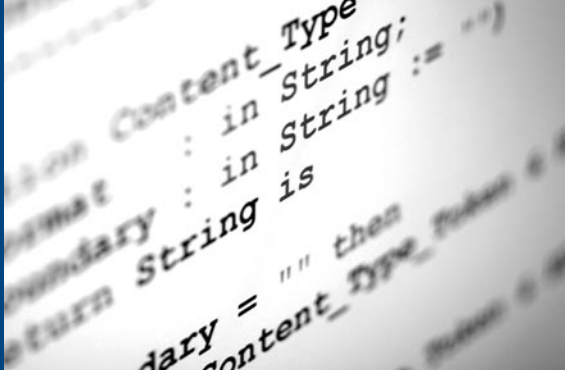
GNAT Pro insider

an AdaCore Publication
Autumn/Winter 2012–2013
www.adacore.com



newsflash

GNATprove Awarded Prize at Formal Methods Symposium

A prototype release of GNATprove, a formal verification tool based on Ada 2012 contracts, received honors in a competition held at the International Symposium on Formal Methods in Paris in August 2012. The contest required the teams to apply verification tools to three sample programs, with the goal of allowing the teams to compare approaches and to learn from each other. GNATprove received the "Distinguished user-assistance tool feature" prize for its integration of proving and run-time assertion checking.

GNAT Pro Chosen for Energy Management System

SmartSide, a Paris-based company providing multi-energy meter data management systems, has adopted the Ada programming language and the GNAT Pro development environment to implement their Smart Devices platform. This platform processes consumption data for all types of energy, as well as environmental data from a variety of devices, and its infrastructure is designed to be scalable, reliable and fault resistant. GNAT Pro and Ada were chosen because of their long and successful track record in the aerospace and defense industries, where high levels of reliability are critical.

contents

GPRbuild 2.0 Enhances Support for Multi-Language Projects

Easier Deployment, New Build Options

A major new release of GPRbuild, the powerful but easy-to-use tool for automating the construction of complex multi-language software, brings a number of enhancements including a new deployment tool GPRinstall and full support for aggregating hierarchies of libraries.

GPRbuild is based around "project files": text files using an Ada-like notation to specify software project properties relevant to generating an executable or performing static analysis. Typical properties include the locations of source and object files, which languages the source code is written in, file naming conventions, and required switches. Project files can be created and edited directly by the developer, or more simply through interactive wizards in the GPS and GNATbench IDEs. Through a project file a single invocation of GPRbuild can generate a complete system or submit the components to various project-aware static analysis tools.

Until the release of GPRbuild 2.0, the installation of the components into their destination locations remained a manual step. GPRinstall, a new installation tool in GPRbuild 2.0, now automates this last step. It uses an enhanced project file syntax that can describe how the software system should be deployed once built.

Another innovation in GPRbuild 2.0 is support for grouping an entire hierarchy of dependent libraries into one single library file. This can simplify the delivery and maintenance of a software product, since keeping track of multiple interdependent library files and making sure they are consistent can prove difficult.

To satisfy this need, the notion of "aggregate library projects" has been introduced. For such a project, all the software components across multiple subsystems are combined into a single library that can then be easily distributed.

GPRbuild 2.0 is compatible with GNAT Pro versions 3.16a1 through 7.1. It is available on most host/target configurations provided by AdaCore. To learn more, please contact **info@adacore.com**.

Ada 2012 Approved by ISO, Featured on New Website

The Ada 2012 standard has been formally approved and published by the International Organization for Standardization (ISO), thus marking the successful completion of this most recent language revision. Ada 2012 brings a number of major enhancements, including contract-based programming and improved support for real-time and multicore applications.

Accompanying the language's approval and publication, a website **www.ada2012.org** dedicated to Ada 2012 has been launched by the Ada Resource Association (ARA), a non-profit organization chartered to promote Ada and maintain the Ada language infrastructure. The site includes a number of short films (3 to 5 minutes in duration) featuring Ada experts discussing the new language facilities and providing historical context. The site also hosts a variety of learning materials, including tutorials, webinars, white papers, and links to software downloads.

GNAT Industrial User Day Keeps Users Informed

AdaCore hosted the 2012 GNAT Industrial User Day in Paris on September 25, with customers, partners, and students participating. The agenda included an Ada 2012 overview, news on the latest tools and toolset features, product and research roadmaps, practical workshops from technology experts, and presentations by customers on their use of Ada and AdaCore tools. The in-depth content and the opportunity for direct interaction between the GNAT Pro developers and the product's users made for a successful day, and a similar event is being planned for 2013.

< current releases >

GPS 5.2

The latest release of the GNAT Programming Studio (GPS) IDE includes a number of improvements:

- Enhanced Search facility, allowing search in current project only, search in the run-time files, and sub-expressions in regexp mode search and replace
- Codefix improvements, including batch mode and support for additional patterns/errors
- Experimental support for the Mercurial version control system
- Improved support for C and C++, with new or improved handling of features such as C++ mangled names, C and C++ interactive smart completion, and Ada to C++ source navigation
- Support for the GNATtest unit test harness generator
- Improved GNATcoverage integration
- Generation of elaboration dependency graphs
- Ability to move files using drag-and-drop in the Files view

SPARK Pro 11

SPARK Pro 11.0 is a major release with many new features:

- Improved support for generic subprograms, including proofs for both absence of run-time errors and demonstration of partial correctness properties
- Major improvements to the use of functions in proof contexts, enabling developers to eliminate the vast majority of axiomatic Simplifier user rules
- The addition of a new statement, the "--- assume" annotation, to support proofs
- The promotion of SPARKBridge to a fully supported feature. This tool can be used to determine why the Simplifier or alternative provers cannot prove an undischarged verification condition
- The inclusion of a new tool, the counter-example generator Riposte.

In addition to these functional enhancements, the Examiner can now analyze larger and more complex programs thanks to its improved dynamic memory management.

< in the pipeline >

GNAT Pro 7.1

This major new release will incorporate a number of enhancements, many based on user suggestions. Key new functionality includes:

- Language-related features, including completion of Ada 2012 support, a new facility for automatic endianness conversion ('Scalar_Storage_Order), and support for dimensionality checking with new aspects and packages
- Run-time improvements, including extended overflow check support and lock-free protected objects
- New or enhanced tools, including GPRbuild 2.0 and GPRinstall (described on page 1), a new version of the GDB debugger (7.5), and new rules in GNATcheck
- Migration of the compiler technology to the GCC 4.7 back-end.

GNAT Pro 7.1 is scheduled for release on most platforms during Q1 2013.

CodePeer 2.2

The upcoming release in Q1 2013 will offer a variety of improvements:

- New features including full support for GNAT project files, support for message review from HTML reports, an option for CSV output (allowing post-processing results from a spreadsheet), and a new -level switch to set the analysis level (accuracy and speed)
- Improvements in the support for math functions and floating point values, in the treatment of No_Return procedures, and in the handling of MessagePatterns and -output-only
- Fewer false positives
- Initial integration into GNATbench

GNATbench 2.7

The GNATbench 2.7 release represents a major upgrade. In addition to supporting both Eclipse 3.8 and 4.2, the new version provides an integration with CodePeer, supports mixed Ada-Java development for both native and Android projects, and directly supports GPRbuild. Debugging has also been significantly improved, with full support for all Ada types and predefined Ada launch configurations. Finally, the documentation has been extended to include the reference manuals for the Ada language, the GNU tools, SPARK, and CodePeer. GNATbench 2.7 will be available during Q1 2013.

< academia corner >

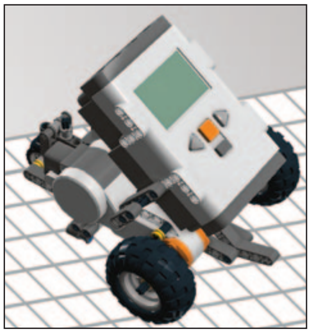
Spotlighting a GAP Member
Universidad Politécnica de Madrid (Madrid, Spain)

Ada, LEGO, and Real-Time at the Technical University of Madrid

A course in real-time programming, including such classical topics as concurrency, real-time scheduling and schedulability analysis, and I/O device programming, is an essential part of a Computer Science curriculum. Professors Juan Antonio de la Puente and Juan Zamorano have been teaching such a course at UPM for many years, using Ada as the main programming language. They chose Ada because of its comprehensive support for reliable software engineering and native concurrency.

Formulating exercises and projects that are both realistic and appropriate for student use presents a challenge. The UPM professors have decided to base the labs on the versatile and low-cost LEGO MINDSTORMS NXT robotics platform, using a GNAT cross development environment to build the Ada applications.

Peter Bradley, a Master's Degree student in the Real-Time Systems and Telematic Services Architecture Research Group at UPM, has ported GNAT for LEGO MINDSTORMS NXT to a Linux host and has put together a set of software development tools for generating Ada real-time control systems that comply with the Ravenscar Profile on the LEGO robotics platform. These tools will be introduced in the real-time systems course as a basis for student projects, supporting all critical concurrency and real-time scheduling aspects, while offering compatibility with modeling tools such as Simulink. Students will thus be able to implement an embedded system from beginning to end. The simplicity of the Ravenscar Profile and the use of programming patterns for high-integrity systems ensures that the limited RAM memory of the MINDSTORMS platform (64KB) is not an issue.



A simple vehicle with two independent motors serves as an instructive example: a PID (Proportional Integral Derivative) control feedback mechanism, implemented and tuned in Simulink, is used to control the motor's speed. The automatically generated C code is then included in the real-time Ada architecture, using the standard Ada interface to C.

Using Ada on the LEGO MINDSTORMS NXT gives students a complete perspective on developing a real-time embedded system, within the constraints of an affordable academic laboratory. This provides a practical alternative to conventional software models that lack a "real" development experience.

For more information please visit **web.dit.upm.es/~str/proyectos/mindstorms**, or contact Prof. de la Puente at **jpuente@dit.upm.es** or Prof. Zamorano at **jzamora@datsi.fi.upm.es**.

Interview with S. Tucker Taft

Vice President, Director of Language Research AdaCore US

Tell us about your background and how you came to be involved with Ada and AdaCore. What is your current role?

I first learned about Ada (then under development) in the late 1970s when I was the systems programmer at the Harvard student computing center. I was already very interested in language design and implementation, and after joining Intermetrics in 1980 I first worked on one of the early Ada environments for the Air Force and then led the design team for the Ada 95 project. This project brought together some of the world's top talent in programming languages and gave me the chance to advance a number of ideas that I had been thinking about for some time. Adding major features like Object-Oriented Programming and protected types to Ada 83 was a definite challenge, and there were some fireworks along the way, but amazingly by the end we were all still talking with one another and were all quite proud of the final result.

Fast forward seven years from 1995. It was 2002, the Internet bubble had burst, and Intermetrics (then called AverCom) had closed down. I went off to found SofCheck along with several colleagues, basing the business on some intellectual property we had purchased from AverCom. This included the early technology underlying what would eventually become CodePeer.

Fast forward another nine years, to 2011. SofCheck's static analysis technology had matured and had been licensed to AdaCore for the CodePeer product, and it made good financial and technical sense at that point for us to join forces with AdaCore. At around that time I became interested in multicore programming, which led to an urge to start designing a new language, this time from scratch. This new language, ParaSail, is now a big part of my focus, and it is acting as a kind of testbed for new parallel programming language ideas that might find their way into AdaCore commercial products at some point. All of which goes well with my official AdaCore title of "Vice President and Director of Language Research."

As the lead designer of Ada 95 and a major contributor to the subsequent language revisions, you have played a significant role in Ada's evolution for more than twenty years. Any "lessons learned" from your experience?

Good language design requires a very firm shared vision about the mission of the language. This can be provided by a strong technical leader, or by a committee with strong common values. Jean Ichbiah supplied that vision for Ada 83, and I made an effort to do likewise for Ada 95. Ada 2005 was different, and in some sense more interesting, because we did not have a single technical lead; instead we relied on a relatively democratic Ada Rapporteur Group under ISO auspices. But the final result was very cohesive, thanks to our sharp common focus on safety, reliability, and efficiency. The Ada 2012 revision was conducted in a similar fashion, but here the goal of providing an excellent "contracts" capability was the guiding theme. Almost everything we did was aimed toward making that capability elegant, powerful, and internally consistent.

What do you see as the main directions for programming language design and research in the future?

The three main directions I see are parallel programming (both tightly coupled and loosely coupled), executable modeling languages, and formal methods. As the number of cores/chips increases, and the price of computing in the "cloud" drops, I believe all developers will need to take advantage of these new resources. Unlike the exponential growth in clock speed we enjoyed for most of my career, this new exponential growth in parallel and distributed computing resources requires a significant paradigm shift for software developers. I believe our languages need to change fundamentally to allow us to tap the full potential of this shift. As far as executable modeling, we are beginning to finally see the progression toward very high level, domain-specific languages; this trend was predicted decades ago but never quite came to pass. AdaCore is already a leader in the application of formal methods to programming with our SPARK Pro, CodePeer, and Ada 2012 offerings, all of which are bringing concepts such as pre- and postconditions to a much broader audience. I see such formal concepts becoming a standard part of industrial-strength software development, as society becomes more and more dependent on software for its critical infrastructure.

Any hobbies or outside interests that you'd like to share?

I am a member of a masters track and field club, called "Mass Velocity," and have been on the national men's 50+ gold-medal 4x400m relay team a couple of times. Quite a thrill, and it keeps me in better shape than I have been at any earlier time in my life.

AdaCore at HILT 2012

ACM SIGAda's annual international conference, renamed HILT (High Integrity Language Technology) to reflect its broader theme, was held in Boston during December 2–6. AdaCore was a Platinum Level sponsor, and company personnel participated actively in the conference organization and program. Ben Brosgol was the Conference Chair, Tucker Taft was Program Co-Chair, and Greg Gicca was Publicity Chair. Johannes Kanig and Tucker Taft delivered tutorials, and paper authors or co-authors from AdaCore included Geert Bosch, Claire Dross, Johannes Kanig, Hristian Kirtchev, Vincent Pucci, and Ed Schonberg.
www.sigada.org/conf/hilt2012/

Lego MINDSTORMS photo by Peter Bradley. Tucker Taft photo by Phyllis Yale.

Dealing with Integer Overflow

GNAT Pro Offers Precise Control of Program Behavior

The discrepancy between the infinite set of mathematical integers and the finite set of integer values on any particular hardware has always presented a challenge for programming languages. A key issue is the semantics when an integer operation overflows, yielding a magnitude that exceeds the capacity of a machine integer.

If an integer operation overflows and overflow checks are enabled, the Ada language definition allows the implementation to either raise an exception (Constraint_Error) or else — and in particular for intermediate results — to compute a correct mathematical result that is then used in subsequent operations. This flexibility may result in non-portable behavior, since mathematically associative operations such as integer "+" are not necessarily associative at the machine level. The integer expression A+B+C may be evaluated in Ada as either (A+B)+C or A+(B+C), but one might yield a correct result while the other one generates an overflow.

Ada 2012's contract-based programming features introduce some further considerations, since integer expressions in assertions such as pre- and postconditions should generally be evaluated mathematically versus as machine integers. For example:

```
procedure P(A, B, C : Integer) with
  Pre => A+B+C <= Integer'Last;
```

In a call such as P(Integer'Last, Integer'Last, Integer'Last) the precondition should evaluate as False rather than raising an exception because of overflow. And in a call such as P(Integer'Last, 1, -1) the precondition should evaluate deterministically as True rather than possibly raising an exception from an intermediate result overflow. To address both the portability issue and the problem of mathematical versus machine interpretation of expressions in assertions, GNAT Pro 7.1 provides comprehensive control over the handling of intermediate overflow. The compiler can operate in three modes, with independent control over the treatment of expressions inside and outside assertions. The modes have the following effect when overflow checks are enabled:

- STRICT: all intermediate computations use the base type's arithmetic, with a check for overflow outside the base type's range;
- MINIMIZED: most intermediate overflow checks are eliminated since computations are performed using the largest native integer type; and
- ELIMINATED: no overflow occurs for intermediate results since computations are performed using arbitrary precision arithmetic as required.

The mode can be specified through either the OVERFLOW_CHECKS pragma, or the -gnato? or -gnato?? compiler switch. For further details please refer to Appendix D (Overflow Check Handling in GNAT) in the GNAT Pro User's Guide.

newsflash

CodePeer Selected by Saab EDS

Saab Electrical Defence Systems in Sweden has adopted the CodePeer static analysis tool for use on the GIRAFFE project, a family of land and sea radar-based surveillance and air defense command and control systems. For this type of critical software, defects must be prevented or at least detected and corrected before they get into deployed code. CodePeer, with its ability to find potential run-time and logic errors statically, is specifically designed to meet this requirement. Employing this tool during new software development helps the Saab engineers to avoid problematic constructs and to produce more robust code.

GtkAda 3.4 to be Released

The next version of GtkAda will support GTK+ v3, a major upgrade which includes an API for CSS-based graphical themes, new widgets, and other enhancements. GtkAda 3.4 will give Ada developers full access to the GTK+ 3 toolkit, with a more strongly typed interface than would be available in other languages such as C. The new Ada API has a more homogeneous naming scheme, making it easier to use, and behind the scenes the new introspection capabilities of GTK+ are being exploited to verify the correctness of the mapping to the underlying toolkit. GtkAda 3.4 is scheduled for release during Q1 2013.

Conferences / Events ■ November 2012 – June 2013

For up-to-date information on conferences where AdaCore is participating, please visit **www.adacore.com/category/press-center/events/**

Automotive 2012
November 14–15, 2012 / Karlsruhe, Germany
AdaCore is a sponsor, and Johannes Kanig is presenting a paper "Integration von Formaler Verifikation und Test".
www.automotive2012.de/

HILT 2012: High Integrity Language Technology
December 2–6, 2012 / Boston MA, USA
AdaCore's participation in this event is described in a companion article in this issue.
www.sigada.org/conf/hilt2012/

Paris Spaceweek
December 17–19, 2012 / Paris, France
AdaCore is exhibiting at this event.
www.paris-space-week.com

RTECC–Real Time & Embedded Computing Conference
January 24, 2013 / Santa Clara CA, USA
AdaCore is exhibiting at this conference.
www.rtecc.com/conferences/view/109

Safety-critical Systems Symposium '13
February 5–7, 2013 / Bristol, UK
AdaCore is a major sponsor of this event, and Robert Dewar is presenting a paper "Testing and Proving: Strange Bedfellows?"
www.safety-club.org.uk/e210

Embedded World
February 26–28, 2013 / Nürnberg, Germany
AdaCore is exhibiting at this event, and Johannes Kanig is presenting a paper "Integrating Formal Program Verification and Testing".
www.embedded-world.de/en/

RTECC–Real Time & Embedded Computing Conference
March 19, 2013 / Dallas TX, USA
AdaCore is exhibiting at this conference.
www.rtecc.com/conferences/view/113

Certification Together
April 16–18, 2013 / Toulouse, France
AdaCore is a Platinum sponsor of this event.
www.certification-together.com/index.php?option=com_content&view=article&id=73&Itemid=80

DESIGN West
April 22–25, 2013 / San Jose CA, USA
AdaCore is exhibiting at this conference (booth #719), and there are presentations from Ben Brosgol (Object-Oriented Programming for High-Integrity Systems: Pitfalls and How to Avoid Them) and Tucker Taft (Systems Programming in the Distributed, Multicore World with Go, Rust, and Parasail).
ubmdesign.com/sanjose/conference/

Ada Conference UK 2013
April 25, 2013 / Birmingham, UK
AdaCore is lead sponsor of this event. Tucker Taft is presenting an Ada 2012 tutorial, and Robert Dewar is speaking at the closing plenary session.
www.adacore.com/ada-uk-2013

Ada-Europe 2013
June 10–14, 2013 / Berlin, Germany
AdaCore is a sponsor of this conference.
www.ada-europe2013.org/

The GNAT Pro insider is published twice a year
simultaneously in New York and Paris by AdaCore

104 Fifth Avenue, 15th floor
New York, NY 10011-6901, USA
tel +1 212 620 7300
fax +1 212 807 0162

46 rue d'Amsterdam
75009 Paris, France
tel +33 1 49 70 67 16
fax +33 1 49 70 05 52

info@adacore.com
www.adacore.com

AdaCore
The GNAT Pro Company