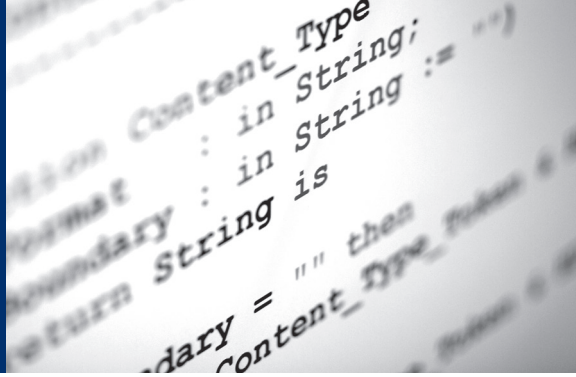
GNAT Pro insider

an AdaCore Publication

Spring/Summer 2014

www.adacore.com



newsflash

Muen Separation Kernel developed using SPARK and GNAT

The Institute for Internet Technologies and Applications at the University of Applied Science in Rapperswil, Switzerland, has issued a preview release of the Muen Separation Kernel, with the goal of creating an Open Source foundation for high-assurance platforms. The Muen Kernel enforces a strict and robust isolation of components to shield security-critical functions from vulnerable software running on the same physical system. To achieve the necessary level of trustworthiness, the Muen team chose the SPARK language and toolset to formally prove the absence of run-time errors, and they used the GNAT development environment to build their software. Future plans include an upgrade of the kernel to SPARK 2014.

Ada in financial systems

Deep Blue Capital (DBC), a propriety trading firm based in Amsterdam, has adopted Ada and GNAT Pro for developing their algorithmic automated trading systems. The software must reliably handle large volumes of price data and daily financial operations on computers that are running continuously, and it also needs to be easily updatable to incorporate new trading strategies. DBC chose Ada as the language that best meets these requirements for both reliability and maintainability.

contents

CodePeer 2.3 Released

Complete standalone package, usable with any Ada compiler

A new major version of the CodePeer static analysis tool is now available. CodePeer performs automated review and validation of Ada source code—including Ada 2012—identifying potential bugs before program execution to find errors. The tool also conducts impact and vulnerability analysis when existing code is modified, and, using control-flow, data-flow and other advanced static analysis techniques, it detects and reports logic errors that would otherwise only be found through labor-intensive debugging.

With CodePeer 2.3, customers will find improved usability, increased functionality, and better tool integration:

- CodePeer 2.3 includes an independent Ada semantic analyzer and can thus operate as a standalone tool, simplifying the installation process. It supports most Ada compilers and allows users to specify target machine characteristics such as endianness and the range of numeric types, and so can now handle the majority of existing Ada 83 and Ada 95 code bases.
- Enhancements to the analysis engine mean fewer "false positives", and CodePeer's diagnostic messages are more precise, for example warning when a formal parameter is declared with a mode that is more general than necessary.
- The tool's support for reviewing messages has been re-engineered to offer additional ways to classify messages. There is also a new optional method for performing a review directly via a source code change (via pragma Annotate), in addition to the existing separate database capability.
- Among its new features CodePeer 2.3 can detect floating point overflow on unconstrained types.
- To simplify the development process, CodePeer 2.3 is more closely integrated into AdaCore's GNAT Programming Studio (GPS) and GNATbench IDEs.

CodePeer comes with several complementary static analysis tools common to the GNAT Pro technology—a coding standard verification tool (GNATcheck), a source code metric generator (GNATmetric), and a document generator.

For additional information, please visit **www.adacore.com/codepeer/**.

GNATcoverage 1.2 Supports Hardware Probes

The latest release of AdaCore's source and object code coverage analysis tool has greatly widened the product's applicability, adding support for trace data generated by hardware probes. GNATcoverage's innovative technology does not require instrumentation of the executable. To meet this goal, previous versions have relied on traces produced by the host-resident GNATemulator target emulator tool. GNATcoverage 1.2 can still use GNATemulator, but it also supports iSystem hardware probes generating Nexus trace data, as well as Valgrind on Linux. By deriving source coverage results from a non-instrumented executable running testsuites directly on the target hardware, GNATcoverage 1.2 can significantly simplify the coverage analysis effort in a certification context.

GNATcoverage 1.2 can handle Ada 95, Ada 2005, and many new features in Ada 2012. It can also be used for SPARK 2014 and provides Beta support for C. Other enhancements include generation of coverage information for generics on a per-instance basis, and improved HTML output.

GNATcoverage has been qualified as a T2 tool for railway applications that need to comply with EN-50128:2011. Qualification material is also available for GNATcoverage usage as a verification tool (DO-178B) or a tool at TQL-5 (DO-178C) for avionics systems. GNATcoverage can supply analysis up to Modified Condition/Decision Coverage (MC/DC) and can thus be used as part of the verification process for systems that need to be certified up to Level A. GNATcoverage is the only non-instrumenting coverage technology that performs full MC/DC analysis/assessment; a technical paper on this subject is available at **www.adacore.com/knowledge/technical-papers/branch-coverage-criteria/**.

For additional information, please visit **www.adacore.com/gnatcoverage/**.

< current releases >

GNAT Pro 7.2

GNAT Pro 7.2 is a major release that incorporates 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 includes several new tools, including GNAT2XML, which generates an XML version of a semantically analyzed Ada program and thus helps developers write Ada analysis tools in any language. Enhancements to existing tools include a new version of the GNATpp pretty printer with improved Ada layout, and an enhanced GPRbuild multipurpose builder that offers greater flexibility and support of both distributed and parallel builds.

This release adds support for Wind River's VxWorks 6 Cert for x86 and LynuxWorks' LynxOS-178 Real-Time Operating Systems (RTOS). It also extends GNAT Pro's ARM support to now include Android, generic Linux on ARM, Bareboard ARM, and Wind River's VxWorks 6 on ARM.

SPARK Pro 14

SPARK Pro 14.0 is the first full release of the next-generation SPARK toolset, which implements the SPARK 2014 language. SPARK 2014 is a rich subset of Ada 2012, excluding only those language features that would make program verification unsound. It enables a novel verification approach that allows combining formally verified code with code that has been verified through traditional means such as testing, and eases the transition from Ada to SPARK for applications that require the additional assurance gained from formal proofs of program properties.

The main features of the new language and toolset include:

- Convergence with Ada 2012 syntax,
- Larger Ada language subset,
- Executable contracts,
- Hybrid verification (the ability to combine unit proof with unit test),
- Formal Container library,
- Generative mode for data dependencies (the ability to perform data flow analysis without explicit global declarations), and
- Improved diagnostics for information flow and verification errors.

SPARK 2014 toolset documentation is available at **www.adacore.com/developers/documentation/category/sparkpro/**. Additional information on the SPARK 2014 language—including a number of programming tips and techniques—may be found at **www.spark-2014.org/**.

Distributed GPRbuild

The latest version of the GPRbuild tool, included in GNAT Pro 7.2, supports distributed compilation and addresses the problem of long build times for very large applications. For users who wish to take advantage of distributed compilation, GPRbuild's innovative approach requires minimal or no changes to project files and does not impose restrictions on what can be built. A project that compiles normally with GPRbuild will also compile in distributed mode.

With distributed GPRbuild, a local machine performs some compilations but also sends compilation requests to any number of remote server machines. Once the compilation phase is done, binding and linking are performed on the local machine. Distributed compilation can be used for Ada, C, C++, or any other language supported by GPRbuild.

Speedup of system builds with distributed GPRbuild can be significant. During the Beta test of the new capability, one user reported that a build previously taking 2.5 hours was now completed in only 10 minutes. Ada is specifically targeted at large and complex applications, and GNAT Pro's new support for distributed GPRbuild helps users take full advantage of Ada's "programming in the large" features while avoiding compilation bottlenecks.

< in the pipeline >

New subscription option for GNAT Pro Safety-Critical

As an add-on service for users of GNAT Pro Safety-Critical, AdaCore is finalizing a new kind of subscription for customers who need access to defect corrections on defined release branches. This facility is especially useful in certification contexts where compiler code generation problems, even when detected long after certification, must be analyzed to assess the potential impact on certified code. Patches to avoid hardware bugs may also be applied. Every corrective action comes with a corresponding impact analysis.

GPS to include new documentation generation tool

The next release of the GNAT Programming Studio IDE will include GNATdoc, a new command-line tool for documentation generation. Among its features are support for Javadoc/Doxygen style of tags in documentation comments, support for comment placement detection, support for separating documentation comments from code comments, and a new extensible HTML back-end.

< academia corner >

Spotlighting a GAP Member

Vermont Technical College (US)

Vermont Tech's CubeSat is in orbit and sending down photos and data

Vermont Tech's Lunar CubeSat, launched into a low earth orbit in November 2013 to test the navigation systems that will be used for the eventual lunar mission, has been successfully transmitting photos and inertial measurement unit data since reaching its initial orbital position. The satellite measures 10 cm x 10 cm x 10 cm and weighs 1.1 kg. The CubeSat project is part of NASA's ELaNa IV program (Educational Launch of Nano-satellites).

The software controlling all aspects of Vermont Tech's CubeSat comprises around 10,000 lines of SPARK/Ada code, written mostly by one undergraduate student. The college is a member of the GNAT Academic Program, and their project was completed using the GPS, GNAT, and SPARK tools from AdaCore and Altran under the direction of Dr. Peter Chapin and project leader Dr. Carl Brandon.

The November 2013 launch included a dozen CubeSats from academia. Vermont Tech's was the only one that used SPARK or Ada; most were programmed in C. "The use of SPARK/Ada helped make our software much more reliable than the others," said Dr. Brandon. "Most of the colleges and universities constructing CubeSats had very little real world experience with space-based systems, and the complicated nature of the software necessary to control a spacecraft can be overwhelming. With only one student doing most of the software work, and with the software evolving rapidly as new requirements emerged, achieving the necessary reliability would not have been possible without the use of the SPARK toolset."

Of the twelve university CubeSats that were launched, Vermont Tech's is the only one still fully functioning

Of the twelve university CubeSats that were launched, Vermont Tech's is the only one still fully functioning. Eight were never heard from, one failed after a week, and one lasted about four months. The other remaining CubeSat only works in sunlight, since the software-controlled battery protection system failed immediately and the batteries were overcharged and destroyed. Although hardware issues may have caused some of the CubeSat failures, software was definitely the problem in one, and possibly with ten others. "From that perspective the use of SPARK/Ada could have prevented many of the CubeSat failures, saving approximately $50,000–$100,000 in hardware, $125,000 in launch costs for each, and years of development time," added Dr. Brandon.

For further information, please visit **www.cubesatlab.org/** or contact Dr. Brandon at **CBrandon@vtc.vsc.edu**.

Interview with Jérôme Guitton

Senior Software Engineer–Research & Development, AdaCore EU

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

I was an undergraduate student at Télécom Paris in 2001 when I first encountered Ada. One of the projects involved implementing an Ada component on the Java Virtual Machine and integrating it into a Java-based application. I naturally wondered why not just use Java instead? Java was a popular language at the time, and the students had a decent knowledge of it. Technically it would have been simpler to go all Java, but, from a pedagogical point of view, assigning Ada was clearly the right choice. After seeing the concept of class-wide types I came to understand object-oriented programming better through this small project than from my two years of using Java.

At that time I was planning on becoming a teacher, but an internship at AdaCore changed my mind and I joined the company's Paris office right after graduation. Since then I've been doing a lot of work in cross technologies for a variety of embedded targets. I've managed the port of GNAT Pro to various configurations including Wind River's VxWorks, SYSGO's PikeOS, and the XtratuM hypervisor from FentISS in Spain.

You have been involved with software verification from many perspectives, including debugger technology, formal methods, and coverage tools. Do you see any trends or developments offering hope that future systems can be less susceptible to the sorts of expensive "glitches" that are so common today?

A lot of these glitches come from a misuse of a component, that is, a misunderstanding of its interface: how to use it, what it does. To tackle this problem, something as simple as source code documentation—comments—has been a basic software engineering practice for decades. Even the simplest documentation is already an informal contract: it usually specifies how to interact with a module, it may describe its invariants, and it generally says something about data flow and control flow.

Executable preconditions, postconditions, and type invariants are really just a natural extension of this well-known practice. In a sense they provide a formal and unambiguous syntax and semantics for comments. But they not only help the human reader, they also have the benefit of generating run-time checks to verify that the program complies with these formal comments. Such features, known as contract-based programming, were a major addition to Ada 2012. During program testing the occurrence of assertion failures helps you detect errors early; compare that with the tedious debugging that would be required when informally specified contracts are violated.

Even better, these contracts can often be checked statically so that violations are detected before the program is run. Sophisticated static analysis and formal proofs are not just research topics anymore, they are now industrial technology. CodePeer has been around for several years, and SPARK 2014 has just been released. So we have the potential for such tools to see growing interest and usage in the future, especially in applications where reliability is critical. Debugging and informal documentation have obvious limitations; I can see static analysis and formal methods moving more and more into mainstream development.

When it is difficult to formally specify the complete behavior for a component, use cases can be a valuable technique. But these do not have to be specified through informal documentation; they can be captured as test cases and some can be realized as executable constructs through the Test_Case aspect implemented in GNAT Pro. The hybrid verification technique combining testing with formal proofs presents some interesting challenges that we are just starting to address, and I think we'll be seeing a lot of exciting developments in this area in the future.

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

I would ideally like to keep up with the many advances in human knowledge in this century, in many areas: contemporary art, music, literature, philosophy, and most especially mathematics. Modern algebra in particular is a fascinating field. Many ideas and results from this discipline do not yet have any application to software development but may do so in the future; being a witness to this process is already an exciting experience.

Photograph of Jérôme Guitton by Gary Matoso

GNAT Industrial User Day

This year's GNAT Industrial User Day will be held on Thursday, September 25, in Paris. Attendees will hear news about the latest tools and toolset features, product roadmaps, and practical tips from technology experts. Presentations will include updates on new technologies such as the upcoming qualifiable model compiler / code generator, and a selection of tutorials will help attendees understand and perfect programming techniques. As always, AdaCore staff will be on hand to answer questions, together with other industrial users who will share their experiences in using GNAT and Ada. For registration details and further information, please contact **events@adacore.com** or visit **www.adacore.com/gnatpro-day/**.

GNAT Pro Safety-Critical for Railway Applications

The GNAT Pro Safety-Critical development environment supports rail applications that need to meet the highest levels of safety certification. It includes run-time libraries specialized for use in safety-critical systems, as well as several tools for static analysis and testing. GNAT Pro Safety-Critical can be used in conjunction with other AdaCore products such as the SPARK Pro formal verification environment or the CodePeer advanced static analysis tool, providing a unique development framework that supports a wide range of verification activities.

In addition to a fully customizable run-time library, GNAT Pro Safety-Critical supplies several predefined run-time profiles (libraries corresponding to restricted feature choices). The Zero Footprint (ZFP) profile reflects an Ada language subset that does not require any Ada run-time routines, thus reducing the memory footprint to user code only. The Ravenscar Minimal profile implements the Ada Ravenscar tasking subset on top of ZFP. These profiles are intended for high-criticality applications, for example, those that need to be certified to Software Safety Integrity Level (SIL) 3/4. For lower levels of criticality, the Ravenscar Extended profile adds features such as exception propagation and stack overflow checking.

GNAT Pro Safety-Critical has been adapted to meet the needs of CENELEC standards for software development processes (EN 50128:2011, EN 50126:1999, and EN 50129:2003, for SIL 3/4), and a variety of certification-related material is available to supplement the product:

- A SIL 3/4 Independent Safety Assessor (ISA) certificate, confirming the Ravenscar Minimal profile's conformity to the CENELEC standard
- Qualification material for several product components:
  - the GNAT Pro compiler as a class T3 tool,
  - the GNATcheck coding standard checker as a class T2 tool,
  - the GNATmetric code metrics generator as a class T2 tool, and
  - the GNATtest / AUnit testing framework as a class T2 tool.

Qualification material is also available for several other tools that can be used in conjunction with GNAT Pro Safety-Critical:

- SPARK Pro's GNATprove as a class T2 tool to show proof of absence of run-time errors,
- the CodePeer static analysis tool as a class T2 tool for data and control flow analysis, and
- the GNATcoverage and GNATemulator dynamic analysis tools as class T2 tools for code coverage analysis.

GNAT Pro Safety-Critical has been used to develop rail systems certified to SIL4 of EN 50128. The product is available for x86 Windows and SPARC Solaris host development platforms, targeting PowerPC. For more information, please visit **www.adacore.com/gnatpro-safety-critical/**.

newsflash

Preventing future Heartbleeds

Security vulnerabilities such as the so-called Heartbleed bug are occurring at an increasing frequency but are readily preventable with appropriate languages, tools, and processes. Please visit **www.informationsecuritybuzz.com/heartbleed-glitch-deja-vu/** for AdaCore Vice President Richard Kenner's response to the Heartbleed bug, explaining how to develop security-critical software with confidence in its correctness.

Introductory Ada Course from AdaCore and Vector Software

A public course will be conducted during the week of September 8–12 in London, UK. Comprising both lectures and hands-on lab sessions, the course will provide a full introduction to programming in Ada and is ideal for software engineers joining a new or existing Ada project. Attendees will receive an introduction to some of the major features introduced in Ada 2012 (notably contract-based programming) as well as an overview of the formal verification techniques in SPARK 2014, and will use AdaCore's latest GNAT technology for the workshop exercises. For more information or to register for this course, please visit **www.adacore.com/training/general-ada-training1/** or contact **info@adacore.com**.

Conferences / Events ■ April–October 2014

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

Embedded Masterclass
April 8–10 / Birmingham, UK
Robert Dewar is giving a talk on using freely-licensed software in critical-embedded systems. AdaCore is a Silver Sponsor and exhibitor.
www.embedded-masterclass.com/

Tool Qualification Symposium
April 9–10 / Munich, Germany
Matteo Bordin is giving a talk on the economics of tool qualification. AdaCore is a sponsor and exhibitor.
www.validas.de/TQS/2014/

Civil Avionics International Forum
April 22–23 / Shanghai, China
AdaCore is a sponsor and exhibitor.
www.galleonevents.com/CAIF2014/en/home.html

High Confidence Software and Systems Conference
May 6–9 / Annapolis MD, USA
Yannick Moy is giving a talk on SPARK 2014: "Formal program verification for all".
cps-vo.org/group/hcss_conference

AUVSI's Unmanned Systems 2014
Association for Unmanned Vehicle Systems International
May 12–15 / Orlando FL, USA
AdaCore is an exhibitor.
www.auvsishow.org/auvsi2014/public/enter.aspx

Australian System Safety Conference 2014
May 28–30 / Melbourne, Australia
AdaCore is a major sponsor.
www.asssc.org/conf2014/

Ada-Europe 2014
June 23–27 / Paris, France
AdaCore is a major sponsor and exhibitor. Ben Brosgol, Yannick Moy, and Tucker Taft are presenting tutorials.
www.ada-europe2014.org/

TAP 2014
8th International Conference on Tests & Proofs
July 24–25 / York, UK
Johannes Kanig is presenting a paper, authored by AdaCore and Altran, on tool-assisted assumptions management.
www.tap2014.org/

GNAT Industrial User Day
September 25 / Paris, France
Plans for this year's event are summarized in an article on page 3 of this newsletter.
www.adacore.com/gnatpro-day/

33rd DASC
Digital Avionics Systems Conference
October 5–9 / Colorado Springs CO, USA
AdaCore is a sponsor and exhibitor.
www.dasconline.org/

IET Systems Safety and Cyber Security Conference
October 15–16 / Manchester, UK
AdaCore is a sponsor and exhibitor.
conferences.theiet.org/system-safety/

ACM SIGAda's HILT 2014
High Integrity Language Technology
October 21–24, 2014 / Portland OR, USA
AdaCore is a Platinum sponsor/exhibitor, and Tucker Taft is Program Chair.
sigada.org/conf/hilt2014/

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