New Tools for Dynamic Analysis of Embedded Systems

GNATemulator and GNATcoverage Help Reduce Testing and Verification Costs

Two new tools, available as GNAT Pro add-ons, will assist developers of embedded systems:

- GNATemulator, an efficient and flexible emulator solution that allows target-level testing without target hardware, and
- GNATcoverage, a non-intrusive coverage analyzer that works at both the source and object code levels.

These tools are particularly helpful in supporting development of safety-critical or high-security applications where full code coverage must be demonstrated.

GNATemulator runs on a host platform and currently supports applications written for PowerPC and SPARC (ERC32, LEON2, LEON3) processors. Based on the Open Source QEMU technology, the tool translates target object code into native host instructions, with the resulting code executing on the host. Developers can then run target code on their own host platforms easily and efficiently, avoiding the expense and inconvenience of managing and sharing actual boards.

GNATemulator is not intended as a complete time-accurate target board simulator but rather as a cost-effective and flexible testbed, installable on standard desktop machines. Based on the Open Source QEMU technology, the tool translates target object code into native host instructions, with the resulting code executing on the host. Developers can then run target code on their own host platforms easily and efficiently, avoiding the expense and inconvenience of managing and sharing actual boards.

GNATcoverage is the first non-intrusive coverage tool to fully support all levels of DO-178B safety certification, and it has full DO-178B qualification material available. At the object code level it provides statement coverage, decision coverage, and Modified Condition/Decision Coverage (MC/DC).

Unlike most current technologies, GNATcoverage works without requiring instrumentation of the executable. Instead, it runs directly on an instrumented version of GNATemulator. Drawing on research from the Couverture project, GNATcoverage can determine mathematically when MC/DC can be inferred from object code coverage.

GNATcoverage’s supported architectures include PowerPC and LEON.

Both GNATemulator and GNATcoverage are well suited to the continuous system integration techniques promoted by the Agile community. Multiple versions of either tool can be launched simultaneously, making it possible to parallelize testing and/or coverage analysis.

AdaCore Awarded Research Funds for New High-Integrity Frameworks/Tools

AdaCore has been awarded public funding to produce next-generation development platforms and tools for high-integrity systems, as part of the Hi-MoCo and pSafeCer research projects.

Hi-MoCo (High-Integrity Model Compilers) will provide an Open Source, highly tunable and qualifiable code generation framework for heterogeneous modeling languages such as Simulink and UML. Hi-MoCo will enhance the technology developed during the recent Gene-Auto project, which has already seen trial use by Airbus, Astrium Satellites, and Thales Alenia Space as a viable code generation solution for high-integrity systems. In addition to AdaCore, Hi-MoCo will involve the Estonian development firm IB Krates and the French research institute IRIT. Hi-MoCo is a two-year project funded by the European framework Eurostars.

The pSafeCer (Safe Certification) project will provide a highly interoperable development platform addressing requirements from multiple domains (aerospace, automotive, railway). AdaCore’s role is to produce the “Certifying Machine”, a certification artifacts management system designed to automate most of the configuration management activities required by standards such as DO-178. Funded by the European framework Artemis, pSafeCer is a two-year project involving more than 20 partners from the entire European Union.

AdaCore has successfully certified their Aerial Refueling Boom System (ARBS) on the A330 Multi-Role Tanker Transport (MRTT). The certification effort was simplified by the use of the qualified GNATcheck tool to verify conformance to the software coding standard required by the ARBS project. Verification of conformance was undertaken as part of the Software Verification Process required by DO-178B, Level A.

EADS CASA has chosen the GNAT Pro High-Integration Edition to implement the data exchange and air-to-ground data links systems for the nEUROn Unmanned Combat Air Vehicle (UCAV) demonstrator. EADS CASA selected AdaCore and GNAT Pro based on the Ada programming language’s suitability and AdaCore’s previous experience with high-integrity systems on aircraft including the Eurofighter, Airbus A330 Multi-Role Tanker Transport, and Airbus A400M.

New Structure, Website for ARA

The Ada Resource Association, a non-profit organization chartered to sustain and promote the Ada language, has introduced a new organizational structure based on sponsorship levels and has modernized and revised the Ada Information Clearinghouse website to better serve the needs of the Ada community.

AdaCore is a Platinum sponsor under the new ARA structure. For further information or to learn of sponsorship opportunities please visit www.adacore.org.
GNAT Pro 6.4

GNAT Pro 6.4 was released during Q1 2011, continuing AdaCore’s regular schedule of annual product upgrades. The major enhancements, summarized in the Autumn 2010 GNAT Pro Insider, include additional support for Ada 2012, improved code generation based on GCC 4.5, and a more flexible and more efficient project manager tool. With this latest release, GNAT Pro is now available for a new embedded platform, Wind River’s VxWorks MILS 2.1.x, and for updated versions of Wind River’s VxWorks 6 Cert (6.6.2) and SYSGO’s ElinOS (5.1). On the native side, GNAT Pro 6.4 supports Red Hat Enterprise Linux 6 on x86 (32- and 64-bit).

CodePeer 2.0

A major new version of the Code Peer source code analyzer/reviewer tool was released during Q1 2011. Principal improvements, summarized in the Autumn 2010 GNAT Pro Insider, include more efficient processing (including the ability to take advantage of multiple cores/processors) and fewer “false positives” (false alarms). CodePeer now comes with a number of complementary static analysis tools common to the GNAT Pro technology—a coding standard verification tool (GNATcheck), a program metric generator (GNATmetric), a semantic analyzer, and a document generator—that can be invoked through the GNAT Programming Studio (GPS) IDE.

Webinar Schedule

A webinar introducing the CodePeer 2.0 features is being presented by Tucker Taft (SofCheck) on Tuesday, May 3. To register, or to learn more about this webinar or any others that have been scheduled subsequent to the publication of this newsletter, please visit www.adacore.com/home/products/gnatpro/webinars/.

Spotlighting a GAP Member

Western Washington University (Bellingham, Washington, US)

In 2004, the faculty of the Computer Science department at Western Washington University conducted a comprehensive review of the undergraduate curriculum. At the heart of their considerations was the choice of programming language for the introductory programming courses. The faculty agreed that their current choice, C++, was not effective because students were spending so much of their time and intellectual effort on the idiosyncrasies of the language and much less time on problem solving and scalable programming practices.

At a time when many computer science departments were switching to Java, faculty at WWU remained unconvincend. They considered Java’s virtual machine concept and the “objects first” approach as major contributors to confusion among introductory students as to just what is really happening during program execution.

After much discussion, the department decided to change to Ada. “We felt that the clarity of Ada’s syntax and semantics would make it easier for students to learn and work with”, said Professor David Bover. “We also felt that Ada’s package and generic features would support and encourage students in the use of good, scalable programming practices”.

As a result of this change, the department was able to reduce its introductory programming sequence from four courses to three, in the trimester system, with improved results in terms of students’ problem solving and programming ability.

The use of Ada in the curriculum has now spread beyond the introductory programming sequence. The department has an upper division course on concurrent and distributed programming, where students learn about Ada tasking, C Pthreads, and the functional approach of Erlang.

Students are also using Ada in senior capstone projects on real-time control of a model railway system. To assist in understanding the complexities of that system, WWU faculty member Martin Osborne has developed a simulator that runs virtual trains on a virtual layout identical to the physical layout in the lab, including all switches and sensors. Use of the simulator during problem analysis has helped students understand the factors to be considered in system specification. In later project stages, the simulator assists in design verification and in more extensive software testing than would be practical with the physical system.

The department’s switch to Ada was initially opposed by the vast majority of students who saw far more job opportunities for software development in C++, Java, and later, C#. However, Professor Bover explained that their opinion has changed: “Students have come to realize that the computer science program at WWU is not about preparing them for a particular job. It is about preparing them for a career, and they see that using Ada at the start and at appropriate points later in the curriculum is providing them with an excellent basis for career-level training”.

For further information please visit www.cs.wwu.edu or contact Professor Bover at David.Bover@wwu.edu.

GAP (the GNAT Academic Program) provides GNAT technology and support to accredited academic institutions worldwide. To learn more, visit www.adacore.com/home/academia/.
After a number of fun experiments with a computer present for Christmas a long time ago, I decided that computing would be a good choice as a major subject in college. I developed a particular taste for software engineering, compilers, and everything related to the lower software layers of computer systems (hardware architectures, OS kernels, real-time and distributed systems). My first contact with Ada was in a course on computer operating systems with modules on distributed algorithms and synchronization. I liked Ada’s high-level features in these areas, and I was fortunate to have teachers who were fond of Ada and who offered convincing practical sessions.

After my undergraduate studies I spent three years as a PhD student on an R&D project at a French aircraft manufacturer. I was subsequently offered a position at AdaCore where I’ve been working with a fantastic team, mostly on the GCC part of the GNAT compiler.

Our base compiler technology needs to constantly evolve, to support new target platforms, to take advantage of modern processor facilities, and to generate better code. Regularly transitioning to recent versions of GCC is critical here, and a constructive interaction with the Free Software community is key in this process.

This interaction is dynamic and well structured. GCC is a large and complex piece of software, with strict policies regarding the integration of changes. Any change that we introduce is validated in accordance with community rules in addition to going through our own QA, and it is reviewed by other GCC maintainers if it affects parts we’re not officially responsible for. Some of us at AdaCore have formal maintenance responsibilities for areas of GCC outside the Ada front end, which reinforces our expertise and helps the project as a whole.

This relationship benefits everyone. The Free Software community receives a modern Ada front end and help from our expertise on other parts of the GCC technology. AdaCore gains access to constantly-improving code generators for a wide range of architectures while strengthening our own QA, and everything gets tested on a very large scale.

I see several trends. First, I expect safety certification to see increased attention in fields where life-critical device control gets implemented in software. I’m thinking of the medical and the automotive domains for example.

Second, as safety-critical systems grow in complexity and application scope, certification will need to account for security issues and for advances in software development methodologies. Building large systems today is different from how we did things ten or twenty years ago; we’re seeing more and more uses of Object-Oriented Technologies, Model-Based Development, and Formal Methods. The DO-178C effort underway in the avionics industry is responding to these sorts of changes.

Leveraging open communities is a very attractive option to support these trends. The Open-DO initiative was launched with exactly this goal, to promote shared development and maintenance efforts for certification-oriented tools. I see the potential for a powerful movement, with constructive and mutually beneficial interactions similar to what I talked about earlier in connection with AdaCore and GCC.

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

I like to play chess, even though I’m not about to challenge any Grandmasters. I also try to jog on a regular basis. In part for the physical exercise, but also (in contrast with my much more structured approach as a software engineer) because it gives me the opportunity to go out without deciding my destination or route in advance.
GtkAda bindings have been upgraded to support Gtk+ version 2.16.6 and beyond. Bindings to a dozen new widgets are now available, ranging from simple (Gtk.Scale_Button,Gtk.Volume_Button) to complex (for example GtkAssistant, which allows adding page-by-page wizards to applications).

Integrating OpenGL areas in applications is now supported on Windows as well as on UNIX and Linux.

Interface to the Cairo graphics library
GtkAda now includes an interface to Cairo, an extensive 2D drawing library that is used as the rendering back end. Cairo can be used to trace lines, polygonal and elliptic shapes, Bézier splines, solid and gradient fills, antialiased text, and images, with an API similar to the drawing operators of PostScript, PDF, and the HTML-5 Canvas. Cairo also provides clipping, image compositing, translucency, and matrix transformations (rotating, scaling, shearing, etc.).

Users can exploit Cairo's support for resolution-independent graphics in GtkAda applications, either by drawing on existing widgets (for instance a Gtk.Drawing_Area), or by employing it as the engine for drawing their own widgets. In addition to on-screen rendering, the Cairo binding supports off-screen rendering, and exporting to PNG files.

Printing
The Gtk+-printing API is now accessible in GtkAda through a high-level, object-oriented API. An application can bring up the Gtk+ print dialog on UNIX and Linux, and the native print dialog on Windows, and then launch print jobs. Using these interfaces, an application may print text or graphics generated with the Cairo API.

AdaCore at Wind River Conferences
AdaCore is a Silver Sponsor for the Spring 2011 Wind River Aerospace and Defense Regional Conferences in the US: April 21 (Dallas, TX), April 26 (Huntsville, AL), May 17 (Washington, DC), May 25 (Boston, MA), and June 7 (Manhattan Beach, CA). www.windriver.com/announces/ad-rc-2011/

AdaCore is also a sponsor for Wind River’s Embedded Konferenz, May 24 in Stuttgart, Germany. For more information (in German): www.embedded-konferenz.de/