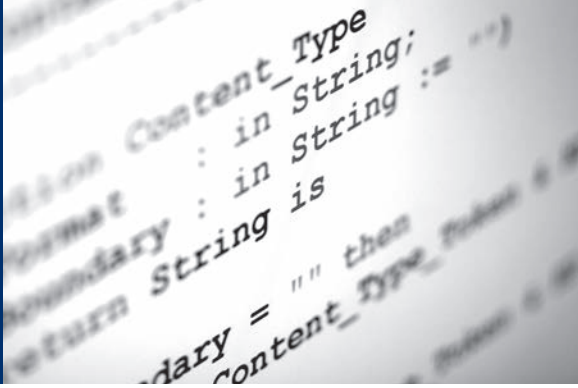
GNAT Pro insider
an AdaCore Publication
Autumn/Winter 2014–2015
www.adacore.com



Happy 20th Birthday, AdaCore!

This special issue focuses on the history of AdaCore in celebration of the 20th anniversary of the company's founding. In July 1994 three members of New York University's Computer Science Department—Robert Dewar, Richard Kenner, and Edmond Schonberg—got together in the New York version of a Silicon Valley garage: Robert Dewar's loft apartment near Union Square. There they made the decision to form a business as an offshoot of the GNAT project that was developing an Ada 95 compiler at NYU. The company was named Ada Core Technologies, and its first employee—Gary Dismukes—soon joined. The next major step came in January 1996 when ACT Europe was formed in France by Cyrille Comar and Franco Gasperoni, computer scientists who had previously done post-doctoral work at NYU. The years since then have witnessed steady growth, a new name—*AdaCore*—and worldwide attention to the company's leading-edge Ada technology. This issue recounts the history of AdaCore's major products and looks at some key milestones over the company's first twenty years.

AdaCore staff form "20" for company birthday

AdaCore US has moved

The company's US headquarters has moved uptown in Manhattan. Expanding into larger office space, the company has relocated to:

150 W. 30th Street, 16th Floor
New York, NY 10001

The telephone and FAX numbers (listed on page 6 of this newsletter) remain the same.

contents

A Brief History of GNAT Pro

GNAT began as a research project led by Robert Dewar and Edmond Schonberg at New York University in the early 1990s, sponsored by the US government's Ada Joint Program Office. The goal was to build a prototype compiler for the first major revision of the Ada language (Ada 95) to verify its implementability and also to ensure that, when the standard was approved, usable implementations would be available on several common platforms. A very early decision was to base the compiler on the GNU compilation system (gcc), and to follow the tenets of the Free Software Foundation to ensure its wide distribution in research and academic circles. This decision has been key to AdaCore's success in fielding compilers for a wide variety of target architectures, and in taking advantage of the latest code optimization developments from the GNU community. AdaCore also uses the GNAT front end to create compilers for targets that are not supported by the gcc back end, including the AAMP processor and hardware simulators.

The name GNAT was originally an acronym for GNU NYU Ada Translator, reflecting the technology's origins. The GNAT name gave rise to the tongue-in-cheek slogan "There is only one bug in GNAT", but because of the acronym people sometimes needed to be reminded that GNAT is indeed a full-fledged compiler to object code and not just a front-end translator to C. The GNAT compiler became AdaCore's first product when the company was launched in 1994. To simplify the nomenclature, the product that was known as the GNAT Professional version was renamed GNAT Pro in 2001.

The GNAT technology has evolved over the last twenty years so that it now implements all versions of the language standard, and it has grown from simply a compiler and debugger to a comprehensive toolset and full development environment. It undergoes continued enhancements, with an annual major release on dozens of platforms, both native and cross, and is subject to rigorous QA including nightly regression tests comprising millions of lines of code. The GNAT front end and most of the run-time library code are themselves written in Ada, so the company makes full use of its own products.

AdaCore and Safety Certification

AdaCore's support for safety-critical systems originated with a specialized set of run-time libraries ("profiles") provided with the GNAT Pro High-Integrity Edition for DO-178B (now known as *GNAT Pro Safety-Critical*) and has steadily expanded over the years to comprise a wide range of products and services. During this time company personnel have played a direct role in the development of some of the current software certification standards; for example Cyrille Comar participated in the working group that defined DO-332, the Object-Oriented Technologies and Related Techniques supplement to DO-178C.

Certifiable Run-Time Libraries

To best meet the system/software certification requirements, coding standards for safety-critical software typically prescribe language subsets (removing features with complex run-time semantics, for example), and AdaCore has thus designed several restricted Ada libraries known as the *High-Integrity profiles*. These have evolved over the years and currently include a minimal library (Zero Footprint Profile, or ZFP) and several others that extend ZFP with features such as exception handling, ARINC 653 support, and the Ravenscar tasking subset.

The High-Integrity profiles have been used in a wide range of safety-certified systems, on target platforms including PowerPC and ARM bare board as well as various RTOSes (VxWorks 6 Cert, VxWorks 653, VxWorks MILS, LynxOS-178, and PikeOS). Supplementing the GNAT Pro Safety-Critical product, certification material can be made available for these profiles (as well as for some other run-time libraries) on a platform-dependent basis, demonstrating compliance with the highest levels of the relevant domain-specific standards. These include DO-178B/C for avionics; CENELEC EN 50126, EN 50128, and EN 50129 for rail systems; and ECSS-E-ST-40C and ECSS-Q-ST-80C for space systems.

Qualified Tools

AdaCore has developed a variety of tools that have been qualified, or are in the process of qualification, in compliance with the software standards in several domains, and the associated qualification material is available as a supplement to GNAT Pro Safety-Critical.

A summary is displayed in the table below.

Tool	Avionics: DO-178B	Rail: EN 50128
GNAT Pro Compiler		Class T3
GNATcheck	Verification Tool, Level A	Class T2
GNATcoverage with GNATemulator	Verification Tool, Level A	Class T2, for code coverage analysis
GNATcoverage with hardware probe	Verification Tool, Level B	Class T2, for code coverage analysis
CodePeer	Verification Tool, Level A	Class T2, for data & control flow analysis
GNATmetric		Class T2
GNATtest/AUnit		Class T2
GNATprove (in SPARK Pro)		Class T2, to show absence of run-time errors
Tool Qualification Summary		

Additionally, qualification is on the roadmap for AdaCore's new QGen code generator and model verifier for Simulink and Stateflow models, as a DO-178C tool at TQL-1 (similar to a DO-178B development tool).

Supplemental services

To complement the GNAT Pro Safety-Critical development environment, AdaCore offers several specialized certification-related services. As an add-on, AdaCore can prepare a traceability package to support the activity described in section 6.4.4.2b of the DO-178B and DO-178C standards: for software at Level A, if object code is generated that is not directly traceable to the source code then "additional verification should be performed on the object code to establish the correctness of such generated code". AdaCore has supplied such traceability analyses / object code verification to several clients in the aerospace industry.

Another service is a specialized subscription option for customers who need access to defect corrections on defined release branches. Known as a "sustained branch subscription", this newly introduced service is especially useful in certification contexts where compiler code generation problems, even when detected long after certification, must be analyzed to assess the impact on certified code.

In summary, safety-critical systems are an excellent match for the Ada and SPARK languages, and AdaCore's product line has evolved to meet that sector's requirements. Continued enhancements to the GNAT Pro Safety-Critical environment, including further tool qualification, is planned.

A Brief History of SPARK Pro

The SPARK technology—the language and its supporting tools—originated in the 1980s as a research project in formal proofs of program correctness at the University of Southampton (UK). SPARK was based on Ada 83, with contracts (annotations) expressed as specially written comments. The SPARK technology was acquired and commercialized by Praxis (now Altran, and so referenced below), and the SPARK language was later extended, and its tools upgraded, to reflect the subsequent Ada 95 and Ada 2005 standards.

SPARK Pro was conceived in 2008, a result of the partnership between AdaCore and Altran. In an effort led by Arnaud Charlet from AdaCore and Roderick Chapman from Altran, SPARK Pro integrated the SPARK tools within AdaCore's GPS IDE. The new product was launched in 2010, with subsequent releases in 2011 and 2013 that brought a variety of enhancements. These included the SPARKbridge tool that targets SMT solvers through a ViCToR interface and optional flow analyses, and the Riposte tool that improves user interaction by generating counter-examples for incorrectly specified contracts.

In 2010 AdaCore and Altran started a research project led by Yannick Moy from AdaCore, which resulted in a major upgrade to the SPARK technology. Its original goal was to extend the SPARK toolset and SMT solvers to a large subset of Ada, replacing the SPARK annotations by the contract syntax (e.g., preconditions and postconditions) that was being added to Ada. However, the new toolset, based on the Why3 intermediate language and the Alt-Ergo SMT solver, turned out to be far more powerful than the existing SPARK proof engine. As a consequence the SPARK tools were completely reimplemented using the new technology; the large subset of Ada 2012 that could be handled was called SPARK 2014.

SPARK Pro 14.0 in 2014 marked the first release of the new language and tools, fully compatible with Ada 2012 and integrated into AdaCore's IDEs. Continued enhancements are planned, both through the AdaCore-Altran partnership and the new AdaCore-Inria ProofInUse joint lab, for example to make the formal verification tools more automated and more interactive. Simultaneously, work is in progress on techniques for exploiting SPARK 2014's support for combining formal verification and testing in the same program. This "hybrid verification" approach allows developers to selectively apply formal verification to only those portions of a program where it is feasible and cost effective. It also facilitates the introduction of SPARK incrementally into a project that is already using Ada.

Over the course of its history SPARK has been used in a wide range of safety-critical and high security military and civilian programs including iFACTS, the current generation of tools for en-route air-traffic control in the UK, and Tokeneer, a secure enclave protection system sponsored by the NSA in the US. SPARK's benefits in terms of low defect rates have been demonstrated quantitatively, and SPARK can be used to prove properties ranging from absence of run-time errors to compliance with formally specified requirements. As the technology has evolved, SPARK has made formal verification a practical and cost-effective approach for achieving confidence in critical software.

The interview in this special history issue is with two of AdaCore's founders, providing their insights on the company's first twenty years and what to expect in the future.

Interview with Robert Dewar, AdaCore President and Cyrille Comar, AdaCore Managing Director

Robert, how did you envision AdaCore's future when you and your colleagues founded the company in 1994? How did your predictions turn out, and were there any surprises along the way?

ROBERT DEWAR: GNAT was originally developed at NYU, but we always had in mind turning it into a commercial product. Since it was distributed under a Free Software License (the GPL and modified GPL), any company could have picked it up. But it soon became clear that no one else would do that, so we founded Ada Core Technologies.

Initially we thought that our business would be devoted to native compilers only. Several other companies were developing cross products, and it seemed a difficult and crowded market to compete in. We started with a contract with Silicon Graphics (SGI) to make GNAT the official Ada compiler for IRIX machines, and we hoped to bring in similar contracts with other hardware manufacturers. Although we had some success—for example a port of GNAT to VMS for DEC Alpha and later for Itanium—it became apparent that these sorts of opportunities would be limited. However, we developed a number of other native compilers—the gcc technology was key here—and those sales together with the SGI contract kept us going in the early years. Our company was always run on revenue, we never had any outside funding.

As is entirely appropriate, our product direction then was heavily dictated by our users. Several customers really wanted to see a GNAT cross-compiler for embedded use, particularly under Wind River's VxWorks, and they waved large enough checks to take us in that unexpected direction from then on. Currently most of our business is in the embedded systems domain, though of course we still serve the native market as well.

The next major development for us was the move into the safety-critical arena. Our first such application was the Canadian Space Arm for the International Space Station nearly fifteen years ago. This was developed under OS/2, and it marked the first usage of our "zero footprint" library. Subsequently we became involved in several large-scale safety-critical avionics applications and then in high-security systems, which led us in the direction of SPARK and formal methods.

Ada for us is an overriding metaphor for highly reliable software. The core of our business is and will remain closely tied to the Ada language as our name implies, but our larger interests will be in providing technology for our customers to develop, verify, and maintain critical systems. The world around us is full of software that consistently fails to meet this requirement, and our mission is to show that reliability in software is both achievable and affordable.

Cyrille, any predictions for the company's directions in the next five to ten years?

CYRILLE COMAR: This question comes at an interesting time for two reasons: first, the whole company has recently completed a very successful introspective exercise to enunciate and flesh out its "core values" and current strategic directions. Second, I just happen to be reading *The Black Swan, the Impact of the Highly Improbable* by N. N. Taleb. This book does a great job at explaining why predictions are almost always completely off the mark, unless they are phrased with sufficient ambiguity so that they can be reinterpreted after the fact to be compatible with what actually happened. Since I do not have Nostradamus's skills, I will concentrate on the current directions that are unlikely to change in the years to come. These fall into several categories.

The first area is verification technology. Our expertise lies in creating and supporting tools that help build, verify and maintain complex and dependable software-based systems. If our original offering was limited to tools around pure Ada programming, we are now convinced that in order to reach higher levels of reliability, we need to help popularize mathematically based static verification techniques such as the proof capabilities of SPARK 2014.

The second area is tool support throughout the full software lifecycle. We need to connect our traditional development toolsets with higher level languages and modeling environments in order to provide guarantees throughout the complete chain of software production. Our new tool QGen is a first step in this direction. Modern systems are best built through a combination of techniques and languages, and thus we will continue to investigate interoperability and strongly support openness in all aspects of software development.

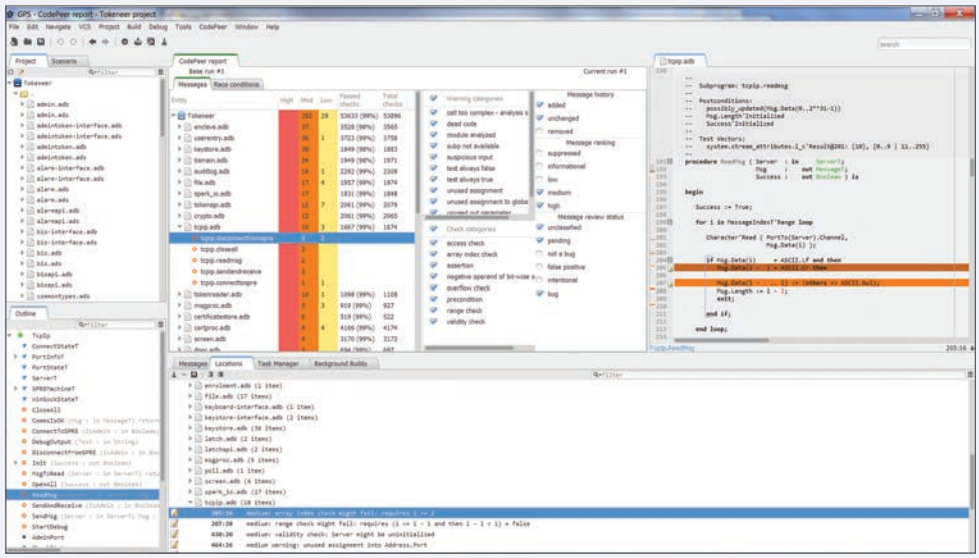
The third area is our customer support infrastructure. Our main way of operating is through annual subscriptions that combine software products, maintenance, customer-suggested enhancements, and proactive internet-based assistance that allows complete tracking of all requests and defect reports. A continuing challenge is to adapt this model to a constantly changing technological landscape (higher bandwidth, increased need for security, etc.) while supporting customers who need stability in order to maintain their very long-lived systems such as for aircraft or trains. We are thus committed to a two-fold process: offering long-term maintenance capabilities on our toolchains so that our users can preserve their software investment, while also helping them transition to and exploit the innovations embodied in our evolving products and technologies.

Each of these is a natural direction for the company, drawing on our strengths and experience. AdaCore has grown steadily and successfully over the past twenty years, and I fully expect that trend to continue into the foreseeable future.

A Brief History of CodePeer

The static analysis technology used in CodePeer got its start in 2001, when Bob Duff and Tucker Taft realized that a highly optimizing compiler could figure out many interesting and deep properties of the program it was compiling, but that it would then keep most of them a secret. What would happen if an advanced compiler optimizer was designed with the sole purpose of alerting programmers to potential problems in their code, including logic errors? An advanced Ada-oriented optimizer could identify places where variables were likely being read before being initialized, where array indices were likely out of bounds, where pointers that were likely null were being dereferenced, and similar problems. Overall, such an optimizer could identify many kinds of bugs and vulnerabilities before the program was executable (working even on partial programs), saving considerable development and debugging effort.

They had visions of turning their idea into a commercial product and, with major contributions from Sheri Bernstein and Mireille Gart, realized this goal while at SofCheck, initially naming it *Statcheck* and then *Inspector*. In 2008 AdaCore and SofCheck began a joint effort to enhance the product, and the newly named *CodePeer* was launched by AdaCore in 2010. Since then it has been extended to provide precise analysis of floating point arithmetic and numeric libraries, support for very large applications, analysis at multiple levels, support for contract-based programming, and enhanced user-interface features such as backtraces and tooltips on possible values. By combining sophisticated data- and control-flow analysis techniques with AdaCore's GNAT front-end, project files, and integrated development environments (GPS and GNATbench), CodePeer has become the most advanced and most precise static analysis tool for Ada. As it continues to evolve, CodePeer still retains its original characteristics: a sound, optimizer-based approach to efficient, scalable, and precise analysis.



CodePeer screenshot

A Brief History of the GPS and GNATbench IDEs

In the late 1990s, with AdaCore's product line broadening to include a growing number of tools, the company realized that a visual Integrated Development Environment was becoming a requirement. The initial offering came in 2000 with an Emacs-based solution called *Glide*, but its rudimentary graphics and Unix-centrism made it apparent that a more modern and user-friendly approach would be needed. At the same time, a project was underway at AdaCore to develop GtkAda, an Ada interface to the Gtk+ graphics library. GVD, a visual front end to the debugger, was implemented through GtkAda and was nearly an IDE; all that was needed was a facility that allowed users to modify and save files, launch compilations, and explore the project hierarchy.

The first version of the GNAT Programming Studio was released in 2002. The choice of a name with the *GPS* acronym was intentional; the IDE's main purpose is to help users navigate through the source files and the various kinds of information that the toolset can derive about their programs. GPS displayed a multiple-document interface, editors with unbounded undo/redo, graphical project trees, and the debugger integration inherited from GVD. As with the vast majority of the company's software, it was programmed in Ada.

Over the years the feature set of GPS has grown, with key contributions from Arnaud Charlet, Emmanuel Briot, and Nicolas Setton. Early enhancements included an engine for source navigation, a lightweight integration of version control systems, and support for C and C++. In 2004 came a technological breakthrough with the introduction of Python as a scripting language. This turned out to be a useful complement to the Ada engine, making it easier to write more advanced regression tests, and allowing users to develop plugins to customize GPS's behavior.

Meanwhile, in the 2003–2005 timeframe the Eclipse IDE was gaining traction in the industry. AdaCore's partner Wind River had decided to base their Workbench IDE on the Eclipse platform, and it was important to have GNAT Pro accessible through this interface. The result was the Eclipse / Workbench plugin known as *GNATbench*, launched in 2006. This IDE was developed at AdaCore with major contributions from Pat Rogers, Quentin Ochem and Philippe Gil. Technically, the IDE engine is written in Ada and shared between GPS and GNATbench. The Java code in GNATbench calls the Ada code using a high-level binding generated by the Ada/Java Interfacing Suite (AJIS).

GPS and GNATbench continue to evolve, taking advantage of the latest technological developments while staying faithful to their original design philosophies: for GPS, providing a lightweight and powerful IDE for the AdaCore tools; and for GNATbench, smoothly integrating Ada support within the Eclipse framework.

Workshop on Medical Device Software Security Cites Benefits from Choosing an Appropriate Programming Language

SPARK helps prevent memory safety errors

A November 2014 workshop convened an invited group of experts from industry, academia, and government to formulate elements of a "building code" that could significantly reduce the vulnerability of medical devices to malicious attacks; Tucker Taft from AdaCore was one of the participants. Sponsored by the IEEE Computer Society's Cybersecurity Initiative, with participation from the NSF's Trustworthy Health and Wellness project, the workshop has published a report (see **sites.google.com/site/bcformdss/home**) summarizing its findings. One of its conclusions was that the choice of programming language plays a significant role in preventing vulnerabilities such as memory safety errors, and SPARK was cited as supporting key elements of the building code.

The workshop identified a number of memory safety errors: buffer overflow, null pointer dereference, pointer usage after being freed ("dangling reference"), use of uninitialized memory, and illegal free (i.e., freeing an already-freed pointer or a non-malloced pointer). The full Ada language prevents the first two errors and with appropriate encapsulation of uses of Unchecked_Deallocation can also prevent dangling references and illegal free. The SPARK subset of Ada prevents all of the errors; indeed, dynamic allocation is prohibited in SPARK. SPARK also directly supports one of the other "building code" elements: automated analysis of programs for critical properties. The verification technology embodied in the SPARK language and tools can provide assurance that formally specified security policies are met.

Group photograph on page 1, Robert Dewar and Cyrille Comar on page 3, and Yannick Moy on page 6 by Gary Matoso; CubeSat photograph on page 5 by Carl Brandon; page 6 photograph of Arnaud Charlet presenting at GNAT Industrial User Day by Jamie Ayre.

GNAT Pro 7.3

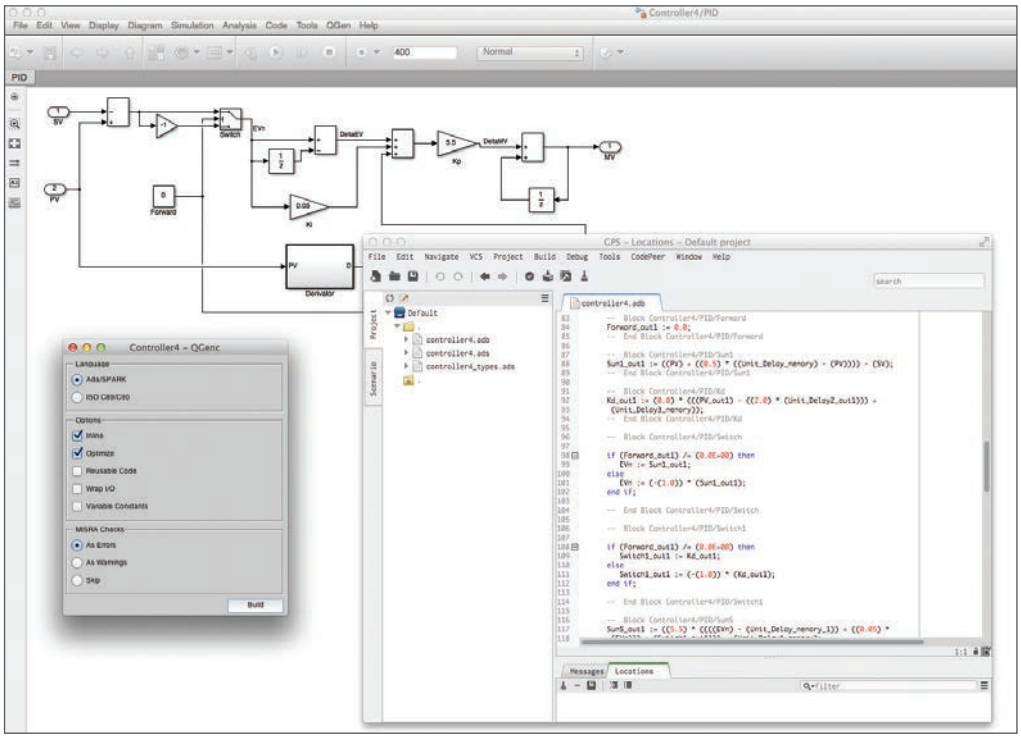
The latest version of the GNAT Pro development environment will be released during Q1 2015. GNAT Pro 7.3 has upgraded to the gcc 4.9 back end and the gdb 7.8 debugging technology, and it incorporates over 175 new features. These include improved diagnostic messages, fine-grained control over the treatment of warnings, extended support for non-default endianness, a certifiable math library for bare board platforms, support for large files on 32-bits systems, improved handling of inlining, overflow checks enabled by default, enhanced code generation and debugging capabilities, and support for aggregate projects in most GNAT Pro tools. For more efficient performance a number of tools, including GNAT2XML and GNATmetric, can take advantage of parallel and incremental processing, and GNATtest now supports the stubbing of units.

Public Ada Course, Spring 2015

An introductory Ada course will be conducted in Paris during the week of March 30–April 3. Combining live lectures with hands-on lab sessions using the latest AdaCore tools, this course introduces software developers to the principal features of the Ada language with a special focus on embedded systems. Attendees will learn how to take advantage of Ada's software engineering support, including the contract-based programming features in Ada 2012, to produce reliable and readable code. No previous Ada experience is required. For more information please visit **www.adacore.com/public-ada-training/**.

QGen

QGen, a qualifiable and customizable code generator and model verifier for Simulink and Stateflow models, will be launched during Q1 2015. This tool can generate SPARK/Ada and MISRA C, producing readable, traceable, and efficient code. It is particularly suited for developing and verifying high-integrity real-time applications, especially where safety certification is required. QGen handles around 100 blocks, and support for Stateflow models is expected during Q2 2015. The tool's static model verifier detects run-time errors such as integer overflow and division by zero, it can find logic errors such as dead execution paths, and it can also verify functional properties through Simulink Assertion blocks. Qualification material for QGen will be available for standards such as DO-178C (avionics), EN 50128 (rail), and ISO 26262 TCL3 (automotive), and the tool is highly configurable thanks to its visible intermediate representation. A QGen demo is available at **www.adacore.com/qgen_demo**.



QGen screenshot

A Brief History of AdaCore and Education

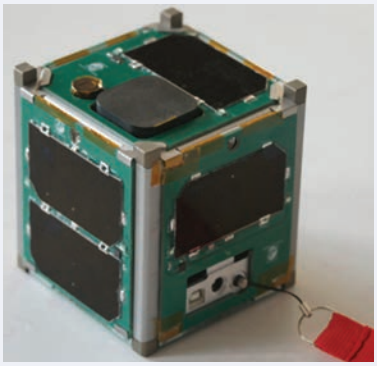
AdaCore has been active in education through three activities: by promoting the usage of Ada and SPARK in academia, by offering live courses on the Ada and SPARK languages and the AdaCore tools, and by supplying online interactive courses for self-instruction.

Encouraging Ada usage in academia has always been a high priority at AdaCore. An immediate example, and one that has been standard policy since the outset, is the company's commitment to providing a GPL edition of the GNAT technology: a no-cost version with a license that is appropriate for academic usage. With that as a basis, AdaCore has established the GNAT Academic Program (GAP) which provides support services (e.g., answering language or tool questions) to professors who are using Ada or SPARK for teaching or research. The GAP initiative was launched in 2004 and currently includes more than 200 academic institutions worldwide. Under the GAP program universities have developed courses in real-time / embedded systems development and have completed projects including robotics, formal verification, secure separation kernels, and satellite control.

Live training has been an important AdaCore service that complements the company's product offering. Courses comprise lectures and hands-on workshops, and cover topics including the Ada and SPARK languages, the GPS IDE, and specific AdaCore tools. Ada language courses are conducted publicly and are also available for delivery at customer sites; in the latter case they can be tailored to meet customer requirements.

The most recent addition to AdaCore's educational activities is AdaCore University. Launched in 2013, this free online resource comprises a series of modules that combine pre-recorded lectures, accompanying visual aids (viewgraphs), and interactive exercises to test students' understanding. It consists of a growing collection of modules, with topics such as the Ada concurrency features planned for early 2015.

Coming from the academic community, AdaCore's founders recognized the importance of education in building an awareness and understanding of advanced technology, and the company's range of activities in this area continue to underscore its priority. For additional information please visit **www.adacore.com/academia/**, **www.adacore.com/training**, and **university.adacore.com**.



CubeSat built by GAP member Vermont Technical College

newsflash

GNAT Industrial User Day 2014

On September 25, 2014, AdaCore hosted its annual *GNAT Industrial User Day* in Paris where customers, partners, and academics learned the latest news about the company's current and planned product offerings and activities. Topics included Ada 2012, Ravenscar and SPARK running on an Atmel ARM M4 using a Tetris example, the new QGen qualifiable code generator and model checker for Simulink models, and AdaCore University. Additionally, three customers—Ansaldo (railway), Rolls Royce Controls and Data Services (avionics) and Omlis (secure mobile payment systems)—each gave a brief talk describing how they use Ada, SPARK, and AdaCore tools to build their systems. The slides from the event are available online at **www.adacore.com/gnatpro-day/2014-gnatpro-day-slides**. 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 2015.

Yannick Moy Recognized as One of Embedded Computing Design's "Top 4 under 40"

Yannick Moy, a Senior Software Engineer at AdaCore, has been selected by Embedded Computing Design as one of their "Top 4 Under 40", an award that recognizes the contributions of engineers under the age of 40. Dr. Moy is a key member of AdaCore's static analysis technology team, which is responsible for products like CodePeer and SPARK Pro that detect bugs and vulnerabilities or verify safety/security properties. Dr. Moy leads the development of SPARK 2014 and has made presentations on this topic in articles, conferences, classes, and blogs (in particular **www.spark-2014.org**). He was recently appointed co-director of the ProofInUse laboratory, a joint effort between AdaCore and INRIA (**www.spark-2014.org/proofinuse**). To read the full interview in ECD, please visit **http://adaco.re/8f**.

AdaCore Technology Days 2015

Based on the positive reaction to the GNAT Industrial User Day in Paris, a similar event is planned for the US and will be expanded to two full days to allow additional time for product demos and training. These *AdaCore Technology Days* will take place in early November, 2015, in the metropolitan Boston area. For up-to-date schedule and registration information, please visit **www.adacore.com/techdays2015**.

Conferences / Events ■ December 2014 – May 2015

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

ARM Tech Symposia 2014
December 4, 2014 / Paris, France
AdaCore is exhibiting at this event.
www.arm.com/about/events/arm-tech-symposia-2014-europe.php

2014 Workshop on Spacecraft Flight Software
December 16–18, 2014 / Pasadena CA, USA
AdaCore is a sponsor of this event, and Tucker Taft is giving a talk "A Complete High-Integrity Software Development Stack".
flightsoftware.jhuapl.edu/

7th International Automotive Electronics Technology Expo: CAR-ELE Japan
January 14–16, 2015 / Tokyo, Japan
AdaCore is an exhibitor at this event, represented by ITAccess.
www.car-ele.jp/en/

RTECC Santa Clara
January 22, 2015 / Santa Clara CA, USA
AdaCore is an exhibitor at this event.
www.rtecc2015.com/

ProofInUse Kickoff
February 2, 2015 / Paris, France
Launch of the ProofInUse research project.
www.spark-2014.org/proofinuse/kickoff

23rd Safety-critical Systems Symposium SSS 2015
February 3–5, 2015 / Bristol, UK
AdaCore is an exhibitor at this event.
scsc.org.uk/e300

Certification Together
February 24–26, 2015 / Toulouse, France
AdaCore is exhibiting at this event.
www.certification-together.com

Embedded World 2015
February 24–26, 2015 / Nürnberg, Germany
AdaCore is an exhibitor at this event.
www.embedded-world.de/en/

3rd Scandinavian Conference on System and Software Safety
March 24–25, 2015 / Stockholm, Sweden
Quentin Ochem is presenting a paper "Reducing the Cost of Defensive Code—the Ada 2012 Approach".
safety.addalot.se/

Public Ada Course at AdaCore
March 30–April 3, 2015 / Paris, France
Please refer to the item on page 5 of this newsletter for information.
www.adacore.com/public-ada-training/

17th International Real-Time Ada Workshop
April 20–22, 2015 / Bennington VT, USA
Robert Dewar is Workshop Chair, and AdaCore is hosting this event.
www-users.cs.york.ac.uk/~andy/IRTAW2015/

Embedded Systems Conference
May 6–7, 2015 / Boston MA, USA
AdaCore is an exhibitor at this event.
www.embeddedconf.com/boston/

AUVSI's Ummanned Systems 2015
May 5–7, 2015 / Atlanta GA, USA
AdaCore is an exhibitor at this event.
www.auvsishow.org/auvsi2015/public/enter.aspx

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

150 W. 30th Street, 16th floor
New York, NY 10001, 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