  
Newsgroups: comp.lang.ada*

During November and December, I had an opportunity to evaluate the Sun Fire T1000, which has 8 cores with 4 strands per core, making it appear to be a 32 CPU machine. My application used GNAT with lots of long-lived threads. You can read my report at <http://www.grebyn.com/t1000> — it was impressive enough that I actually WON the evaluation system from Sun in their Open Performance Contest (see <http://www.sun.com/tryandbuy/prm/perf/winners.jsp> for other winners — primarily web servers, databases and other transactional systems).

*From: Dmitry A. Kazakov  
<mailbox@dmitry-kazakov.de>  
Date: Tue, 6 Mar 2007 09:44:14  
Subject: Re: Does Ada tasking profit from multi-core CPUs?*

*Newsgroups: comp.lang.ada*

[...]

> Can a task switch cores? If yes, what is the overhead of switching?

That depends on the mapping of Ada's schedulable units to the OS ones. But it is Ada question too. Because the processor affinity of a task is not specified, the scheduler should be able to switch it from processor to processor to achieve an optimal performance. Otherwise multi-core would make no sense for Ada. AFAIK, Windows indeed switches thread's processors, so GNAT 3.15p should follow it as well (provided, tasks are mapped on threads and no thread affinity mask is set). Interesting would be to know the penalty of such switching.

> I don't know how one asks "which core am I currently running on" — or indeed if that questions makes any sense.

I suppose it should be `NtGetCurrentProcessorNumber` under Windows. That apparently is only available on Windows Server 2003 and Vista.

*From: Steve <steved94@comcast.net>  
Date: Tue, 6 Mar 2007 19:58:31  
Subject: Re: Does Ada tasking profit from multi-core CPUs?  
Newsgroups: comp.lang.ada*

On Windows, which uses symmetric multiprocessing, I believe two cores works the same as two CPUs. With two CPUs the two highest priority threads that are in the ready state run concurrently, so yes a task can switch cores. Sorry I don't know about the overhead of switching. I have run tests on a system with 2 CPUs and found that a single task that does a lot of switching winds up using 50% of the CPU time on both CPUs (from the task viewer).

*From: Jacob Sparre Andersen  
<sparre@nbi.dk>  
Date: Fri, 16 Mar 2007 15:29:15  
Subject: Arguments for single-mutex-exclusion on protected types (Was: Does Ada tasking profit from multi-core CPUs?)  
Newsgroups: comp.lang.ada*

> The last time I looked at the GNAT sources for protected objects, each had a mutex associated with it that was obtained before any action, even functions. I guess that hasn't changed.

Isn't that the most efficient implementation on a POSIX system?

a) We use threads (and not processes) for tasks, since it is more efficient and a better conceptual match.

b) We know that it is considered good style to make protected function bodies small.

c) Since we use threads (a) we should use mutexes (and not semaphores) to implement inter-task exclusion.

d) Since the protected function bodies can be assumed to be small (b), there will be a relatively large overhead in keeping track of the number of queued calls to protected functions and procedures.

e) Since the protected function bodies can be assumed to be small (b), the probability of colliding calls to protected functions is relatively small.

f) It is thus likely that using a single mutex to provide exclusion on a protected object is more efficient than adding counters.

The balance in (f) depends on your average collision rate for protected function calls. More concurrent threads (more CPU cores) increase this value. So does larger protected function bodies.

Now that multi-CPU-core systems are more common, it may be worthwhile to make a proper examination of the numbers which may change the balance against (f).

This could also be a challenge for code profilers and optimisers. It may be that some protected types need a detailed queue, while others need a single mutex — depending on the code and the number of available CPU cores.

*From: Jeffrey R. Carter*  
<jrcarter@acm.org>

*Date: Sat, 17 Mar 2007 17:52:55*

*Subject: Re: Arguments for single-mutex-exclusion on protected types (Was: Does Ada tasking profit from multi-core CPUs?)*

*Newsgroups: comp.lang.ada*

> [...] why do you say ceiling locking won't work if function calls lock out other function calls?

As I understand ceiling locking (which probably isn't as well as you), its whole point is the absence of any actual lock. The necessary mutual exclusion is achieved through tasks' actual priorities and their positions in the ready-to-run queues when they're preempted.

I guess the behavior will be the same with or without the lock. Ceiling locking might be a little faster since it eliminates the overhead of the explicit lock.

*From: Randy Brukardt*  
<randy@rrsoftware.com>

*Date: Sat, 17 Mar 2007 18:06:57*

*Subject: Re: Arguments for single-mutex-exclusion on protected types (Was: Does Ada tasking profit from multi-core CPUs?)*

*Newsgroups: comp.lang.ada*

Isn't it true that ceiling locking is essentially irrelevant on multiprocessor systems? That is, the point is to get rid of the lock, but you can't do that on a multiprocessor (one processor can be running a lower priority task without anything being wrong, and that task had better be blocked from accessing the protected object). So

ceiling locking has no advantage on a multi processor, it just restricts what you can do. If multiple cores continue to grow in popularity, it seems that the whole ceiling locking thing will become essentially irrelevant — just another case of premature optimization. [...]

*From: Randy Brukardt*  
<randy@rrsoftware.com>

*Date: Mon, 19 Mar 2007 16:49:12*

*Subject: Re: Arguments for single-mutex-exclusion on protected types (Was: Does Ada tasking profit from multi-core CPUs?)*

*Newsgroups: comp.lang.ada*

> On a multiprocessor, you can use ceilings plus spin locks to protect protected objects. The point is to avoid any queued waiting to enter a PO (all the queuing is done with entry queues in this model). That can be made quite efficient, and the ceilings still prevent certain forms of priority inversion. So, no, I don't think ceilings are entirely irrelevant on multiprocessors.

I believe that latter point (ceiling prevent some forms of priority inversion, because they boost the priority of everything in the PO), but I don't see the former. I don't see any reason that you would have to use a queued (rather than a spin) lock with or without ceiling locking. Until you have the spin lock, your priority doesn't matter (if you get pre-empted, so what?). And afterwards, it's just a special case of the normal potential priority inversion of a PO: if it isn't an issue for the entire object (assuming you can start a protected action), it surely won't matter how you start that action.

The problem with ceiling locking is that it depends on boosting the priority of tasks. That means its a big problem for longer-running operations (such as I/O, which aren't allowed in protected operations for this reason). And it's a big problem for reusable libraries, which can't know ahead of time what the ceiling ought to be. (Make it too high, and critical tasks could be starved by lower-priority ones operating in the library, make it too low and tasks aren't even allowed to access the library.) Consider trying to set the ceiling for a container library implemented with protected objects. (At least we now can do this on the fly; in Ada 95, it was impossible.)

> Ceilings are largely irrelevant if you're building Ada tasking on top of some other system (posix or windows threads, for example) that wants to do things differently. I mean, if you're using the one-Ada-task-per-thread model.

Is there any other kind? I haven't heard of many bare-machine Ada projects in recent years; almost everything is on top of some sort of RTOS or other OS. (That's too bad, really, for a lot of projects, Ada

provides nearly everything you need in an RTOS.)

> Another issue is that Ada's priority model has nothing to say about processor affinity.

True enough, but I don't think most OSEs have much to say on this topic, either. That makes it pretty hard to say anything about it (unless you use a one-thread for all tasks model — but that's a pessimising implementation!)

*From: Randy Brukardt*  
<randy@rrsoftware.com>

*Date: Tue, 20 Mar 2007 12:51:54*

*Subject: Re: Arguments for single-mutex-exclusion on protected types (Was: Does Ada tasking profit from multi-core CPUs?)*

*Newsgroups: comp.lang.ada*

> My concern was that you said it was impossible in Ada 95. It may not be ideal, but it doesn't seem to be impossible.

I was thinking about a solution that doesn't clutter the client's view of the library with (usually) irrelevant details (such as whether the library is implemented with protected objects). After all, information hiding is good! If you're willing to ignore that (and you are), then it certainly is possible. But I was thinking about a library that uses as the ceiling whatever the highest priority it is called with: that can't be implemented in Ada 95. (Such a library would not make anything having to do with priorities visible.)

If you really care about priorities, then your solution is probably better (it allows more analyzability). Which just demonstrates that you can't just make something "task-safe". You have to answer the question of "task-safe for what?". And that tends to lead to families of libraries rather than an all-in-one solution (like Ada.Containers) — or impacts reusability.

*From: Robert A Duff*

*Date: Mon, 26 Mar 2007 19:38:36*

*Subject: Re: Arguments for single-mutex-exclusion on protected types (Was: Does Ada tasking profit from multi-core CPUs?)*

*Newsgroups: comp.lang.ada*

Info hiding is good, but one can't always have it, sadly.

If we have human beings specifying numeric priorities of things, then we have the property that priorities only have meaning relative to other priorities. And that property implies that priorities are a global issue — can't be neatly encapsulated. That's true whether we're talking about priority of tasks, or PO's (ceilings), or I/O events, or anything else. So I wouldn't blame this on ceilings specifically — I'd blame on the general

model (numeric priorities, set by programmers).

In a non-real-time context, it's easy to make it all automatic. I haven't typed "nice" on Unix very often lately, and yet my editor responds well, even when I have a compute-bound process also running.

In a real-time context, we have various partial solutions (earliest deadline first, and so forth).

## ANNA — Annotation Language for Ada

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

*Date: Sat, 12 May 2007 21:19:37*

*Subject: Re: Where is "Anna" — an old Ada language extension*

*Newsgroups: comp.lang.ada*

> Anna is an old language extension of Ada (Ada 83, as I believe) to include facilities for formally specifying the intended behavior of Ada programs. As understand, Anna is bit like today's SPARK. See [http://pavg.stanford.edu/previous\\_research/index.html](http://pavg.stanford.edu/previous_research/index.html) [snip]

I can't help you with the location of the Anna archives (although others have) but I can help with its relationship with SPARK. ANNA is an acronym which is something like "Annotation Language for Ada". Like SPARK, the idea was to add annotations (e.g. pre/post conditions) to subprograms to facilitate proof using Hoare logic. ANNA was more ambitious than SPARK because it attempted to cover the entire Ada language. I am not sure there ever were any ANNA tools.

When we designed SPARK we found it necessary to limit the full Ada language in various ways to make it unambiguous so that the proofs actually meant something!

*From: Ed Falis <falis@verizon.net>*

*Date: Sat, 12 May 2007 21:56:22*

*Subject: Re: Where is "Anna" — an old Ada language extension*

*Newsgroups: comp.lang.ada*

> I am not sure there ever were any ANNA tools.

There were. I was on the team in 1982–3.

*From: Manuel Collado*

*<m.collado@lml.ls.fi.upm.es>*

*Date: Fri, 11 May 2007 19:43:46*

*Subject: Re: Where is "Anna" — an old Ada language extension*

*Newsgroups: comp.lang.ada*

It seems that the files are still there: <ftp://pavg.stanford.edu/pub/anna/read.me>

## Ada Advice Inquiry

*From: Richard Riehle*

*<adaworks@sbcglobal.net>*

*Date: Thu, 3 May 2007 19:01:08*

*Subject: An Ada Advice Inquiry*

*Newsgroups: comp.lang.ada*

I recently received an inquiry from a colleague I have not seen in many years. He is currently in a position where he is asked to advise a client about whether to choose Ada or C++ for a project. He is not a language junkie because his expertise is a very different level of software engineering, but he is a highly intelligent and capable consulting engineer.

Perhaps that's why he inquired of me. ☺

One of the concerns is whether anyone else is choosing to use Ada for new projects. He does not want to be the only one to make such a choice. It is well-known that there are a lot of Ada software products being maintained, but is anyone choosing it for new projects? Good question, I answer.

So. What do I tell him? At one time, I could contact the AJPO and ask this question. Although they were never very good at providing complete information, they did have some knowledge of where Ada was being used for new projects. Now, there is no place to look for that information.

It presents a little dilemma. If no one else is using Ada for new projects, he can wonder why he should be the alone in his decision. If every other consultant, manager, or program director asks the same question, and gets no information about others choosing Ada, no one will want to make that choice.

Is anyone choosing Ada for new projects? What kind of projects? Can we collect information about them and create a catalog of them in the [adaic.com](http://adaic.com) website? If not, why not?

The compiler publishers know who is buying their compilers, but they don't want to compromise customer confidentiality. The programmers are afraid of losing their jobs if they tell anyone what they are doing. Everyone who can provide the information is afraid to do so. When I was actively consulting on Ada, I was often told, by my client, not to tell anyone they were using Ada. So. We keep the secret. The secret keeps Ada a secret. And Ada begins to wane because no one wants to reveal their decision to use it.

Is this a recipe for killing any incentive for choosing the language for future projects? What do I tell my old friend? Should he take the risk of using Ada even without the knowledge that anyone else is using it for real projects? Hmmmmmm. Is anyone out there really choosing it for new software projects?

*From: Richard Riehle*

*<adaworks@sbcglobal.net>*

*Date: Fri, 4 May 2007 08:31:58*

*Subject: Re: An Ada Advice Inquiry*

*Newsgroups: comp.lang.ada*

> Why is this a concern?

It is a concern because of the appearance of diminishing usage of the language. He sees C++ magazines, books, articles, and information without even seeking it. As noted in another thread, the Cohen book is out-of-print and will not be reprinted or updated. Few other books are available for the new Ada, at this writing

Anyone who wants to start a new project also wants to know that there is going to be support over the long haul.

> I can understand needing to be able to rely on Ada compilers/tools being available in the future, and on being able to find people trained in Ada. Surveying available third party packages is harder. I have not even tried in my field; hard real-time satellite simulation.

Is your organization creating satellite software in Ada? Are you at liberty to identify what company you are with and what kind of satellites you are developing using Ada for the software?

I need to prepare my reply to my colleague today, but I can always send follow-up information early next week.

*From: Stephen Leake*

*<stephen\_leake@stephe-leake.org>*

*Date: Sat, 05 May 2007 19:44:21*

*Subject: Re: An Ada Advice Inquiry*

*Newsgroups: comp.lang.ada*

[...]

> Are you at liberty to identify what company you are with and what kind of satellites you are developing using Ada for the software?

NASA Goddard Space Flight Center.

My team uses Ada; we develop simulators for testing satellite software before launch.

I hope that eventually, enough people will have rotated thru my team that some flight software team will seriously consider using Ada. It could happen!

*From: Randy Brukardt*

*<randy@rrsoftware.com>*

*Date: Fri, 4 May 2007 00:47:01*

*Subject: Re: An Ada Advice Inquiry*

*Newsgroups: comp.lang.ada*

> I don't know if it would be considered a compiler company, but Praxis seems to be actively developing new reliability tools with Ada, I presume for new developments, and they seem willing to be public.

Speaking of Praxis, they just had an announcement on a new air traffic control system (in the UK) that is programmed in Ada. You can find a link to it on the news page at [www.adaic.com](http://www.adaic.com) (it's too late for me to go look it up right now).

*From: Rod Chapman*

*<roderick.chapman@gmail.com>*

*Date: 4 May 2007 01:47:06*

*Subject: Re: An Ada Advice Inquiry*

*Newsgroups: comp.lang.ada*

Indeed — iFacts is very large, new, and using Ada (and SPARK of course...) [...] We\_won\_ this project by bidding Correctness by Construction, Formal Methods, Ada, SPARK etc. etc, against some very stuff competition.

*From: Fionn Mac Cumhaill  
Date: Sat, 05 May 2007 02:35:41  
Subject: Re: An Ada Advice Inquiry  
Newsgroups: comp.lang.ada*

I just started a new project where I work, and I'm using Ada. It's about as different from an air traffic control system as you can get. I'm going to be using Ada to interrogate a MySQL database and push the resulting reports to a web server on our in-house network. I'm running a 2-person shop, and the other guy only does Crystal Reports, so I need all the help I can get from my programming language. I don't want to waste time and effort on fighting both C and SQL. Ada and I can gang up on SQL and beat it into submission.

*From: xavier <xavier@ipnnaval.in2p3.fr>  
Date: Fri, 04 May 2007 11:52:58  
Subject: Re: An Ada Advice Inquiry  
Newsgroups: comp.lang.ada*

I'm working in a research physic laboratory. We use Ada for our new acquisition system (Narval). The system is a set of process coordinated at low speed by Annex E of Ada 95. For high speed data transfer we use socket and Unix FIFO.

At the start of the project only our local particle accelerator used it, but now we have the responsibility of the data flow of an European detector named AGATA (220 Kg of very pure germanium, 180 detectors, 7000 channels). The system will have to handle about 13,5 GBytes/s out of the electronics boards down to many CPU farms, the system will have to process them and store data at 100 MBytes/s. Others physic teams also use it now for their experiments.

*From: John McCormick  
<mccormick@cs.uni.edu>  
Date: 5 May 2007 05:27:03  
Subject: Re: An Ada Advice Inquiry  
Newsgroups: comp.lang.ada*

> One of the state schools in the area just dropped C++ as the primary teach language (still offered)...

My state school, the University of Northern Iowa, has just decided to drop Java and return to Ada in the first year courses. The faculty nearly unanimously agreed that our students learned more about computer science with Ada than Java. It is not unusual for a school to drop a language, it is unusual to return to something that worked rather than move on to the latest fad.

*From: Ludovic Brenta <ludovic@ludovic-brenta.org>*

*Date: Fri, 04 May 2007 11:08:51  
Subject: Re: An Ada Advice Inquiry  
Newsgroups: comp.lang.ada*

At Barco Avionics, all our products use Ada: both the ones we evolve from existing ones, and the new ones. We have no plans to change

The web site, unfortunately, doesn't tout that: <http://www.barco.com/aerospace/>

*From: Christoph Grein (Fa. ESG)  
<Christoph.Grein@eurocopter.com>  
Date: Fri, 04 May 2007 13:53:43*

*Subject: AW: An Ada Advice Inquiry  
Newsgroups: comp.lang.ada*

> Can I infer, from your reply, that Eurocopter is also using Ada?

For Tiger, NH90, two different flight simulators for Tiger (delivered) and a new simulator for NH90.

*From: Ludovic Brenta <ludovic@ludovic-brenta.org>*

*Date: Fri, 04 May 2007 13:41:32  
Subject: Re: AW: An Ada Advice Inquiry  
Newsgroups: comp.lang.ada*

> Nice to hear. I guess these CDUs are the ones that are used in Tiger H/C.

Yes. That's one of our first products, we call it the CDMS-1TG, and most of it is programmed in Ada (there is also some C in the keyboard microcontroller).

*From: Richard Riehle  
<adaworks@sbcglobal.net>  
Date: Sun, 6 May 2007 07:47:40  
Subject: Re: An Ada Advice Inquiry  
Newsgroups: comp.lang.ada*

[...] What I have learned, from contributions to me inbox and comments in this forum, is that a great many new projects are being started in Ada. As usual, quite a few of my correspondents have asked not to be quoted/cited.

I have also had some “buyer's remorse” messages. That is, more than a few of those email messages have said how disappointed they are with the decision to use C++ instead of Ada. The C++ decision has not been as successful as they had hoped, and there is some wishful thinking that they would have been better to have stayed with Ada. Now that the code is written in C++, they have to live with the agony of C++ while longing for the ecstasy of Ada. I suppose the “grass is always greener” effect is a human trait that will endure for as long as humans inhabit the planet.

I did send another email to my colleague yesterday summarizing the comments from this forum and from the contributions to my inbox. This morning (Sunday), I received a few more messages with similar information. Again, I have been asked not to cite or attribute any of the information specific people. Some of these contributors are not corporate officers with official authorization to speak about the projects they are doing.

Rather, they are worker-bees who are speaking for themselves, and revealing information that might get them fired. Therefore, I am stripping attribution from those kind of messages before providing it to my colleague, the person who originally made the inquiry of me.

As to the suggestion that I make this inquiry of the C++ forums, my experience is that the vast number of C++ users are woefully ignorant about Ada. This is certainly the case where I am currently employed. On the other hand, C++ is so obnoxiously ubiquitous that a person who uses Ada cannot avoid C, C++, or Java. In my own case, I have to use C++, teach it in my classroom, and endure its messiness on a regular basis. The more I see of it, the more I learn of it, and the more deeply I am required to spend time with it, the more I realize how much better Ada is for most things.

That being said, I also have come to like Python for a lot of my day-to-day programming. Ada is not perfect for every project. There are other languages that have great virtue for different kinds of projects. However, as nearly as I can tell, at this point in the world of programming, C++ has outlived its usefulness for most of the tasks for which it was once used. It has become the close equivalent of an object-oriented Assembler, forcing a lot of programmers to focus on low-level concerns that ought to be a marginal aspect of modern software engineering.

## What is wrong with Ada?

*From: Chip and Allie Orange  
<acorange@comcast.net>  
Date: Mon, 9 Apr 2007 21:14:01  
Subject: Re: What is wrong with Ada?  
Newsgroups: comp.lang.ada*

> I've never been scared of “non mainstream” languages, but I was wondering, what is wrong with Ada? Why don't more people use it? What do most people think of it? To me, it seems like a nice, concise, and safe language.

I'm just now starting to learn Ada, but I've been “looking at it” for quite a while. I'd say the answer to your question is that no one uses it much, because, no one uses it much.

My boss won't allow anything at work that isn't already being used by everyone else. A lot of software choices are made for more or less those reasons. As a corollary to this, it's impossible for us to hire anyone with Ada programming experience, as an entry level programmer, and hiring someone without experience in your software development systems of choice “just isn't done”. Since what we have access to are endless visual studio programmers, that was used as an argument for what we should be using to develop with. (...)



From: Jim Rogers

<jimmaureenrogers@worldnet.att.net>

Date: 21 Apr 2007 14:42:32

Subject: Re: What is wrong with Ada?

Newsgroups: comp.lang.ada

(...) Several years ago I taught Java to an investment firm. All their legacy code was written in COBOL. The Vice President in charge of information systems announced, without consulting any developers, that the company was moving all its programming to Java. During the class one of the students asked me what kind of performance improvement they should expect from Java over COBOL. I told them, if they were lucky they would see a -30% increase. Yes, that is a negative increase. Java was not cheaper than COBOL for this organization. They currently had all the tools and training needed to support COBOL. According to the students the Vice President made the change after talking with some of his friends at his country club. Everybody else was using Java. He felt embarrassed that his organization still used COBOL.

Programming language choices are seldom made based upon either economics or technical merit. Managers choose programming languages because it is safe to run with the herd.

> We downloaded Java for free with free Netbeans and Eclipse. If we need a "tool" then we do need to go through the evaluation process. But the problem is that Ada is the "base" and not a "tool". No one is going to pay for entry into a development language when others are free. People will pay to enhance the tool. But paying to evaluate something on a small scale for a year isn't going to happen on a regular basis.

Try downloading C++ for free with the equivalent of Netbeans. Eclipse is a wash for all its supported languages. Did your application need to do anything Java cannot do by itself? For instance, did it need to interface with any existing libraries written in another language? If so, you needed another language such as C or C++, as well as somebody who understands how to write a Java Native Interface for the foreign code. This means acquiring tools and training in at least two languages.

What is the cost of tools and training for any specific language? How does that cost compare to the development and testing costs for production code? The tools and training costs are typically less than 1% of the development costs for a project producing over 1,000,000 SLOC.

One might well ask what are the different testing costs for different languages. Many people would be surprised by the question. As mentioned in another thread, I am the software safety engineer on an

extremely safety critical software project. The management chose C++ as the primary language for development of new code for this project. The decision was made before I was hired. For C++ to be as safe as Ada, you need to specify a very strict subset of C++. One example of such a subset is the JSF C++ Avionics Coding Standard. It contains 221 rules.

Approximately 200 of those rules are specification of a C++ subset. The cost of verifying 221 rules against 2,000,000 lines of code is extremely high. Some of the rules checking can be automated. Some cannot. None of those costs (manual checking and creation of automated checkers) would have been encountered if management had chosen Ada. None of those costs were considered when the decision was made.

From: Tom Moran <tmoran@acm.org>

Date: Wed, 11 Apr 2007 12:53:16

Subject: Re: What is wrong with Ada?

Newsgroups: comp.lang.ada

> aren't a whole bunch of Ada developers out there but there are schools on several continents churning out armies of Java developers.

In the early days of the California gold rush, miners lived in tents because lots of people were able to put up tents in a hurry. Later they realized that, though quick, tents aren't as robust against the elements as houses, and the miners who made money built houses, even when they had to hire some skilled craftsmen to do it.

From: Richard Riehle

<adaworks@sbcglobal.net>

Date: Sat, 21 Apr 2007 05:51:27

Subject: Re: What is wrong with Ada?

Newsgroups: comp.lang.ada

> So perhaps most of the people don't write safety-critical software? And there is, consider, more than one kind of safety ...

Granted. I am not a one-language person. It is always a matter of choosing the right tool for the right job.

I was once asked by a three-star general whether he should be converting all his Fortran code to Ada, as many on his staff wanted to do. I gave him my opinion as, "No." Old code that has worked a long time is almost always better than new code that has not yet proven itself in the field, regardless of what language is used.

In his environment, there were a lot of programs that needed to be written other than translating the Fortran. It seemed to me that those who wanted to translate Fortran were simply avoiding the new problems that were waiting to be solved, and that could best be solved in Ada.

A large bank was persuaded to do a bunch of projects in Java. It was a stupid decision, and that soon became apparent. All the other code in the bank was in

COBOL. My advice. Instead of using Java or C++, or even Ada, train the programmers to use Object-oriented COBOL. Java is a terrible choice for business data processing where there is a lot of accounting activity. C++ is probably worse. Ada could work because of the built-in accounting-oriented data types, but that would also a wrong choice for this environment.

We live in a multi-language world. Perhaps it would be better if we didn't. However, there does not exist, at present, any single language that is superior for every kind of programming problem in every context. One thing I like about Ada that is not as well-formed in other languages is the Annex C, Interfacing to Other Languages facility. Ada is more hospitable to other programming languages than most of the alternatives.

From: Jason King <jhking@airmail.net>

Date: Wed, 11 Apr 2007 10:21:08

Subject: Re: What is wrong with Ada?

Newsgroups: comp.lang.ada

(...) We don't use C/C++ at our shop because its difficult to get the task accomplished quickly so we look to other tools. As we survey the landscape we see [Visual Basic], Delphi, Ada, Java etc.

Initially Ada looks very attractive because we're an Oracle shop and PL/SQL (the stored procedure/trigger language inside the Oracle DB) is based on Ada 83. Then we look for database access, tools to build client GUIs and tools to build web applications. The tools available are substantially inferior in number and outside documentation to the ones available for Java. Then the bosses look around to see how hard it is to hire new people for the shop as our client list expands. There aren't a whole bunch of Ada developers out there but there are schools on several continents churning out armies of Java developers. Maybe the one ada developer is more productive than 1.5 or even two java developers but when we have to overcome the lack of tools + the lack of developers that makes us look other places. (...)

From: Alexander E. Kopilovich

<aek@VB1162.spb.edu>

Date: Thu, 12 Apr 2007 22:55:24

Subject: Re: What is wrong with Ada?

Newsgroups: comp.lang.ada

> (...) In the world as it exists the availability of support materials and experienced developers outweighs the technical merits of Ada.

The word "support" is crucial here. Ada is exactly support-oriented language. But "support" for Ada means regular professional support, and not a kind of random support from the elements in and around the industry.

Therefore, if you can afford such regular professional support (for example, like that provided by AdaCore for users of

their GNAT Professional compiler) then you can use Ada for your business applications quite successfully. But if not, then, well, perhaps you better do not try to use Ada in your business applications unless your developers are great Ada enthusiasts.

*From: Richard Riehle  
<adaworks@sbcglobal.net>  
Date: Tue, 20 Mar 2007 17:37:05  
Subject: Re: why learn C?  
Newsgroups: comp.lang.ada*

[...] Ada is a language designed to maximize the amount of error detection as early in the development process as possible, and throughout each subsequent step of that process.

Ada provides the developer with more engineering tools and idioms for the development of software than one finds in a lot of other languages. A source code model, in the form of Ada package specifications, can be created long before anyone writes any algorithmic code. The entire design can be structured and tested prior to committing too much to small details. For a large project, where many thousands of lines of code are required, that project can be decomposed and distributed over several programming teams without excessive worry about whether the resulting code will conform to the design.

Unlike the C family of languages, which are generally no help at all, an Ada compiler frequently eliminates the need for extensive debugging of silly errors; e.g., the kind that originate from case-sensitivity, wrong order of arguments to parameters, improper spacing, missing curly-braces, and the plethora of language “features” that plague those who have chosen to use any member of the C family of languages.

It has always seemed rather odd to me that people who are as intelligent as programmers, rarely question the silliness of using tools such as C and C++ for serious work. They simply take for granted that the kind of errors that are so common in those languages are somehow akin to the laws of nature. Java inherits many of those same properties and is not much better. It is taken as an article of faith that programming involves heavy use of debuggers. Instead of asking why this is so, they simply believe it is a fundamental law of programming — mainly because the languages they have chosen make it so.

Ada is a programming language that makes it easier to focus on the design before dealing with the details of that design. Then, Ada makes it easier to ensure that the details conform to the design. I wonder why that is such a difficult concept for so many.

*From: Brian May  
<bam@snoopy.apana.org.au>*

*Date: Mon, 16 Apr 2007 12:09:48  
Subject: Re: why learn C?  
Newsgroups: comp.lang.ada*

I thought the following links were interesting.

[ - “Autoconf and #ifdef Considered Harmful” ]

[ - “Rusty on Interface Simplicity” ]

”Rusty says the key criteria for an interface is how easy it is to use correctly or (by Murphy’s Law) how hard it is to misuse. He delineates a spectrum from best to worst, which is so good I am going to shamelessly reproduce it here:

1. Compiler/linker won't let you get it wrong.
2. Compiler will warn if you get it wrong.
3. The simplest use is the correct one.
4. The name tells you how to use it.
5. Do it right or it will break at runtime.
6. Follow common convention and you'll get it right.
7. Read the documentation and you'll get it right.
8. Read the implementation and you'll get it right.
9. Read the correct mailing list thread and you'll get it right.
10. Read the documentation and you'll get it wrong.
11. Follow common convention and you'll get it wrong.
12. Do it right and it will break at runtime.
13. The name tells you how not to use it.
14. The obvious use is wrong.
15. Compiler will warn if you get it right.
16. Compiler won't let you get it right.
17. It's impossible to get right.”

## Allocation and Deallocation

*From: Maciej Sobczak  
<maciej@msobczak.com>  
Date: 9 May 2007 09:27:25  
Subject: Unchecked\_Deallocation vs. delete  
Newsgroups: comp.lang.ada*

What’s the benefit of Unchecked\_Deallocation as a generic library procedure vs. built-in deallocation operator like delete in C++?

The disadvantage, as far as I perceive it, is that it breaks the symmetry that should be expected with regard to the allocation operation. If “new” is built-in, then the deallocation should be built-in as well. Making it a generic library procedure just makes more work for the programmers for no clear reason.

What clear reason am I missing?

*From: Robert A Duff  
Date: Wed, 09 May 2007 16:54:42  
Subject: Re: Unchecked\_Deallocation vs. delete  
Newsgroups: comp.lang.ada*

I think you're right. Not missing anything.

The original reason why U\_D is a generic procedure is probably because the designers of Ada 83 imagined that garbage collection would be ubiquitous, so U\_D would rarely be necessary. I suppose they thought that “with U\_D;” would be a red flag, at the top of the package that does such “evil” things. Note that U\_D is banished to chapter 13, where all the other machine-dependent stuff lives.

*From: Robert A Duff  
Date: Wed, 09 May 2007 16:56:49  
Subject: Re: Unchecked\_Deallocation vs. delete  
Newsgroups: comp.lang.ada*

> The symmetry is apparent. Pointers can be constructed using two methods: new T and X'Access.

Except that X'Access did not exist when this design decision was made. In Ada 83, “new” and U\_D really are symmetric.

> ...Further even if a pointer is constructed with new, it can be subject of GC.

Right.

*From: Keith Thompson <kst-u@mib.org>  
Date: Wed, 09 May 2007 13:59:55  
Subject: Re: Unchecked\_Deallocation vs. delete  
Newsgroups: comp.lang.ada*

Allocation is safe; it either succeeds or fails cleanly. Deallocation, if you happen to deallocate the wrong thing, can fail in arbitrarily bad ways. Requiring you to instantiate Unchecked\_Deallocation (emphasis on “Unchecked”) is intended as a reminder that it's entirely up to \*you\* to get it right; if you mess up the argument, the implementation isn't going to save you.

*From: Randy Brukardt  
<randy@rrsoftware.com>  
Date: Wed, 16 May 2007 14:25:25  
Subject: Re: Unchecked\_Deallocation vs. delete  
Newsgroups: comp.lang.ada*

'new' is not an operator, it is a primitive operation. Operators (like "+", "abs") can be user-defined; operations (like "new", "in", "!=") cannot. (Well, actually operators are also operations, so it's really only operations that are not operators that cannot.) [...]

Note that some operations like "new" and "!=" can have some user-defined behavior using another mechanism (storage pools for "new", controlled types for "!="), but that is different than being completely user-defined. Complete redefinition would be hard for operations, because some of the needed semantics cannot easily be modeled as a subprogram call.

## Limited interfaces and the LSP

*From: Ludovic Brenta <ludovic@ludovic-brenta.org>*

*Date: Wed, 23 May 2007 22:32:49*

*Subject: Re: Ada Interfaces and the Liskov Substitution Principle*

*Newsgroups: comp.lang.ada*

> To me, it seems as if Ada 2005 is bluntly violating the Liskov Substitution Principle. [...]

I think the declaration of Stepchild.Object is illegal because ARM 3.9.4(12/2) states: "A type derived from a nonlimited interface shall be nonlimited." [...]

*From: Randy Brukardt <randy@rrsoftware.com>*

*Date: Wed, 23 May 2007 17:00:51*

*Subject: Re: Ada Interfaces and the Liskov Substitution Principle*

*Newsgroups: comp.lang.ada*

Right. The most general type of interface is a limited interface; every interface should be declared limited if possible. (We didn't make that the default solely because it would be inconsistent with the rest of the language.) And if it is not possible to declare the interface limited (as in the example), then all types that include it must be nonlimited. At least interfaces don't have the rather limiting rule that regular tagged types do, where the limitedness can't be changed at all.

*From: Dmitry A. Kazakov <mailbox@dmitry-kazakov.de>*

*Date: Thu, 24 May 2007 20:39:14*

*Subject: Re: Ada Interfaces and the Liskov Substitution Principle*

*Newsgroups: comp.lang.ada*

> Well a task type is limited by default, idem for a protected object types.

Randy probably meant that if it were limited then one would need to introduce "not limited" qualifier to be able to

declare a non-limited interface. (In addition to awful "not null")

*From: Randy Brukardt <randy@rrsoftware.com>*

*Date: Thu, 24 May 2007 17:44:42*

*Subject: Re: Ada Interfaces and the Liskov Substitution Principle*

*Newsgroups: comp.lang.ada*

Correct, that's what I meant. When Ada has a choice between non-limited and limited, it makes you write "limited" if you meant that. That's annoying in this case, but it's less annoying than having:

```
type A is tagged record ...
type B is tagged limited record ...
type C is tagged private;
type D is tagged limited private;
type E is not limited interface;
type F is interface;
```

which would be forever confusing.

*From: Randy Brukardt <randy@rrsoftware.com>*

*Date: Wed, 23 May 2007 16:58:05*

*Subject: Re: Ada Interfaces and the Liskov Substitution Principle*

*Newsgroups: comp.lang.ada*

> My humble opinion: assignment of class-wide types should be prohibited in the first place. I mean — at compile time. It's just asking for troubles.

That doesn't make much sense. Yes, it is a bit weird that assignment is potentially a dispatching operation, but otherwise there is no semantic problem with it. (The OP having an illegal example, thus proving nothing.) It might be a bit hard to use usefully, but that's no reason for banning it.

*From: Stefan Lucks <lucks@th.informatik.uni-mannheim.de>*

*Date: Thu, 24 May 2007 13:12:56*

*Subject: Re: Ada Interfaces and the Liskov Substitution Principle*

*Newsgroups: comp.lang.ada*

> [...] "constant", "in", "out", "not null", etc. All these are examples of non-LSP subtypes. [...] mere passing a variable as "in" does it as well in the sense that "in T" is not an LSP-subtype of T.

You are using a very broad and generalised interpretation of the LSP. My interpretation — and I believe this is the common and usual one — is that "X: in T" in the parameterlist of a subprogram does not deal with some "artificial" type "in T", just with "T". The "in" is part of the subprogram's contract, not a part of X's contract. So there is no conflict with LSP.

> LPS is totally irrelevant as long as substitutability violation can be detected at compile time. This is why "constant" does not worry anybody. A method disallowing is perfectly OK, if you cannot call it.

So your very broad and generalised interpretation of the LSP is totally irrelevant, except for the special case where it overlaps with the more narrow usual interpretation. Perhaps you should follow the crowd and narrow your interpretation as well?

Whatever interpretation, the stuff below is right.

> LSP violation becomes a problem when substitutability is indeterminable until run-time. In many cases we still choose to live with that. Constrained Ada subtypes is an example of. Another is multi-methods Foo (X, Y : T), when called on different children of T. In such cases Ada adds Constraint\_Error to the interface of each subprogram and things become "substitutable" again.

Yes, that is an ugly patch. But it appears tricky to come up with a better solution ...

> LSP violation is catastrophic when undetected. [...]

It is bad enough if detected after lengthy testing and debugging sessions.

# Conference Calendar

This is a list of European and large, worldwide events that may be of interest to the Ada community. Further information on items marked ♦ is available in the Forthcoming Events section of the Journal. Items in larger font denote events with specific Ada focus. Items marked with ☺ denote events with close relation to Ada.

The information in this section is extracted from the on-line *Conference announcements for the international Ada community* at: <http://www.cs.kuleuven.be/~dirk/ada-belgium/events/list.html> on the Ada-Belgium Web site. These pages contain full announcements, calls for papers, calls for participation, programs, URLs, etc. and are updated regularly.

---

## 2007

- ☺ July 01-02     **12<sup>th</sup> International Workshop on Formal Methods for Industrial Critical Systems (FMICS'2007)**, Berlin, Germany. Affiliated with CAV'2007. Topics include: Design, specification, code generation and testing with formal methods; Verification and validation of complex, distributed, real-time systems and embedded systems; Verification and validation methods that aim at circumventing shortcomings of existing methods with respect to their industrial applicability; Tools for the design and development of formal descriptions; Case studies and project reports on formal methods related projects with industrial participation (e.g. safety critical systems, mobile systems, object-based distributed systems); Application of formal methods in standardization and industrial forums.
- July 02-06     **6<sup>th</sup> International Conference on Integrated Formal Methods (IFM'2007)**, Oxford, UK.
- ☺ July 03     **Ada-Deutschland Tagung – Sichere Software - Entwicklung und Bewertung für eingebettete Systeme**, Karlsruhe, Germany
- July 03-05     **20<sup>th</sup> Conference on Software Engineering Education and Training (CSEET'2007)**, Dublin, Ireland. Deadline for submissions: November 1, 2006 (abstracts), November 10, 2007 (workshops, panels)
- ☺ July 05-07     **6<sup>th</sup> International Symposium on Parallel and Distributed Computing (ISPDC'2007)**, Hagenberg, Austria. Topics include: Parallel Computing; Algorithms, Models and Formal Verification; Tools and Environments for Program Analysis; Task and Communication Scheduling and Load Balancing; Real-time Systems; Distributed Software Components; Real-time Distributed Systems; Security; Fault Tolerance; Applications and Case Studies; etc.
- July 08-11     **8<sup>th</sup> Conference on Communicating Process Architectures (CPA'2007)**, Surrey, Guildford, UK. Topics include: Verification and analysis of concurrent systems; Tools and languages for hardware-software co-design; Programming languages and environments for concurrent systems; Environments for distributed applications, such as grid applications; Programming and implementation issues for concurrent languages, such as deadlock-freedom by design, starvation, ...; System issues for programming languages supporting concurrency, such as multithreading kernels and interrupt architectures; Applications that exploit, or rely on, concurrency; etc.
- July 08-15     **34<sup>th</sup> International Colloquium on Automata, Languages and Programming (ICALP'2007)**, Wroclaw, Poland. Topics include: Principles of Programming Languages, Formal Methods, Models of Concurrent and Distributed Systems, Program Analysis and Transformation, Specification, Refinement and Verification, etc.
- July 09-11     *AdaCore – Public Ada and GNAT Courses*, New York City, NY, USA. Topic: "Ada 2005 for Ada 95 Programmers"
- ☺ July 09-12     **2007 International Conference on Software Engineering Theory and Practice (SETP'2007)**, Orlando, FL, USA. Topics include: all areas of Software Engineering and all related areas, such as: Component-based software engineering; Critical and embedded software design; Distributed and parallel systems; Distribution and parallelism; Education (software engineering curriculum design); Embedded and real-time software; Empirical software engineering and metrics; Evolution and maintenance; High assurance software systems; Interoperability; Legal issues and standards; Object-oriented techniques; Program understanding issues; Programming languages; Quality management; Real-time software engineering; Reliability; Reverse engineering and software maintenance; Software architectures and design; Software components and reuse; Software cost estimation techniques; Software design and design patterns; Software engineering methodologies; Software engineering versus systems engineering; Software policy and ethics; Software reuse; Software safety and reliability;

Software security; Software testing, evaluation and analysis technologies; Software tools and development environments; Survivable systems; Technology adoption; Verification, validation and quality assurance; etc.

- July 11-14      12<sup>th</sup> IEEE **International Conference on the Engineering of Complex Computer Systems (ICECCS'2007)**, Auckland, New Zealand. Topics include: Avionics and Automobile Software; Formal Methods and Approaches to Manage and Control Complex Systems; Interoperability and Standardization; Real-time and Embedded Systems; Software Architecture and System Engineering; Systems and Software Safety and Security; Tools, Environments, and Languages for Complex Systems; Verification Techniques for Complex Software Systems; etc.
- July 14            2<sup>nd</sup> **Workshop on UML and AADL (UML&AADL'2007)**. Topics include: all aspects of the representation, analysis, and implementation of DRE (distributed, real-time and embedded systems) behaviour and/or architecture models.
- ☺ July 22-25      2<sup>nd</sup> **International Conference on Software and Data Technologies (ICSOFT'2007)**, Barcelona, Spain. In conjunction with ENASE'2007. Topics include: Programming Languages (Object-Oriented Programming, Languages and compilers, ...); Software Engineering (Reliable software technologies, Dependable computing, Software components, Software maintenance, Real-time software, Software economics, ...); Distributed and Parallel Systems; etc.
- July 22            2<sup>nd</sup> **International Workshop on Metamodelling Utilization in Software Engineering (MUSE'2007)**
- July 23-25      2<sup>nd</sup> **International Working Conference on Evaluation of Novel Approaches to Software Engineering (ENASE'2007)**, Barcelona, Spain. In conjunction with ICSOFT'2007. Topics include: Model driven engineering; Software components and component-based software engineering; Generative software development; Evolutionary design; New methodologies, practices, architectures, technologies, tools, metrics; etc.
- July 23-26      *Objektum - Public Ada 95 Training Course*, London, UK. Topics include: the concepts and notation of Ada 95; promotes the concept of reusable software components and encourages a style of programming, which enforces a disciplined approach to software engineering; for those who have had little or no exposure to developing Ada programs and who want to master this elegant language.
- ☺ July 30 – Aug 03    21<sup>st</sup> European **Conference on Object-Oriented Programming (ECOOP'2007)**, Berlin, Germany. Topics include: all areas relevant to object technology.
- ☺ July 30            11<sup>th</sup> **Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts**. Topics include: successfully used exercises, examples, and metaphors; approaches and tools for teaching (basic) object-oriented concepts; teaching refactoring and/or design patterns; misconceptions related to object technology; etc.
- ☺ July 30            17<sup>th</sup> **Doctoral Symposium and PhD Students Workshop**. Topics include: Design Patterns; Components, Modularity; Concurrency, Real-time, Embeddedness, Distribution; Domain Specific Languages, Language Workbenches; Adaptability; Generative Programming; Language Design, Language Constructs, Static Analysis; Language Implementation; Model Engineering, Design Languages; Software Evolution, Versioning; Formal methods; Tools, Programming environments; etc.
- ☺ July 30            2<sup>nd</sup> **Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS'2007)**. Topics include: implementation of fundamental OOL features: inheritance (object layout, late binding, subtype test, ...), genericity (parametric types), memory management; runtime systems: compilers, linkers, etc; optimizations: static and dynamic analyses, etc; resource constraints: real-time systems, embedded systems; relevant choices and tradeoffs: separate compilation vs. global compilation, dynamic checking vs. proof-carrying code, etc.
- August 12-15      26<sup>th</sup> Annual ACM **SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC'2007)**, Portland, Oregon, USA

- ☺ August 21-24 13<sup>th</sup> IEEE **International Conference on Embedded and Real-Time Computing Systems and Applications** (RTCSA'2007), Daegu, South Korea. Topics include: Real-Time Systems (Scheduling, Fault-tolerance, Programming languages and run-time systems, Middleware systems, Design and analysis tools, Formal methods, Case studies, Applications, etc.); Embedded Systems (Scheduling, HW/SW co-design, Embedded system design practices, etc.); etc.
- August 25-31 2<sup>nd</sup> **International Conference on Software Engineering Advances** (ICSEA'2007), Cap Esterel, French Riviera, France. Topics include: Advances in fundamentals for software development; Advanced mechanisms for software development; Advanced design tools for developing software; Open source software; Software deployment and maintenance; Software economics, adoption, and education; etc.
- ☺ August 28-31 13<sup>th</sup> **International Conference on Parallel and Distributed Computing** (Euro-Par'2007), Rennes, France. Topics include: the promotion and advancement of all aspects of parallel and distributed computing, such as support tools and environments, distributed systems, parallel and distributed programming, etc.
- ☺ August 28 **Workshop on Highly Parallel Processing on a Chip** (HPPC'2007). Topics include: (parallel) programming paradigms, languages, libraries, and support tools for efficient and manageable exploitation of highly parallel multi-core architectures; etc.
- ☺ Aug 28 – Sept 01 5<sup>th</sup> **International Symposium on Parallel and Distributed Processing and Applications** (ISPA'2007), Niagara Falls, Ontario, Canada. Topics include: Tools and environments for software development; Distributed systems and applications; Reliability, fault-tolerance, and security; High-performance scientific and engineering computing; etc.
- ☺ Aug 28 – Sept 01 **International Workshop on Multicore and Hybrid Systems for Numerically Intensive Computations** (MHSN'2007), Niagara Falls, Ontario, Canada. In conjunction with The 5th International Symposium on Parallel and Distributed Processing and Applications (ISPA'2007). Topics include: parallel programming models, compiler technology, runtime systems and libraries, etc.
- ☺ September 03-07 9<sup>th</sup> **International Conference on Parallel Computing Technologies** (PaCT'2007), Pereslavl-Zalessky, Russia. Topics include: New trends and models in Parallel Programming; All aspects of the applications of parallel computer systems; Languages, environment and software tools supporting parallel processing; General architecture concepts, enabling technologies; Teaching parallel processing; etc.
- September 03-07 Joint 11<sup>th</sup> **European Software Engineering Conference** and 15<sup>th</sup> **ACM SIGSOFT Symposium on the Foundations of Software Engineering** (ESEC/FSE'2007), Dubrovnik, Croatia. Topics include: Specification and Verification; Software Architecture and Design; Distributed Systems and Middleware; Open Standards and Certification; Software Quality and Performance; Case Studies and Experience Reports; Dependability (safety, security, reliability); Component-based SW Engineering; Tools and Environments; Evolution and Refactoring; etc.
- Sept 03-04 9<sup>th</sup> **International Workshop on Principles of Software Evolution** (IWPSE'2007). Topics include: software reliability and evolution, empirical studies related to software evolution, methodologies and environments for evolutionary design and development, experience reports and lessons learned, etc.
- September 04 2<sup>nd</sup> **International Workshop on Engineering Fault Tolerant Systems** (EFTS'2007). Topics include: Software architecture and fault tolerance; OO frameworks and design patterns for fault tolerance; Fault-tolerant software development processes; Error recovery through exception handling in the software life-cycle; Design and implementation of fault-tolerant distributed systems; Rigorous approach to fault tolerance programming; Verification and validation of fault-tolerant systems; etc.
- ☺ September 04-07 **International Conference on Parallel Computing** 2007 (ParCo2007), Juelich & Aachen, Germany. Topics include: all aspects of parallel computing, including applications, hardware and software technologies as well as languages and development environments. Deadline for submissions: July 31, 2007 (full papers)
- September 04-07 18<sup>th</sup> **International Conference on Concurrency Theory** (CONCUR'2007), Lisbon, Portugal. Topics include: all areas of semantics, logics, and verification techniques for concurrent systems, related verification techniques and tools, related programming models, etc.

- September 10-12 7<sup>th</sup> **International Workshop on Automated Verification of Critical Systems (AVoCS'2007)**, Oxford, UK. Topics include: tools and techniques for the verification of critical systems; such as automated verification, including model checking, theorem proving, abstract interpretation, and refinement pertaining to various types of critical systems (safety-critical, security-critical, business-critical, performance-critical, ...)
- ☉ September 10-14 5<sup>th</sup> IEEE **International Conference on Software Engineering and Formal Methods (SEFM'2007)**, London, UK. The aim is to advance the state of the art in formal methods, to scale up their application in software industry and to encourage their integration with practical engineering methods. Topics include: software specification, validation and verification; programming languages and type theory; program analysis; fault-tolerant computing; embedded systems; real-time and hybrid systems theory; software architectures and their description languages; CASE tools and tool integration; applications of formal methods and industrial case studies; etc.
- ☉ September 15-19 16<sup>th</sup> **International Conference on Parallel Architectures and Compilation Techniques (PaCT'2007)**, Brasov, Romania. Topics include: Compilers and tools for parallel computer systems; Support for correctness in hardware and software (esp. with concurrency); Parallel programming languages, algorithms and applications; Middleware and run time system support for parallel computing; High performance application specific systems; etc.
- ☉ September 18-21 26<sup>th</sup> **International Conference on Computer Safety, Reliability and Security (Safecomp'2007)**, Nuremberg, Germany
- ☉ September 18-21 **Embedded Systems Conference 2007**, Boston, USA
- September 20-21 1<sup>st</sup> **International Symposium on Empirical Software Engineering and Measurement (ESEM'2007)**, Madrid, Spain. Incorporating ISESE and Metrics. Topics include: Evaluation and comparison of techniques and models; Reports on the benefits derived from using certain technologies; Empirically-based decision making; Industrial experience in process improvement; Quality measurement and assurance; Evidence-based software engineering; Effort and cost estimation, defect rate and reliability prediction; etc.
- ♦ September 25 **Ada UK Conference 2007**, Manchester, UK. Theme: "Building better, safer software". This UK-based Ada conference is being organised to promote awareness of the Ada 2005 language revision, and to highlight the increased relevance of Ada in safety-critical programming.
- ☉ September 26-28 3<sup>rd</sup> Latin-American **Symposium on Dependable Computing (LADC'2007)**, Morelia, Mexico. Topics include: Dependability Modeling, Prediction and Evaluation; Dependable Applications; Distributed Systems; Parallel, Clustered and Grid Systems; Real-Time and Embedded Systems; Safety-Critical Systems; Security of Computing Systems; Software Engineering of Dependable Systems; Software Reliability; Software Testing, Validation and Verification; Survivability of Computing Systems; etc.
- Sept 30 – Oct 01 7<sup>th</sup> IEEE **International Working Conference on Source Code Analysis and Manipulation (SCAM'2007)**, Paris, France. Co-located with ICSM'2007
- Sept 30 – Oct 05 10<sup>th</sup> **International Conference on Model-Driven Engineering Languages and Systems (MoDELS'2007)**, Nashville, TN, USA. Topics include: advanced techniques for expressing design intent at a higher level of abstraction, keeping abstract models of complex software systems in sync with the underlying code, etc.
- October 02-05 23<sup>rd</sup> IEEE **International Conference on Software Maintenance (ICSM'2007)**, Paris, France. Topics include: software and systems maintenance, evolution, and management.
- October 11-12 7<sup>th</sup> **International Conference on Quality Software (QSIC'2007)**, Portland, Oregon, USA. Topics include: Software quality (review, inspection and walkthrough, reliability, safety and security, ...); Evaluation of software products and components (static and dynamic analysis, validation and verification); Information and knowledge management (economics of software quality, ...); Formal methods (program analysis, model checking, ...); Applications (component-based systems, distributed systems, embedded systems, safety critical systems, ...); etc.

- October 11-12 12<sup>th</sup> Nordic **Workshop on Secure IT Systems** (NordSec'2007), Reykjavik, Iceland. Topics include: Language-based Techniques for Security, Security Education and Training, Trust and Trust Management, etc. Deadline for submissions: July 23, 2007
- ☺ October 15-17 1<sup>st</sup> **Workshop on Advances in Programming Languages** (WAPL'2007), Wisla, Poland. Within the framework of International Multiconference on Computer Science and Information Technology (IMCSIT). Topics include: Compiling techniques; Domain-specific languages; Formal semantics and syntax; Generative and generic programming; Languages and tools for trustworthy computing; Language concepts, design and implementation; Metamodeling and modeling languages; Model-driven engineering languages and systems; Practical experiences with programming languages; Program analysis, optimization and verification; Program generation and transformation; Programming tools and environments; Proof theory for programs; Specification languages; Type systems; etc
- ☺ October 16 **International Workshop on Real-Time Software** (RTS'2007), Wisla, Poland. Within the framework of the International Multiconference on Computer Science and Information Technology (IMCSIT). Topics include: real-time system development, real-time scheduling, safety, reliability, dependability, fault-tolerance, standards and certification, software development tools, model-based development, automatic code generation, real-time systems curricula, etc.
- ☺ October 18 4<sup>th</sup> **Workshop on Programming Languages and Operating Systems** (PLOS'2007), Stevenson, WA, USA. Topics include: critical evaluations of new programming language ideas in support of OS construction; type-safe languages for operating systems; language-based approaches to crosscutting system concerns, such as security and run-time performance; language support for system verification; etc.
- ☺ October 21-25 22<sup>nd</sup> Annual **Conference on Object-Oriented Programming, Systems, Languages, and Applications** (OOPSLA'2007), Montreal, Canada. Topics include: programmer productivity, secure and reliable software, changing hardware platforms, ultra-large scale systems, improve programming languages, refine the practice of software development, etc.
- ☺ October 21 6<sup>th</sup> "**Killer Examples**" **Workshop**. Theme: "Process in OO Pedagogy" Topics include: "killer" examples of teaching the process of programming; "killer" examples of teaching OO modeling and programming; analyses of the process which students use to solve problems; what are needs of industry - what will keep students competitive in the workplace?; etc. Deadline for submissions: September 3, 2007. Deadline for early registration: September 11, 2007
- October 21 3<sup>rd</sup> **Workshop on Library Centric Software Design** (LCSD'2007). Topics include: Design and implementation of libraries; Program and system design based on libraries; Evolution, refactoring, and maintenance of libraries; Design of language facilities and tools in support of library definition and use; Validation, debugging, and testing of libraries; Extensibility, parameterization, and customization; Specification of libraries and their semantics; Assessing quality of libraries; Using several libraries in combination; etc. Deadline for paper submissions: August 1, 2007
- ☺ October 22 2<sup>nd</sup> **International Conference on System Safety** 2007, London, UK. Includes: talk on "Certification of Object Oriented Programs", by Robert Dewar, AdaCore
- October 26 1<sup>st</sup> York **Doctoral Symposium on Computing** (YDS'2007), York, UK. Topics include: High integrity system engineering, within the context of Formal methods, Verification and Formal verification, Theorem proving, Model checking, Testing; Information systems, within the context of Formal Methods in Software Engineering, Model Driven Development, Object-Oriented Modelling and Development, Systems Engineering Methodologies, Modelling Formalisms (Languages and Notations), CASE Tools for System Development, Security, Component-Based Development, Software Architecture, Software Engineering for Concurrent and Distributed Systems, Software Quality, Software Verification (Validation and Inspection), ...; IT Security; Programming languages and systems; Real-time systems; etc. Deadline for submissions: July 30, 2007 (abstracts), August 6, 2007 (papers)
- ☺ October 30-31 4<sup>th</sup> **Workshop on Object-oriented Modeling of Embedded Real-Time Systems** (OMER-4), Paderborn, Germany. Topics include: Architectures/frameworks for platform independent, reusable software components; Formal verification at the model and code level; Software components as products; Software quality; Standards and guidelines (e.g., AUTOSAR, IEC 61508, MISRA, UML, ...); Respective trends in automotive software development; etc.



- ◆ Nov 04-08 2007 ACM **SIGAda Annual International Conference** (SIGAda'2007), Washington, DC, USA. Sponsored by ACM SIGAda (ACM approval pending), in cooperation with SIGAPP, SIGCAS, SIGCSE, SIGPLAN, SIGSOFT, Ada-Europe, and Ada Resource Association (Cooperation approvals pending). Topics include: Safety, security and high integrity development issues; Language selection for a high reliability system; Use of ASIS for new Ada tool development; Mixed-language development; High reliability software engineering education; High reliability development experience reports; Static and dynamic code analysis; Use of new Ada 2005 features/capabilities; etc.
- November 05-09 18<sup>th</sup> IEEE **International Symposium on Software Reliability Engineering** (ISSRE'2007), Trollhaettan, Sweden. Topics include: Reliability, availability and safety of software systems; Quality/reliability-related security issues; Verification and validation; Industrial best practices; Empirical studies of those topics; etc.
- November 07-09 6<sup>th</sup> **International Conference on Software Methodologies, Tools, and Techniques** (SoMeT'2007), Rome, Italy. Topics include: Software methodologies, and tools for robust, reliable, non-fragile software design; Automatic software generation versus reuse, and legacy systems, source code analysis and manipulation; Intelligent software systems design, and software evolution techniques; Software optimization and formal methods for software design; Software security tools and techniques, and related Software Engineering models; End-user programming environment; Software Engineering models, and formal techniques for software representation, software testing and validation; etc.
- November 13-15 9<sup>th</sup> **International Conference on Formal Engineering Methods** (ICFEM'2007), Boca Raton, Florida, USA. Topics include: Abstraction and refinement; Tool development and integration for system design and verification; Techniques for specification, verification and validation; Techniques and case studies for correctness by construction; Applications in real-time, hybrid and critical systems; Development methodologies with their formal foundations; etc.
- November 14-16 10<sup>th</sup> IEEE **International Symposium on High Assurance Systems Engineering** (HASE'2007), Dallas, Texas. Topics include: Design and development of highly reliable, survivable, secure, safe, and time-assured systems; Policies for reliability, safety, security, integrity, privacy, and confidentiality of high assurance systems; Formal specification, specification validation, testing, and model checking for high assurance systems; High assurance software architectures and design; Case studies, experiments and tools for high assurance systems; etc. Deadline for submissions: July 9, 2007 (panels, fast abstracts)
- ☉ November 25-30 9<sup>th</sup> **International Symposium on Distributed Objects and Applications** (DOA'2007), Vilamoura, Algarve, Portugal. Topics include: Application case studies of distribution technologies; Interoperability with other technologies; Reliability, fault tolerance, quality-of-service, and real-time support; Scalability and adaptivity of distributed architectures; etc. Deadline for registration: September 10, 2007
- Nov 29 – Dec 01 5<sup>th</sup> **Asian Symposium on Programming Languages and Systems** (APLAS'2007), Singapore. Topics include: foundational and practical issues in programming languages and systems, such as semantics, type systems, language design, program analysis, optimization, software security, safety, verification, compiler systems, programming tools and environments, etc.
- ☉ December 03-06 28<sup>th</sup> IEEE **Real-Time Systems Symposium** (RTSS'2007), Tucson, Arizona, USA. Topics include: all aspects of real-time systems design, analysis, implementation, evaluation, and case-studies.
- ☉ December 03-06 8<sup>th</sup> **International Conference on Parallel and Distributed Computing, Applications, and Techniques** (PDCAT'2007), Adelaide, Australia. Topics include: Formal methods and programming languages, Software tools and environments, Component-based and OO Technology, Parallel/distributed algorithms, Task mapping and job scheduling, High-performance scientific computing, etc.
- December 03-12 3<sup>rd</sup> **International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering** (CISSE'2007), Internet. Include 4 e-conferences, among others the International Conference on Systems, Computing Sciences and Software Engineering (SCSS'2007) with topics: Programming Models and tools, Parallel and Distributed processing, Modeling and Simulation, Embedded Systems and Applications, Programming Languages, Object Based Software Engineering,

Parallel and Distributed Computing, Real-Time Systems, Multiprocessing, etc. Deadline for submissions: October 5, 2007

December 10 Birthday of Lady Ada Lovelace, born in 1815. Happy Programmers' Day!

---

## 2008

- ☺ January 10-12 35<sup>th</sup> Annual ACM **SIGPLAN-SIGACT Symposium on Principles of Programming Languages** (POPL'2008), San Francisco, California, USA. Topics include: fundamental principles and important innovations in the design, definition, analysis, transformation, implementation and verification of programming languages, programming systems, and programming abstractions. Deadline for submissions: July 16, 2007 (papers)
- ☺ March 04-07 **CISIS2008 - International Workshop on Multi-Core Computing Systems** (MuCoCoS'2008), Barcelona, Spain. Topics include: programming languages and models; performance modelling and evaluation of multi-core systems; tool-support for multi-core systems; compilers, runtime and operating systems; etc. Deadline for paper submissions: October 10, 2007. Deadline for early registration: December 15, 2007
- ☺ March 12-14 **SIAM Conference on Parallel Processing for Scientific Computing** (PP'2008), Atlanta, Georgia, USA. Topics include: Programming languages, models, and compilation techniques; The transition to ubiquitous multicore/manycore processors; Tools for software development and performance evaluation; Parallel computing in industry; Distributed/grid computing; Fault tolerance; etc. Deadline for submissions: August 14, 2007 (mini-symposium proposals), September 11, 2007 (abstracts)
- ☺ March 12-15 39<sup>th</sup> ACM **Technical Symposium on Computer Science Education** (SIGCSE'2008), Portland, Oregon, USA. Visit the ACM SIGAda booth!
- March 16-20 23<sup>rd</sup> ACM **Symposium on Applied Computing** (SAC'2008), Fortaleza, Ceara, Brasil.
- ☺ Mar 16-20 *Track on Object-Oriented Programming Languages and Systems* (OOPS'2008). Topics include: Design and implementation of novel abstractions, constructs and mechanisms; Multi-paradigm features; Language features in support of adaptability; Component-based programming; Generative programming; Program structuring, modularity; Distributed objects and concurrency; Middleware; Compilation techniques; etc. Deadline for submissions: September 8, 2007
- Mar 16-20 *Track on Software Verification*. Topics include: Data flow analysis, control flow analysis, type effect systems, constraint systems and abstract interpretation techniques for verification; Techniques to validate system software (such as compilers) as well as assembly code or bytecode; Software certification and proof carrying code; Integration of formal verification into software development projects; etc. Deadline for full paper submissions: September 8, 2007
- Mar 16-20 *Track on Software Engineering* (SE'2008). Topics include: Component-Based Development and Reuse; Dependability and Reliability; Fault Tolerance and Availability; Maintenance and Reverse Engineering; Verification, Validation, Testing, and Analysis; Formal Methods and Theories; Empirical Studies, Benchmarking, and Industrial Best Practices; Applications and Tools; Distributed, Embedded, Real-Time, High Performance, Highly Dependable Systems; etc. Deadline for paper submissions: September 8, 2007
- Mar 29 – Apr 06 European **Joint Conferences on Theory and Practice of Software** (ETAPS'2008), Budapest, Hungary. Deadline for submissions: October 5, 2007 (research and tool paper abstracts), October 12, 2007 (research and tool papers)
- ☺ April 02-04 3<sup>rd</sup> European **Conference on Computer Systems** (EuroSys'2008), Glasgow, UK. Topics include: All areas of operating systems and distributed systems; Systems aspects of: Dependable computing, Parallel and concurrent computing, Distributed algorithms, Programming language support, Real-time and embedded computing, Security, ...; Experience with existing systems; Reproduction or refutation of previous results; Negative results; Early ideas. Deadline for submissions: September 14, 2007 (abstracts), September 21, 2007 (papers)

- May 07-09 7<sup>th</sup> European **Dependable Computing Conference** (EDCC-7), Kaunas, Lithuania. Topics include: Architectures for dependable systems; Fault tolerant distributed systems; Fault tolerance in real-time systems; Hardware and software testing, verification, and validation; Formal methods for dependability; Safety-critical systems; Software reliability engineering; Software engineering for dependability; etc. Deadline for submissions: September 20, 2007
- ◆ June 16-20 13<sup>th</sup> **International Conference on Reliable Software Technologies** (Ada-Europe'2008). Venice, Italy.
- June 30 – July 02 13<sup>th</sup> Annual **Conference on Innovation and Technology in Computer Science Education** (ITiCSE'2008), Madrid, Spain
- December 10 Birthday of Lady Ada Lovelace, born in 1815. Happy Programmers' Day!

## Ada Conference UK 2007

### Building better, safer software

25 September 2007 — Hilton Deansgate Hotel, Manchester, UK

### Ada answers.

As the need for robust and reliable software systems increases, Ada continues to prove itself as the answer for many of today's most complex programming challenges – especially in the areas of real-time, embedded and safety-critical applications.

**Event focus:** to promote awareness of the **Ada language**, and to highlight the increased relevance of Ada in safety- and security-critical programming.

**Event outline:** Plenary session by **Robert Dewar**, *plus* a series of technical talks by leading industrial experts, *plus* a stream of vendor talks running in parallel; *also* a broad range of leading Ada product vendors will be present in the *exhibition* atrium. Full details of the event and programme will be posted at:

[www.ada-uk-conference.co.uk](http://www.ada-uk-conference.co.uk)

### Event opportunities:

**Meet** members and colleagues from all sectors of the Ada community; this high-calibre event is expected to attract many professional Ada users and reinforce links between all sections of the Ada community.

**Examine** an extensive range of technologies from the leading Ada toolset and service vendors.

**Learn** about the latest issues and developments of relevance to Ada users and vendors, providers and practitioners, specialists and suppliers, designers and developers – topics such as:




- the latest revision of the Ada programming language and its adoption in major industrial applications
- ongoing work on the upcoming DO-178C standard
- the advantages of Ada for real-time programming

**Discuss** innovative and challenging experiences with the Ada language.

**Event location:** the conference venue is in the heart of Manchester, at the spectacular Deansgate Hilton Hotel, with excellent access by road, rail and air.

**Registration and enquiries:** Joan.Atkinson@ncl.ac.uk +44 191 221 2222

**The not-to-be missed Ada event of 2007!!**

<p>Event lead sponsor and advocate: <a href="http://www.adacore.com">www.adacore.com</a></p> 	<p>Event sponsors:</p> <p><a href="http://www.windriver.com">www.windriver.com</a>      <a href="http://www.basesystems.co.uk">www.basesystems.co.uk</a>      <a href="http://www.mbda.co.uk">www.mbda.co.uk</a></p> <p><small>QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.</small></p>  		
--	---	--	--



Event operated by CSR, in cooperation with the Safety-Critical Systems Club: [www.safety-club.org.uk](http://www.safety-club.org.uk)



# 13<sup>th</sup> International Real-Time Ada Workshop (IRTAW-13)

*Juan Antonio de la Puente*

*Universidad Politécnica de Madrid (UPM), E-28040 Madrid.; Tel: +34 913367342; email: jpunte@dit.upm.es*

## Abstract

*The 13<sup>th</sup> International Real-Time Ada Workshop was held in Vermont, USA. The main focus was on analysing the impact of the new Ada 2005 standard on the development of real-time systems, and the future of the language in this important domain. The workshop was very successful in identifying new developments and exploring new application areas. The delegates also had an enjoyable time thanks to the efficient work of the local organizer Ben Brosgol.*

*Keywords: Real-time systems, Ada 2005.*

## 1 Introduction

The 13<sup>th</sup> International Real-Time Ada Workshop was held between the 17<sup>th</sup> and the 19<sup>th</sup> of April, 2007, at the impressive Woodstock Inn, in Woodstock, Vermont, USA (figure 1). Woodstock is a mountain resort with a beautiful view of forests and snow-covered hills that was the first ski station with mechanical tows to be operated in the USA (1934). The town has several covered bridges (figure 2), which are one of the distinctive features of Vermont. In spite of the advanced time of the year, we enjoyed plenty of snow, even a little too much for the trip from Boston to Woodstock on the first day, under the last winter storm of this year.

The local organization by Ben Brosgol was excellent, and there was plenty of time for discussions and informal conversation.



**Figure 1** View of Woodstock Inn

The Program Committee accepted eleven papers as a basis for discussion, which are being published as part of the

official Proceedings of the Workshop [1]. There were eighteen participants, coming from Europe (Spain, UK, Italy, and France) and North America (USA and Canada). As in previous IRTAW meetings, all the attendants took active part in the technical discussions which were at the core of the workshop. The main points of the discussions and the overall conclusions are summarized in the rest of this report.

## 2 Technical program

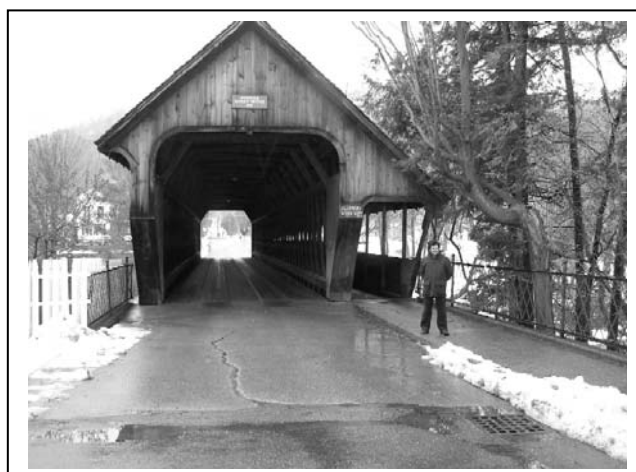
The technical program was organized into five technical sessions and a conclusion session. Each session had a chair person and a rapporteur, who was in charge of writing a report of the session including the agreements reached. The sessions were organized into slots of 3½ hours, with a long lunch break according to the IRTAW tradition.

The contents of the technical sessions are summarized in the next sections.

### 2.1 Language issues

The first session focused on open issues related to the real-time aspects of the Ada 2005 standard. Three main points for discussion were identified:

- EDF semantics;
- Requeuing via interfaces;
- distributed extensions to Ravenscar



**Figure 2** Covered bridge

*Correcting the EDF definition in the Ada 2005 LRM*

The first part of the session dealt with a problem related to the definition of priority inheritance in the Ada 2005 language reference manual (ALRM) [2], which was

intended to be the same as Ted Baker's Stack Resource Protocol (SRP) [3]. Alan Burns proposed a rewording of the current definition that eliminates the problem [4]. The workshop agreed that an Ada Issue (AI) should be addressed to the Ada Rapporteur Group (ARG).

#### *Requeuing via interfaces*

The second part of the discussion was centred on the possibility of allowing requeuing on procedures defined in synchronized or protected interfaces, and possibly, but not necessarily, implemented as protected entries. This possibility raises some semantics problems [5], which have to be solved in some way which is consistent with the current definition of requeue to an entry. The participants agreed that this is a useful feature, even in a restricted form (e.g. according to a static scheme). Consequently, a proposal to include requeue to synchronized or protected procedures will be put forward for discussion at the ARG.

#### *Distributed extensions to the Ravenscar profile*

The next discussion addressed a proposal to complement the Ravenscar profile (ALRM D13.1) with a restricted form of the Distributed Systems Annex (ALRM E) [6]. Some restrictions were proposed on DSA in order to keep with the Ravenscar profile motivation for predictable behaviour and simplicity of implementation. The workshop agreed that here is a need to build high-integrity distributed systems in Ada, and that more work is required on this topic.

## **2.2 Programming patterns and libraries**

The second session addressed some proposals of architectural frameworks and coding patterns for Ada 2005:

#### *Component framework for real-time utilities*

The first part of the session was focused on a proposal by Andy Wellings to agree on a component framework for real-time utilities, much in the line of RT-Java [7].

#### *Programming patterns for servers*

The second topic for discussion was introduced by Alan Burns, who proposed a set of patterns for programming servers in Ada 2005 [8].

#### *Code patterns for model-driven development.*

The last proposal was introduced by Tullio Vardanega [9], who presented a framework for developing high-integrity systems in Ada 2005 using a design paradigm based on a four-view model-driven architecture.

After a general discussion on these proposals, the workshop agreed that there is interest in defining a common framework for component-oriented implementation of real-time systems. It was recognized that more effort is needed before a recommendation for standardization can be made, and a proposal to continue the work on this topic through a

series of meetings in the framework of the ARTIST network<sup>1</sup> was made by Alan Burns.

## **2.3 Implementation experience with Ada 2005**

The next session was devoted to discussing early implementation experiences with new real-time features, and reviewing the support provided by the new real-time mechanisms. Some time was also devoted to discussing some of the features which were proposed in previous workshops but have not been included in the new standard.

#### *Implementation experiences*

Two implementations of the new Ada 2005 real-time mechanisms were described first. Mario Aldea presented the work on implementing Ada 2005 services in MaRTE OS and GNAT [10]. The implementation is almost complete, except for the new task dispatching policies and the priority band scheduling mechanism. A GNAT version for MaRTE OS including the new features will shortly be distributed by Ada Core.

Juan Zamorano described an implementation of the new real-time services in the Open Ravenscar Kernel (ORK) on a LEON 2 computer board [11]. This implementation is integrated with the GNAT for LEON compiler, and is also complete except for the scheduling mechanisms.

The workshop was not aware of any other implementations of the new Ada 2005 real-time services.

#### *Discussion*

Most of the discussion was on the accuracy of the CPU clock and timer mechanisms. Some sources of inaccuracy were identified, e.g. related to interrupt processing time being charged to application tasks.

Significant overheads due to execution-time monitoring mechanisms were also found in some cases, especially in architectures with poor hardware timer support such as LEON 2.

The discussion continued with an account on the implementation of application-defined scheduling. This feature was not included in Ada 2005, but Michael González-Harbour presented an implementation that can be added to GNAT as an extension. The workshop agreed that this is an interesting feature to have.

Finally, there was a discussion on the possibility of adding execution-time timers to the Ravenscar profiles, as proposed by Juan Antonio de la Puente. Some potential problems were identified, which require further work on this topic to be done before a consensus can be reached.

## **2.4 Beyond Ada 2005**

The fourth session was aimed at discussing future directions in computing and their potential impact on Ada. Two main topics were addressed: stream-based parallelism and synchronous multiprocessing.

<sup>1</sup> [www.artist-embedded.org](http://www.artist-embedded.org)

*Stream-based parallelism*

Neil Audsley presented a view of massive parallel computer architectures based on the concept of *system on a chip* [12]. An example of an Ada programmable system, using a tasking profile almost identical to Ravenscar, was described. The main problem for the use of Ada with such a system is the lack of a mechanism for simultaneously releasing groups of tasks for reading protected data, which is not possible with the current protected object model, especially under the Ravenscar restrictions. The workshop agreed that further work is need on such an issue.

*Synchronous multiprocessing*

Andy Wellings started a discussion on how Ada can be used with the kind of synchronous multiprocessing (SMP) systems that are expected to be commonplace in the near future, including multicore architectures [13]. The current support for such architectures is very basic, and a number of potential issues were identified, especially related to the allocation of tasks to processors. The definition of task groups in Ada 2005 may be useful for defining affinities of task groups to processors, in addition to task affinity. A vivid discussion was held on these and other related subjects, and the workshop agreed that this indeed a very important area in which work should be continued and proposals should be put forward for subsequent workshops.

**2.5 Ada and other standards**

The last discussion session dealt with the relationship between the Ada standard and the POSIX and RT Java standards.

*POSIX Ada binding*

Stephen Michell introduced the discussion on the POSIX Ada binding, which has not been updated since 1998 [14]. Subsequent revisions of both the POSIX and Ada standards have led to minor inconsistencies and functionality mismatches. Stephen proposed to develop a new approach to interfacing Ada with operating systems through a set of library packages.

The workshop expressed some doubts about the suitability of putting a high amount of effort into this task. However, the participants agreed that a minimum update that would remove the current errors or inconsistencies would be useful.

*Real-time Java*

Ben Brosgol made a presentation of the current status of Real-Time Java and the efforts for developing a high-integrity version of RT Java. The discussion which followed highlighted a number of interesting RT Java features which may be useful in Ada, e.g. priority inheritance, garbage collection, or different kinds of physical memory. The workshop concluded that the Ada community should continue investigating such issues.

**3 Conclusions**

The meeting was considered successful by the participants. A number of issues have been solved, and many points of

interest for developing real-time systems in Ada have been identified and discussed. A final session was held that summarized the main conclusions of the workshop:

- An AI will be submitted to the ARG in order to correct the current definition of the EDF dispatching policy.
- A proposal will be made to the ARG to allow requeue to synchronized interface procedures. The details of the proposal need further investigation, especially with regard to the implementation cost.
- A meeting will be held in order to consolidate the proposals for a real-time component framework in Ada.
- The accuracy and cost of execution-time monitoring mechanisms needs further investigation.
- Further investigation on using executing-time monitoring mechanisms with the Ravenscar profile is encouraged,
- Research should continue on developing an extension of the Ravenscar profile for distributed systems.
- There is a need for enhanced support of multiprocessor systems and stream-based parallel systems in Ada. Research in this area is encouraged.
- A minimum update of the POSIX Ada bindings is to be made, in order to remove errors and inconsistencies.
- Work should continue on watching the progress in RT Java in order to identify potential enhancements to Ada.

**Next Workshop**

The participants agreed on planning a new IRTAW meeting for September 2008. Tullio Vardanega volunteered to organize it in Italy, and Neil Audsley volunteered to serve as Program Chair.

**Social program**

A reception and dinner was held at the Simon Pearce restaurant, in the neighbouring town of Quechee. The restaurant is located in an impressive old mill building, which currently also hosts a glass factory. The dinner was excellent and as usual provided a great occasion for strengthening the links among the participants and talking in an informal atmosphere which greatly contributed to the lively technical session discussions.

## Acknowledgments

Ben Brosgol did an outstanding job with the local organization. Juan Antonio de la Puente was Program Chair, and Ben Brosgol, Alan Burns, Michael Gonzalez Harbour, Stephen Michell, Javier Miranda, Luis Miguel Pinho, Jorge Real, José Ruiz, Tullio Vardanega, and Andy Wellings also served in the Program Committee.

The Workshop was sponsored by Ada-Europe, ARA, AdaCore, and the ARTIST network.

## References

- [1] *Proceedings of the 13<sup>th</sup> International Ada Real-Time Workshop* (2007). To be published in Ada Letters.
- [2] S.T. Taft, R.A. Duff, R.L. Brukaradt, E Plödereder, P. Leroy (eds) (2006). *Ada 2005 Reference Manual. Language and Standard Libraries*. International Standard ISO/IEC 8652/1995(E) with Technical Corrigendum 1 and Amendment 1. LNCS 4348, Springer-Verlag.
- [3] T.P. Baker (1991). *Stack-based scheduling of realtime processes*. *Real-Time Systems*, **3** (1).
- [4] A. Zerzelidis, A. Burns, A.J., Wellings (2007). *Correcting the EDF protocol in Ada 2005*. In *Proceedings of the 13<sup>th</sup> International Ada Real-Time Workshop*, Ada Letters.
- [5] A.J. Wellings, A. Burns (2007). Integrating OOP and Tasking - The missing requeue. In *Proceedings of the 13<sup>th</sup> International Ada Real-Time Workshop*, Ada Letters.
- [6] S. Urueña, J. Zamorano (2007). *Building High-Integrity Distributed System with Ravenscar Restrictions*. In *Proceedings of the 13<sup>th</sup> International Ada Real-Time Workshop*, Ada Letters.
- [7] A.J. Wellings, A. Burns (2007). *A Framework for Real-Time Utilities for Ada 2005*. In *Proceedings of the 13<sup>th</sup> International Ada Real-Time Workshop*, Ada Letters.
- [8] A. Burns, A.J. Wellings (2007). *Programming Execution-Time Servers in Ada 2005*. In *Proceedings of the 13<sup>th</sup> International Ada Real-Time Workshop*, Ada Letters.
- [9] J. Pulido, J.A. de la Puente, M. Bordin, T. Vardanega, J. Hugues (2007). *Ada 2005 Code Patterns for Metamodel-Based Code Generation*. In *Proceedings of the 13<sup>th</sup> International Ada Real-Time Workshop*, Ada Letters.
- [10] M. Aldea, M. González Harbour (2007). *Operating System Support for Execution Time Budgets for Thread Groups*. In *Proceedings of the 13<sup>th</sup> International Ada Real-Time Workshop*, Ada Letters.
- [11] S. Urueña, J. Pulido, J. Redondo, J. Zamorano (2007). *Implementing the New Ada 2005 Real-Time Features on a Bare Board Kernel*. In *Proceedings of the 13<sup>th</sup> International Ada Real-Time Workshop*, Ada Letters.
- [12] M. Ward and N.C. Audsley (2007). *Suggestions for Stream Based Parallel Systems in Ada*. In *Proceedings of the 13<sup>th</sup> International Ada Real-Time Workshop*, Ada Letters.
- [13] A.J. Wellings and A. Burns (2007). Beyond Ada2005: Allocating Tasks to Processors in SMP Systems. In *Proceedings of the 13<sup>th</sup> International Ada Real-Time Workshop*, Ada Letters.
- [14] S. Michell (2007). *Interfacing Ada to Operating Systems*. In *Proceedings of the 13<sup>th</sup> International Ada Real-Time Workshop*, Ada Letters.



# Template Driven Design in UML and Ada 95

*Wain Glaister*

*AgustaWestland, Yeovil, BA20 2YB, England; email: wain.glaister@agustawestland.com*

## Abstract

*Patterns have been developed in relation to software design for many years, but patterns are of limited use if we do not know how to apply them. If we allow patterns to be implemented each time they are used we will be overrun with many different versions that are all slightly different and cannot be reused. A template on the other hand is a case specific description or implementation of a pattern that can be directly reused.*

*The creation of templates is an important task and must be incorporated into a project at an early stage for its full potential to be realised. This paper describes a useful form of templates, the benefits which may be gained from their use, how they can be integrated with other project code and some of the ways in which they can be applied by examining a mission computer system for a future helicopter.*

## 1 Introduction

Avionic mission computers are large and complex systems dealing with interaction from multiple users and other avionic systems. Although not safety related, a mission computer is a mission critical, real-time, distributed system, which performs several concurrent operations. AgustaWestland is currently developing a mission computer for future helicopters and in order to achieve the goals of the system a modular and flexible design has been chosen.

This paper describes the philosophy behind the use of patterns, templates and foundation code in the AgustaWestland development of the mission computer system. It then provides a number of examples showing a selection of the templates that have been developed to date and how these templates can be applied.

## 2 Mission Avionic Computer

AgustaWestland is, at the time of writing, in the first year of a five year development of a new mission computer for a future helicopter system. The main goals of the system are to manage the helicopter mission system by:

- a) Allowing the crew to control the on-board mission sensors;
- b) Processing data received from on-board mission sensors;
- c) Displaying selected mission data to the crew in multiple formats.

This set of system goals means that the mission computer is a large and complex system with several concurrent tasks.

Due to the complexity and goals of the system it is inherently a real-time, multi-tasking system. It is desirable with a system of this complexity to partition the system and to deploy these partitions onto multiple processor cards. This adds a small amount of complexity in relation to communication between the partitions, but provides a number of benefits in terms of reduced impact of change, certification and modular design.

There are two relevant avionic standards (ASAAC and ARINC 653) which outline the architectural requirements and guidelines related to the development of software for modular avionic systems. None of the topics or techniques described in this paper are specific to, or ruled out by, either standard, so they will not be considered in any additional detail.

The mission computer software is being developed in Object Oriented Ada 95, with a small amount (<5%) of C code, which is used to interface to a third party graphics library. The Ada code is layered, with lower layers being formed from reusable foundation code and higher layers being formed from bespoke goal related code. This approach works hand in hand with a partitioned system to separate the code according to system goal or responsibility.

The entire mission computer design is held in a UML model where the class structure and relationships are defined by the UML diagrams, and the procedure bodies are stored textually in the model. Storing the entire design in a UML model allows for a full code generation to be performed. This is very beneficial, but a detailed description is outside the scope of this paper.

## 3 Patterns

Christopher Alexander is an architect noted for his theories about design and is widely regarded as the originator of architectural design patterns. He is one of the authors of the influential book 'A Pattern Language: Towns, Buildings, Construction' (Alexander [15]) in which a 'pattern language' is described, which can be used by any person to design and build at any scale. The book provides rules and describes methods for constructing practical, safe and attractive designs at every scale, from entire regions, through cities, and down to individual fixtures. The idea of a pattern language has been especially influential in software engineering where patterns have been used to document collective knowledge in software design.

A pattern can be described as the generic solution to a generic problem. It is worth noting that algorithms are not thought of as patterns since they solve computational problems rather than design problems. The pattern language directs the definition of a pattern to first describe the situations in which the problem occurs, and then to propose a solution. The use of a language to document the pattern is of vital importance and should not be overlooked. A pattern is, however, not a finished design at this stage and cannot be transformed directly into code.

Gamma [16] realises many patterns in the areas of creation (patterns that deal with the creation of objects), structure (patterns that deal with the composition of and the relationships between objects) and behaviour (patterns that deal with interaction between objects). Although these patterns are well understood and generally simple in concept, each pattern must be programmed anew into each application that uses it. If we take a simple example of the problem that this can cause, we could consider the case of creating a drawing of a flower. We can assume that almost everyone will have recognised the pattern for the shape of a flower, but if we ask people to actually draw a flower each will be different.



**Figure 1** Flowers drawn from pattern knowledge

Looking at figure 1 we can see that although all of the examples are recognisable flowers we have flowers of differing sizes, styles and complexities. This is not a desirable outcome. If not properly controlled each pattern will be programmed repeatedly with slightly differing implementations for slightly different (or indeed for identical) applications or uses of the pattern. For this reason patterns alone are of limited use; they tell us what we need to do but not how to do it.

## 4 Templates

Templates within software engineering suffer from a much poorer definition and a lack of recognition and general use when compared to patterns. In this context we can define a template as the implementation of a pattern which can be directly reused multiple times.

We can relate templates back to patterns by reconsidering the creation of a drawing of a flower. The flower we originally envisaged was a rose. We can provide the template for a rose as a stencil. Looking at figure 2 we can see that we now have a set of flowers which conform to our original intention. By producing an implementation of a pattern and making it available we are providing ourselves with a number of benefits.



**Figure 2** Flower template

Firstly, we enable reuse and ensure that a common style and approach is followed when using the template. This does not mean that every application is now identical however, as we can have flowers of differing sizes and colours. The provision of a template means that a design and implementation can be used without specialist knowledge, thus complex templates can be created which can be reused by any person; in the case of our flowers no artistic ability is required. There is also a great time saving gained by not producing multiple copies of similar designs and code for the original pattern.

It is worth noting that we are constraining the user. We are limiting them to using the template implementation for each pattern. This constraint is well defined and is usually exactly what we are aiming to achieve by the use of the template. If multiple (and significantly different) implementations are required for some reason, then it is not difficult to produce an additional template for this alternate scenario.

## 5 Foundation code

Foundation code can be thought of as a set of building blocks, which form a low-level framework of code items. These building blocks are created early in the lifecycle of a project and are then maintained with only minor modifications and additions.

The intention of foundation code is to provide a well understood and easily reusable set of building blocks, which can be used as the basis for any other design within the system.

In the first release of the mission computer software 6% of the generated classes were foundation code. The foundation code was reused to generate 40% of the remainder of the classes, with 54% bespoke classes.

### 5.1 Key benefits

By reusing the foundation building blocks we encourage a common approach and style, and abstract users away from the complexity within each of the building blocks. Another major benefit of foundation code, resulting directly from the reuse of the code, is that the foundation code need only be tested once even though the code will be used many times over. Test effort can range anywhere between 20% and 70% depending on various project specific conditions. Any savings we can make in this area are thus of great benefit, both from a time and cost perspective. It can be beneficial to test foundation code to a higher level than that of other code in the project in order to gain a higher level of confidence in the code that will be built on so many times.

## 5.2 Things to remember

It can clearly be seen that foundation code and templates hold a number of the same benefits and aims. In fact, all code used in the templates either comes from or is added to the foundation code and so the template code gains all of the foundation code benefits discussed.

Although there is a time saving, and thus a cost saving, associated with reuse, there is also a time and cost associated with the generation of both foundation code and templates. The generation process requires a certain amount of preparation at the beginning of the project in order that the reusable code can be effectively included in the design.

There is a difficulty in that a foundation implementation cannot be ‘tweaked’ to cater for a rare or abnormal case. This issue can be dealt with by ensuring that the foundation code is created at the correct level of generality and that the foundation code is sufficiently well understood so that it is not misused. This requires the early preparation of a solid definition of the aims of the foundation building blocks.

We cannot know all of the system requirements at the start of the project. The requirements will be augmented at a later date, either due to additional customer requirements or as problems are identified with the original requirements. This is the most difficult aspect of foundation code to manage. We can mitigate the risk of any changes to basic system functionality but it requires an early analysis of the requirements and thus preparation.

## 6 Examples

We will now explore some of the templates and their application methods, which are being employed in the mission computer development.

### 6.1 Factory template

The factory pattern is a creational pattern, which is used to create, and allow access to, multiple instances of a class.

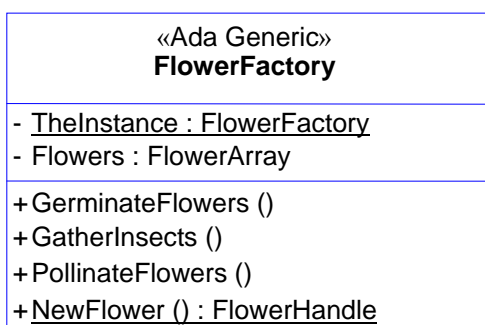


Figure 3 Flower factory template

We can represent this pattern easily in UML and Ada 95 as in figure 3, which considers the template for the creation of a flower using the factory pattern. This template uses an Ada Generic class. Generics are a very simple and common approach to the creation of templates in Ada, and their combination with UML gives a very strong and flexible representation that, for example, could easily add a note to

describe the FlowerFactory class. This template is applied by generic instantiation of the FlowerFactory class.

The factory class provides us with a clean, common interface to the creation of objects, in this case flowers. As is common to all well designed Object Oriented classes, we are abstracted away from the implementation. The factory could dynamically create new flowers for each request or it could contain a previously created set of flowers, which it gives out on request. Additionally, we can see that the factory has been specialised to its usage by the addition of integration with some larger (eco)system that can control the germination and pollination of the flowers created by the factory. We can relate these two features to mission computer software by considering dynamic memory allocation restrictions and integration with system initialisation mechanisms.

### 6.2 Observer template

We will leave flowers behind now and consider the observer pattern, which is a behavioural pattern used to communicate changes between objects in an event driven manner.

The observer pattern might typically be implemented as in figure 4 where ‘Subject’ and ‘Observer’ super-classes are provided and specify the required relationship for the pattern. If we were operating in an unconstrained system the Subject class would also provide storage for a list of Observers. However, when considering a system with static memory allocation requirements, we cannot use a dynamically sized array to store the list and it is not desirable to hold a statically sized list big enough for the requirements of any subclass. For these reasons we must delegate responsibility for the storage of, and access to, the list of Observers to the ExampleSubject class. In this implementation any classes that wish to participate in an observer relationship must be subclasses of Subject and Observer. If we look at the interface provided by Observer it is very bland and provides no information to the ExampleObserver of what has been updated. This effectively means that ExampleObserver can not easily observe more than a single ‘subject’. These requirements and constraints are quite large burdens on the designer in terms of the code to be rewritten for each implementation, and the restrictions limiting integration with other elements of the system.

We could provide the designer with an example of how to use this pattern, but it is a large effort to implement and would require a lot of testing due to the re-implementation for each usage. Alternatively, we can consider providing a more template oriented solution using mixin inheritance as shown in figure 5. Again we see the use of a generic class in the template and the use of UML to represent the class structure and interface. The SubjectMixin is added to the ExampleSubject by generic instantiation. ExampleSubject requires a public version of the Attach and Detach procedures of the mixin to be made available for the ExampleObserver to use, as seen in figure 6.

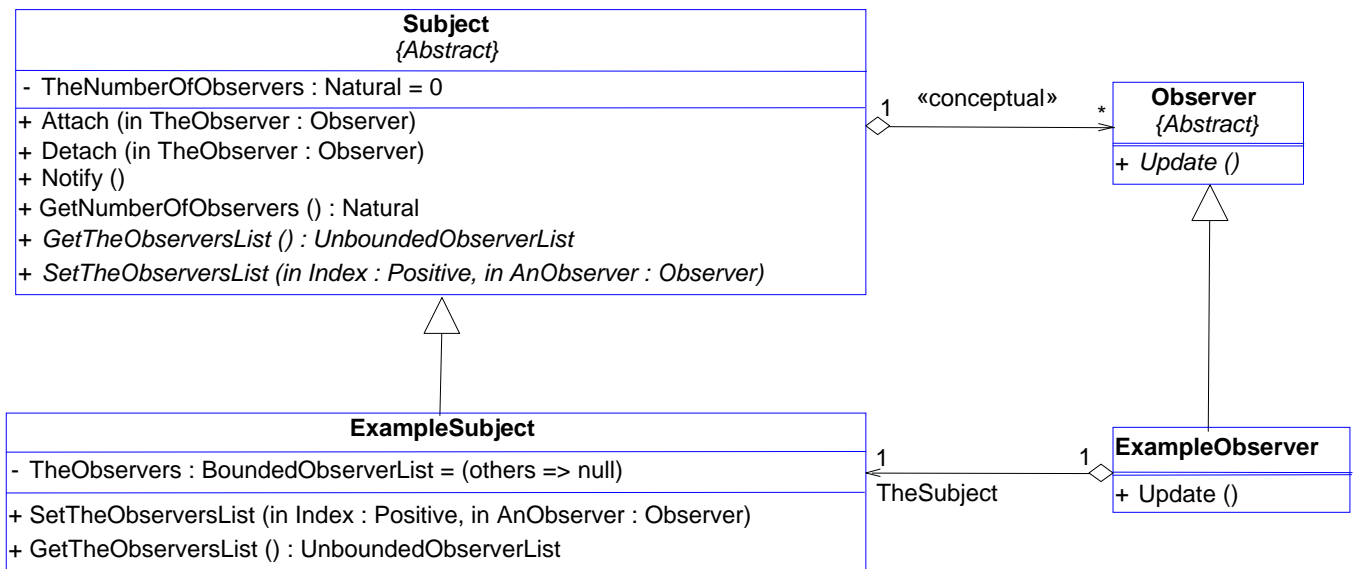


Figure 4 Typical observer pattern implementation

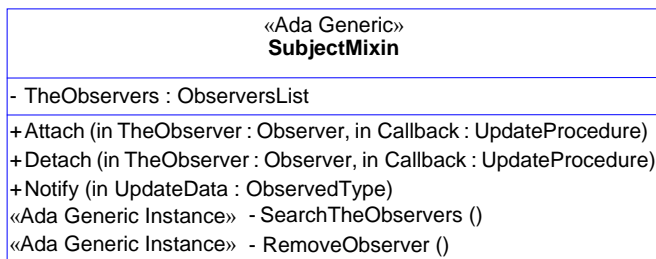


Figure 5 Subject mixin template

The SubjectMixin class provides a solution to all of the problems of the typical implementation and all of the subject complexity is hidden within the mixin class. The ExampleObserver has no visibility of the mixin class.

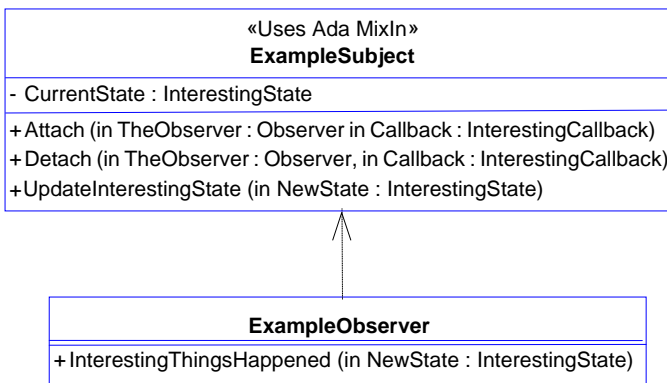


Figure 6 Use of subject mixin

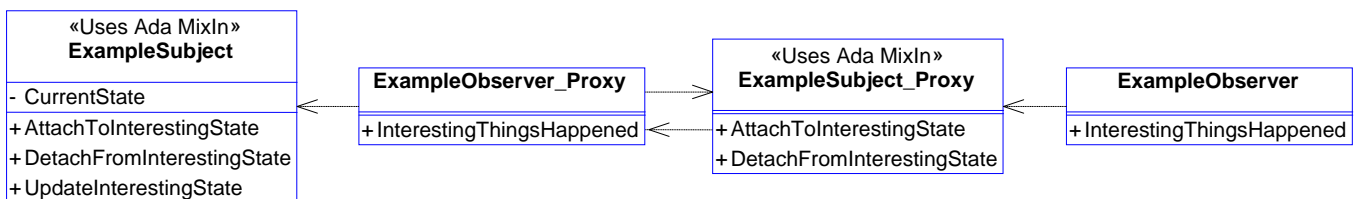


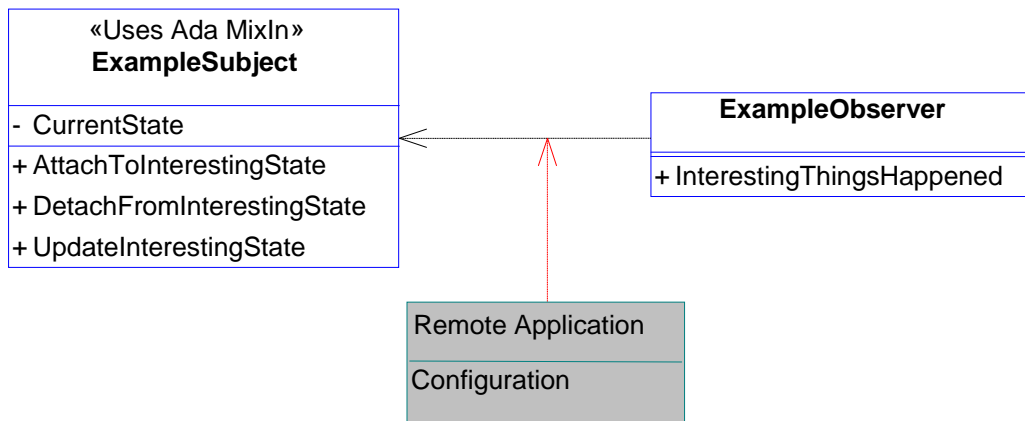
Figure 7 Proxy classes for inter-partition communication

This is a very flexible template, which could be created and used in a number of different ways depending on the generic parameters chosen during the design.

### 6.3 Remote observer

We will finally look in slightly more detail at a template application mechanism. In the previous example we considered a simple observer pattern that can be used between two objects resident in the same partition. If we now consider our requirement to deploy our system across multiple partitions and processor cards, we need to modify this basic design. Figure 7 describes the addition of proxy classes which will allow the template to be used between partitions. This is a very interesting addition to the basic template as we are layering a proxy pattern (a structural pattern used to provide a placeholder for another object) on top of an observer pattern, and we can see yet more reuse in the ExampleSubject\_Proxy class, which is itself using a subject mixin.

We also have a desire to maintain some degree of flexibility in the deployment of classes to the partitions within the mission computer system. We wish to be able to decide that an observer, which is currently local to its subject, should be moved onto a different partition without incurring a large modification burden. Additionally, we do not want our designers to have to worry about the deployment scheme while they are performing their design. In other words, we don't want the designer to modify the UML model so that the ExampleSubject or ExampleObserver classes reference the proxy classes.



**Figure 8** Template application during code generation

In order to perform communication between partitions we can assume that we have some pipes down which we can send information, and that each pipe has some identification mark by which we can refer to it. The proxy classes use the pipes to encapsulate and hide all of the communication and integrate into the system by communicating with the `ExampleSubject` and `ExampleObserver` classes. We can think of the pipe identifiers as being configuration information for the template application, and the dependency between the `ExampleObserver` and `ExampleSubject` classes (see figure 6) as being the application point of the template (the part of the basic design which needs to be changed to insert the proxy classes).

AgustaWestland have developed, with our UML modelling tool supplier, a mechanism by which a template can be defined (as in figure 7) and applied at code generation time using configuration information as in figure 8. The template is defined in UML with the possibility of adding some scripting code to tailor the template application where UML does not provide sufficient descriptive power, and the configuration information is used during the template application to properly configure the proxy classes.

At code generation time the `ExampleSubject` and `ExampleObserver` classes will be automatically edited so that they no longer refer to each other. The proxy classes will be inserted and new links will be formed connecting the proxy classes to `ExampleSubject` and `ExampleObserver`. Using this technique to apply templates we have achieved abstraction during design and implementation from the physical location of the classes.

## 7 Conclusions

This paper has discussed the definitions of patterns, templates and foundation code and has described the ways in which they can be used together and applied to a design. Patterns are common knowledge throughout the software

industry, but without the addition of further information which describes how the pattern is to be implemented they are of little use. Templates provide this additional information and allow a single implementation of a pattern to be reused. Foundation code contains all of the template code and allows the templates to be used effectively in a design and implementation process. The foundation code also provides benefits by abstracting and hiding implementation complexity and reducing testing due to the inherent reusability of the contained code.

We have realised that preparation is of paramount importance if errors are not to be introduced during the analysis and definition of a foundation code framework. If preparation is neglected then it is highly likely that the framework will not be ready when it is required to be used and that a large number of changes will need to be made at a later date due to problems found with the analysis and changes made to requirements.

A foundation code framework is appropriate in any project and, once defined, can be reused across multiple similar projects. Remember, however, that the form of the framework will be different depending on the scope and certification requirements of the project.

Although the templates which have been demonstrated in this paper are relatively small, the techniques described are scalable and are limited only by the templates you can recognise a requirement for. The combination of a well defined foundation code framework and the use of templates is a very time and resource efficient technique which deserves more recognition and development in the software industry.

## References

- [15] C. Alexander, S. Ishikawa and M. Silverstein (1977), *A Pattern Language: Towns, Buildings, Construction*.
- [16] E. Gamma, R. Helm, R. Johnson and J. Vlissides (1995), *Design Patterns: Elements of Reusable Object-Oriented Software*.

# Ada-Europe Awards



Year	Best Paper	Best Presentation
2007	<p>Matteo Bordin and Tullio Vardanega: “Correctness by Construction for High-Integrity Real-Time Systems: a Metamodel-driven Approach”</p> <p>Francisco Garcia, Javier Miranda, José Fortes Gálvez: “Incorporating Precise Garbage Collection in an Ada Compiler”</p>	<p>Lars Hanke: “Secure software-download as part of a complex business process”</p>
2006	<p>Benjamin M. Brosgol and Andy Wellings: “A Comparison of Ada and Real-Time Java™ for Safety-Critical Applications”</p>	<p>Sri Narayanan: “Secure Execution of Computations in Untrusted Hosts”</p> <p>Michael Ward: “Parallel Graphical Processing in Ada”</p>
2005	<p>Peter Amey, Rod Chapman, and Neil White: “Smart Certification of Mixed-Criticality Systems”</p>	<p>Michael González-Harbour: “RT-EP: A Fixed-Priority Real-Time Communication Protocol over Standard Ethernet”</p>
2004	<p>Alan Burns, Andy J. Wellings and S. Tucker Taft: “Supporting Deadlines and EDF Scheduling in Ada”</p> <p>Peter Amey and Neil White: “High Integrity Ada in a UML and C World”</p>	<p>Adrian J. Hilton: “High-Integrity Interfacing to Programmable Logic with Ada”</p>
2003	<p>Miguel Masmano, Jorge Real, Ismael Ripoll, Alfons Crespo: “Running Ada on Real-Time Linux”</p>	<p>Jorge Real: “Running Ada on Real-Time Linux”</p>
<p>The awards for years before 2003 were published in AUJ 24-3 (September 2003) The complete list of awards is also published at <a href="http://www.ada-europe.org/prizes.html">http://www.ada-europe.org/prizes.html</a></p>		

# Ada-Europe 2007 Sponsors

<b>AdaCore</b>	8 Rue de Milan, F-75009 Paris, France Tel: +33 1 49 70 67 16 Email: sales@adacore.com	Fax: +33 1 49 70 05 52 URL: www.adacore.com
<i>Contact: Kathy Fairlamb</i>	Email: fairlamb@adacore.com	
<b>Aonix</b>	66/68, Avenue Pierre Brossolette, 92247 Malakoff, France Tel: +33 1 41 48 11 02 Email: info@aonix.fr	Fax: +33 1 41 48 10 20 URL: www.aonix.fr
<i>Contact: Patricia Langle</i>	Email: patricia.langle@aonix.fr	
<b>Ellidiss Software TNI Europe Ltd</b>	Triad House, Mountbatten Court, Worrall Street, Congleton, Cheshire, CW12 1DT, UK Tel: +44 1260 29 14 49 Email: pam.flood@ellidiss.com	Fax: +44 1260 29 14 49 URL: www.ellidiss.com
<i>Contact: Pam Flood</i>	Email: pam.flood@ellidiss.com	
<b>Green Hills Software Ltd</b>	Fleming Business Centre, Leigh Road, Eastleigh, SO50 9PD, UK Tel: +44 (0) 2380 649660 Email: mktg-europe@ghs.com	Fax: +44 (0)2380 649661 URL: www.ghs.com
<i>Contact: Christopher Smith</i>	Email: chriss@ghs.com	
<b>Praxis High Integrity Systems Ltd.</b>	20 Manvers Street, Bath, BA1 1PX, UK Tel: +44 1225 466 991 Email: sparkinfo@praxis-his.com	Fax: +44 1125 469 006 URL: www.sparkada.com
<i>Contact: Rod Chapman</i>	Email:	
<b>Programming Research BV</b>	Postbus 59, 3700 AB, Zeist, Netherlands Tel: +31 30 2435 997 Email:	Fax: +31 30 2435 999 URL: www.programmingresearch.com
<i>Contact: Frans Nomdem</i>	Email: Frans_Nomden@programmingresearch.com	
<b>Telelogic UK Ltd</b>	1 Cornbrash Park, Bumpers Way, Chippenham, Wiltshire SN14 6RA, UK Tel: +44 1249 467 600 Email: info@telelogic.com	Fax: +44 1249 467 610 URL: www.telelogic.com
<i>Contact: David Brown</i>	Email: steven.fitchett@telelogic.com	