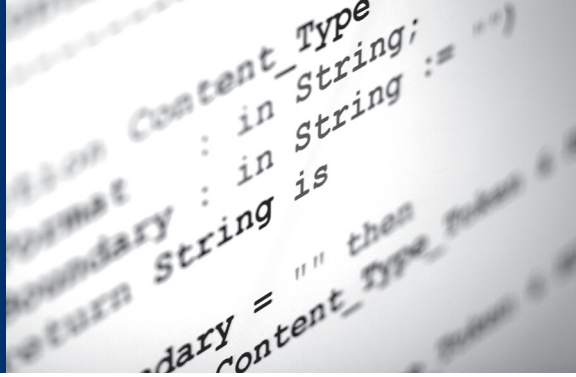
GNAT Pro insider
an AdaCore Publication
Autumn/Winter 2013–2014
www.adacore.com



newsflash

Astrium Selects GNAT Pro and PolyORB for International Space Station

EADS' Astrium subsidiary has selected the GNAT Pro development environment and PolyORB middleware toolset for software in the Core Ground System (CGS) on the Columbus laboratory, one of the European contributions to the International Space Station. GNAT Pro and PolyORB help achieve efficient, reliable communication across the Ada applications that Astrium developed for Columbus's ground and onboard applications, and PolyORB also supplies the interoperability necessary for smoothly integrating components written in Java or C++. The combination of GNAT Pro and PolyORB provide Astrium with a 'one-shop' solution for both the Ada development and the middleware implementation of the CGS.

TOYOTA ITC Japan Selects SPARK Pro Language and Toolset

TOYOTA InfoTechnology Center (ITC) Japan has selected the SPARK language and SPARK Pro toolset for a high-reliability software research project, with the goal of demonstrating that software requirements can be transformed into an implementation that can be proved free of run-time errors. The project is planned as a first step to composing larger ultra-low-defect vehicular systems, using formal methods to mathematically assure a variety of correctness properties and to reduce the development and maintenance effort by eliminating the need for post-deployment corrections.

contents

GPS 6.0 Released

New Major Revision Upgrades IDE's Look and Feel

For this new GNAT Programming Studio (GPS) release, the AdaCore development team has conducted a complete review of the existing interface and made it cleaner and easier to use. Based on customer feedback, the improvements include more space for editing and a number of design changes to bring program-related information within easy reach. This is supported by a new relational database at the heart of the GPS engine to make searches much more efficient. The guidelines underlying the GPS 6.0 revision help the IDE achieve its main goal: to serve as a customizable platform for multi-language multi-tool integration, usable by developers at all experience levels.

The improvements to the IDE's look and feel take advantage of the latest Gtk+/GtkAda graphical toolkit and fall into several categories:

- **Interface reorganization**
  - More economic usage of screen real estate, for example by placing tabs on top of the views
  - Improved toolbar with monochrome icons
  - Ability to position tabs on the side, for better appearance on wide screens
  - Better handling of documents, for example with editors and views clearly separated
- **Omni-search**
  - Ability to search in all contexts, including source filenames, file contents, and code entities, directly from the main toolbar
- **Views**
  - Ability to directly access view preferences from a local toolbar
  - Support for "compact mode" (specification and body on same line) and "flat mode" for outline view
- **Colors**
  - Ability to change the color of many GPS elements, including selection of white foreground and dark background
  - Support for color themes, to switch quickly among preselected settings

GPS 6.0 also brings improved performance and new capabilities:

- Use of a relational database for a more efficient (and persistent) storage mechanism for cross reference information
- Language support for SPARK 2014, plus syntax highlighting and tooltips for Ada 2012 and SPARK 2014 aspects
- Several editor enhancements, ranging from completion and autofix to text formatting
- A number of additions to the scripting API

The GPS 6.0 enhancements have received an enthusiastic response from the product's beta sites.

An IDE is more than a tool, it is a platform and framework that developers depend on to organize and conduct their daily work. A major effort has thus been invested in improving the overall usability of GPS based on customer comments, giving it an intuitive and visually attractive look and feel. In short, and somewhat subjectively, the idea is to provide users with a "pleasant experience". This has been a major objective for GPS 6.0 and will continue to drive product development.

Future enhancements to GPS are planned in several areas, including debugger integration, documentation generation, and support for aggregate projects.

A GPS 6.0 demo will be available at **www.adacore.com/gps-demo**. For further information, please contact **info@adacore.com**.

< current releases >

SPARK Pro 11.1

This new release incorporates a number of enhancements to the SPARK Pro toolset as well as some minor bug fixes, and also brings the technology to several new platforms. To ease the transition from SPARK 2005 to SPARK 2014, the Examiner can now optionally ignore SPARK 2014 aspects. Various improvements to the proof system mean that users should see more Verification Conditions being discharged automatically. The toolset is now available for 64-bit Windows, and Victor is now available on Solaris.

SPARK Pro 14.0 Beta

A beta release of the SPARK 2014 tools is now available to SPARK Pro customers. This is a major upgrade to the SPARK tools, with a ground-up rewrite based on the GNAT Pro front end and an entirely new flow analysis engine. Verification Condition generation and proof are now handled by GNATprove (using Why3 technology) with proof results presented at the source code level in the GPS IDE. For more details of the SPARK 2014 language please see the Technology Corner article on Page 4 of this newsletter.

GNAT Pro for Wind River Linux

GNAT Pro is now available on the Wind River Linux platform. GNAT Pro can be used in conjunction with Wind River products to compose multilanguage applications combining Ada, C, and C++, and Ada code can be manipulated and analyzed through Wind River's Linux browser and tools. GNAT Pro for Wind River Linux supports all versions of the Ada language standard from Ada 83 through Ada 2012 and is tightly integrated into the Wind River Workbench development environment.

< in the pipeline >

GNAT Pro 7.2

The next major release of the GNAT Pro toolsuite will incorporate more than 120 new features, including Ada 2012 mode enabled by default, many new warnings and improved diagnostics, code generation optimizations, support for symbolic traceback in shared libraries, and improved cross Ada/C++ exception handling. GNAT Pro 7.2 will also introduce two new tools:

- GNAT2XML, for generating XML files from Ada sources, which will help in writing Ada analysis tools quickly in any language
- GNATdashboard, for quality assurance monitoring and history (see below)

GNAT Pro 7.2 will also provide enhancements to existing tools, including:

- A new version of GNATpp, providing improved Ada layout and greater flexibility
- Support in the GPRbuild multipurpose builder for distributed builds, and better support for parallel builds

The GNAT Pro 7.2 toolsuite will be released during Q1 2014, and supported platforms will include the ARM architecture (bare board, GNU/Linux, VxWorks 6 and Android), x86 VxWorks 6 Cert, and Windows 8.

GNATdashboard

GNATdashboard is a new visual tool that supports quality assurance activities, integrating and aggregating the results of AdaCore's many static and dynamic analysis tools within a common interface. Supplied as a plug-in to the open source SonarQube code quality management platform, GNATdashboard fits in naturally with a software development team's workflow by using project files to configure, run, and analyze the output from AdaCore tools. A driver program processes data such as compiler warnings, CodePeer diagnostic messages, style check violations, and coverage data, and makes it available for reference and analysis through the web-based SonarQube application. Developers can thus quickly obtain up-to-date information, logically organized and presented, concerning the various quality factors associated with their project. A beta release will be available in Q1 2014.

CodePeer 2.3

The next release of the CodePeer automatic code review and validation tool will include its own GNAT front-end (which can be installed independently of GNAT Pro), improved support for legacy Ada (83) code, message review capabilities via pragma Annotate, more precise messages and fewer "false positives", new warnings when a formal parameter could be declared with a more restrictive mode, support for floating point overflow on unconstrained types, support for IEEE floating point semantics, and improved GPS and GNATbench integration. The product will be available during Q1 2014.

< academia corner >

AdaCore University Launched

To make it easier for developers to quickly come up to speed in Ada, a free, web-based educational resource center known as AdaCore University has been launched by AdaCore. The site contains pre-recorded courses and other learning materials and provides access to AdaCore's GNAT Ada toolset for writing and running example programs. Students at all levels of experience and expertise can begin writing programs quickly and can proceed at their own pace. The site has flexible controls for navigating through the lessons, including pausing, rewinding, fast forwarding, etc.

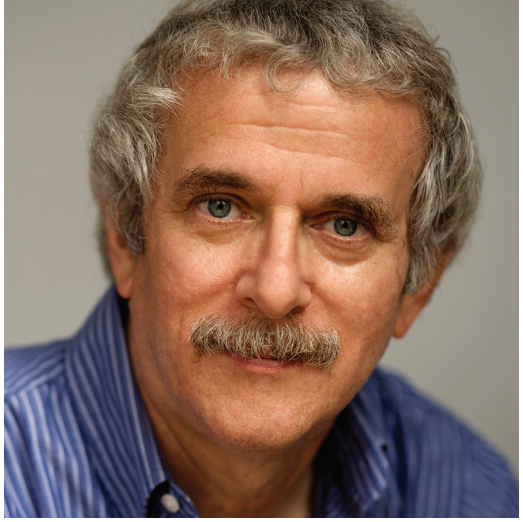
The courses educate through examples, allowing students to see, understand and experiment with most features of the Ada programming language, and comparisons with C, C++, and Java are used to illustrate particular points of semantics. Each course includes one or more interactive quizzes, with detailed explanations of the solutions. Drawing on the experience and teaching credentials of Ada experts such as AdaCore founders and New York University Emeritus Professors Robert Dewar and Edmond Schonberg, the courses explain Ada's technical concepts with insight into the rationale and usage of particular features.

The initial curriculum includes two courses, presented by AdaCore University project leader Quentin Ochem:

- Ada 001 (Overview) – a module that presents an overall picture of the language and shows how to build and run programs
- Ada 002 (Basic Concepts) – a module that comprises a series of lessons on Basic Types, Statements, Arrays, Records, Subprograms, and Packages

These courses are complemented by a set of lab exercises to reinforce the concepts covered by the lectures. The courses cover the latest version of the Ada language—Ada 2012—so students will learn about new features such as contract-based programming (preconditions, postconditions, invariants). The AdaCore University website also hosts a number of technical papers on Ada, offering insight into particular aspects of the language's design and usage.

AdaCore University is an ongoing, live project that will be expanded to include more advanced courses on Ada, as well as training materials for SPARK 2014—a new version of the Ada-based SPARK programming language for high-integrity software. For more information, please visit **u.adacore.com.**



Interview with Ben Brosgol Senior Technical Staff, AdaCore US

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

I was a grad student in Applied Math and Computer Science at Harvard back in the late 1960s and early 1970s, and that's where I became intrigued by programming languages and compilers. This was a golden age for research in software methodology, and new ideas like structured programming, encapsulation (information hiding), object orientation, and concurrency primitives (semaphores, monitors) were revolutionizing the programming world. After receiving my Ph.D.—for trivia buffs the subject of my dissertation was *Deterministic Translation Grammars*—I joined Intermetrics in 1973 and soon got involved with the project that would lead to Ada. I was in charge of the "Red" language in the design competition, but we ultimately lost out to the "Green" language from Jean Ichbiah at CII-Honeywell-Bull in France.

Ada has pretty much been my career ever since. In early 1983 I joined Jean Ichbiah's company, Alsys, and helped found its US branch. I did some independent consulting in the early 1990s, and a good chunk of that work was on the Ada 95 project under the direction of Tucker Taft. After another stint with Alsys (then Aonix), I joined AdaCore in March 2000. My work here has been a grand mix of technical and marketing: writing articles, presenting papers and tutorials at conferences, conducting professional Ada courses, preparing press releases, and so on. On occasion I've managed some development contracts, including a current project to produce qualification evidence for GNATcoverage as a verification tool under DO-178B. Oh yes, I also edit the *GNAT Pro Insider* newsletter, which makes this interview an interesting exercise.

I've been involved with the ACM SIGAda organization since its inception, including a long stretch as Vice Chair for Meetings and Conferences and two terms as Chair. I guess I can take the credit (or blame) for launching the TRI-Ada series of conference/trade-shows back in the late 1980s, which helped put Ada on the map.

Occasionally I've branched out into other areas. I was a member of the Expert Group that produced the Real-Time Specification for Java, and that work was both fun and challenging, although it hasn't had much impact in the embedded systems marketplace. Hard real-time was not what Java was originally designed for, so we had to extend Java's thread scheduling and memory management features to get the necessary predictability. I've also been following the DO-178C effort and have presented several tutorials and papers on that topic.

You've been working in the computer software industry for a long time. Any insights or predictions?

It's depressing that the term "software crisis" has been kicking around for forty-five years but seems just as timely today as when it first appeared. So this is a chronic condition, not a crisis. Unfortunately, some of the solutions that have appeared over the years—such as fancy environments that make it easy to write and run programs—may be part of the problem, since they can encourage a "slap something together and debug it" approach to software development. But debacles such as Knight Capital Group's $440 million "glitch" last year tend to get people's attention, so there's hope for progress. Tools and methods that focus on the front end of the development cycle (requirements capture/analysis) or on the interface between the software and the overall system are especially valuable, since that's where the subtle issues tend to lurk. Languages are important of course; they can help in expressing requirements as part of the source code and in detecting errors early. We've seen this with SPARK and, more recently, Ada 2012 with contract-based programming. And combining formal methods with traditional testing can be very effective.

Software does not have to have bugs. Industries with a rigorous certification process and a longstanding culture that focuses on safety and reliability, such as commercial aviation, have shown that the problems are solvable.

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

I'm a bit of a movie buff, with a particular fondness for some of the early Alfred Hitchcock thrillers. *The 39 Steps* is a treat, with the standard Hitchcock plot device of a hero pursued by both the bad guys and the law. And *Rebecca* is one of my favorite films, with brilliant performances from Laurence Olivier, Joan Fontaine, and Judith Anderson. It won the Best Picture award in 1940, but regretfully Hitchcock did not win as Best Director. Another gem from later in that decade was Carol Reed's *The Third Man,* with its memorable zither music and Orson Welles at his cynical best. Without CGI special effects to attract an audience in those days, movies had to focus on character and story. The good ones did this well.

GNAT Industrial User Day

AdaCore hosted the 2013 GNAT Industrial User Day in Paris on September 25, with customers, partners, and GNAT Academic Program members attending. With the theme of "Meet, Learn, Share", the event featured presentations from company staff, partners, and very importantly, AdaCore customers. The company's presentations covered a range of topics including the role of formal verification, the new AdaCore University initiative, Ada on ARM processors, and plans for future tool / technology development. The "upcoming and future" sessions were particularly well received, with much excitement around GPS 6.0, GNATdashboard, GNAT Tracker 3.0, and the more general roadmap. 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 that will be repeated in 2014.

Photograph by Gary Matoso

What's New in SPARK 2014?

The SPARK language has a long history of successful usage in safety-critical and high-security software, since it supports statically proving properties such as freedom from run-time errors or compliance with a formal specification of a program's requirements. And because SPARK is a subset of Ada, SPARK programs can be compiled by any standard Ada compiler. A major new version of the language, known as SPARK 2014, preserves these benefits while introducing several significant innovations.

Consistency with Ada 2012 and Hybrid Verification

Since Ada 2012 provides contract-based programming features (e.g., pre-/post-conditions, and quantified and conditional expressions) and a uniform framework for aspects, SPARK's "--#" annotations-as-structured-comments syntax has been replaced by constructs with valid Ada semantics. This is not just a cosmetic change. The new syntax is at the heart of SPARK 2014's support for hybrid verification, a novel approach to safely combining formally verified and traditionally verified code in one program based on run-time checking of contracts at their boundary. For example, if a formally verified subprogram is called by code that is not subject to formal verification, the precondition of the called subprogram can be checked when the call is executed. If the precondition fails, then the called subprogram's proofs are inapplicable. Using hybrid verification, developers can selectively apply formal verification to only those portions of a program where it is feasible and cost-effective. Executable contract checking also allows correct contracts to be developed more quickly; this is valuable for both formally and traditionally verified code.

Larger Language Subset

SPARK 2014 supports a richer subset of Ada (now Ada 2012) than did earlier versions of SPARK. The fundamental restrictions that the proof technology relies upon are still enforced (e.g., no access types, no side effects in functions, no exception handlers, no tasks, no controlled types), but many other restrictions are now relaxed. SPARK 2014 allows discriminated types, dynamic constants, dynamic subtypes, recursion, and generic packages and subprograms. A predefined SPARK container library (generic packages implementing maps, sets, vectors, and lists) provides much of the essential functionality of access types for managing flexible data structures, but without the verification problems associated with access types in general.

Easier Ada-to-SPARK Transition

SPARK 2014 makes it easier to transition from non-SPARK Ada to SPARK. Part of a subprogram's contract is the set of global variables that it depends upon (directly or via calls to other subprograms) as inputs and outputs. SPARK 2014 offers an alternative where data dependencies for a given subprogram are determined automatically by analyzing the body. Non-SPARK completions for SPARK interfaces (e.g., a non-SPARK Ada body for a SPARK subprogram declaration) are also supported.

For more information, please see **www.spark-2014.org/about**.

```
-- SPARK 2005:
procedure Accelerate (Increment : Positive);
      --# global in out Control_State;
      --# derives Control_State from *, Increment;
      --# pre  Control_State.Mode /= Braking;
      --# post Control_State =
      --#      Control_State~[Speed => Control_State~.Speed
                                   Increment;
      --#                     Mode  => Accelerating];

-- SPARK 2014:
procedure Accelerate (Increment : Positive) with
  Global  => (In_Out => Control_State),
  Depends => (Control_State =>+ Increment),
              -- Outgoing value of Control_State depends on both
              -- its incoming value and Increment
  Pre     => Control_State.Mode /= Braking,
  Post    => Control_State =
             Control_State'Old'Update
              (Speed => Control_State'Old.Speed + Increment,
               Mode  => Accelerating);
```

newsflash

SPARK and GNAT Help Guide NASA-sponsored CubeSat

A student team from Vermont Technical College has used the SPARK language/toolset and GNAT technology to develop the navigation and control software for a Lunar CubeSat project sponsored by NASA's ELaNa IV program (Educational Launch of Nano-satellites). Measuring 10 cm x 10 cm x 10 cm and weighing 1.33 kg, the CubeSat is scheduled for a November launch into a 500 km earth orbit, where it will remain for three years to test the systems that will be used for the eventual lunar mission. Although they had no previous knowledge of SPARK or Ada, the students came up to speed quickly and were able to produce robust code taking advantage of SPARK's statically verified contracts. For further information about the project, please visit **www.cubesatlab.org**.

Conferences / Events ■ October 2013–April 2014

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

32nd Digital Avionics Systems Conference (DASC)
October 6–10, 2013 / Syracuse NY, USA
Tucker Taft is giving a talk on integrating testing and proof with SPARK. **dasconline.org**

IET International System Safety Conference
October 15–17, 2013 / Cardiff, UK
AdaCore is a sponsor and exhibitor.
conferences.theiet.org/system-safety/

ACM SIGPLAN's SPLASH 2013
Systems, Programming, Languages and Applications: Software for Humanity
October 26–31, 2013 / Indianapolis IN, USA
Tucker Taft is presenting a tutorial and a "tech talk" on parallel/multi-core programming.
splashcon.org/2013/program/tutorials-tech-talks

RTECC Seattle
Real-Time & Embedded Computing Conference
November 5, 2013 / Seattle WA, USA
AdaCore is exhibiting at this event.
rtecc.com/events/details/?id=121&city=Seattle

ACM SIGAda's HILT 2013
High Integrity Language Technology
November 10–14, 2013 / Pittsburgh PA, USA
Tucker Taft is presenting a tutorial on proving safety of concurrent programs. AdaCore is a Platinum Sponsor and exhibitor.
www.sigada.org/conf/hilt2013/

Safety-critical Systems Symposium 2014
February 4–6, 2014 / Brighton, UK
AdaCore is a major sponsor of this event.
www.safety-club.org.uk/e263

ERTS² 2014
Embedded Real Time Software and Systems
February 5–7, 2014 / Toulouse, France
AdaCore is a Gold Sponsor of this event.
www.erts2014.org

Embedded World 2014
February 25–27, 2014 / Nuremberg, Germany
AdaCore is exhibiting at this event - Hall 5/5-348.
www.embedded-world.de/en/

Design West
March 31–April 3, 2014 / San Jose CA, USA
AdaCore is exhibiting at this event - Booth 719.
www.ubmdesign.com/sanjose

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