Inside AdaCore

Autumn/Winter 2015–2016

- Drone Safety
- Interview with Martyn Pike
- Upcoming Releases:
  GNAT Pro 7.4, CodePeer 3.1,
  SPARK Pro 16.0, QGen 2.1
- Ada Courses in UK and US
- Automating Test Case
  Generation with GNATtest

AdaCore Participating in Certified Drone Autopilot Project

AdaCore is participating in CAP2018 (Certified AutoPilot 2018), a project sponsored by the French government with the goal of producing a reliable, safe, and secure autopilot for commercial and leisure drones. Project partners Sogilis and Squadrone Systems will be developing the autopilot using AdaCore's GNAT Pro compilation environment, GNATcoverage test coverage analyzer, and SPARK Pro formal verification tools. The autopilot will run on a chip with an ARM Cortex-M processor, using a GNAT Ravenscar run-time library for safe tasking. The GNATcoverage tool will provide coverage analysis without instrumenting the code.

A key goal is to develop and verify the autopilot software so that it can meet the Level A objectives of avionics standard DO-178C. Compliance will provide high confidence in the quality expected of such software, and it will offer a competitive advantage in an environment where drone safety is attracting increased attention and where regulations are expected to be strengthened. For example the European Aviation Safety Agency (EASA) has proposed a regulatory framework based on associated risk, where drones at the highest risk level will need to be certified: their airworthiness will need to be demonstrated in the same way as is done for manned aircraft.

The SPARK Pro technology will provide mathematical assurance of specified safety and security properties for the autopilot, such as absence of run-time errors. It will also help reduce development costs by using formal verification as an alternative to certain types of testing, as allowed by DO-178C's Formal Methods supplement (DO-333). The autopilot will be available under a Free Software license, and the certification evidence will be commercialized by Sogilis. As implied by its name, the CAP2018 project is scheduled to be completed by 2018.

Using SPARK for Drone Safety

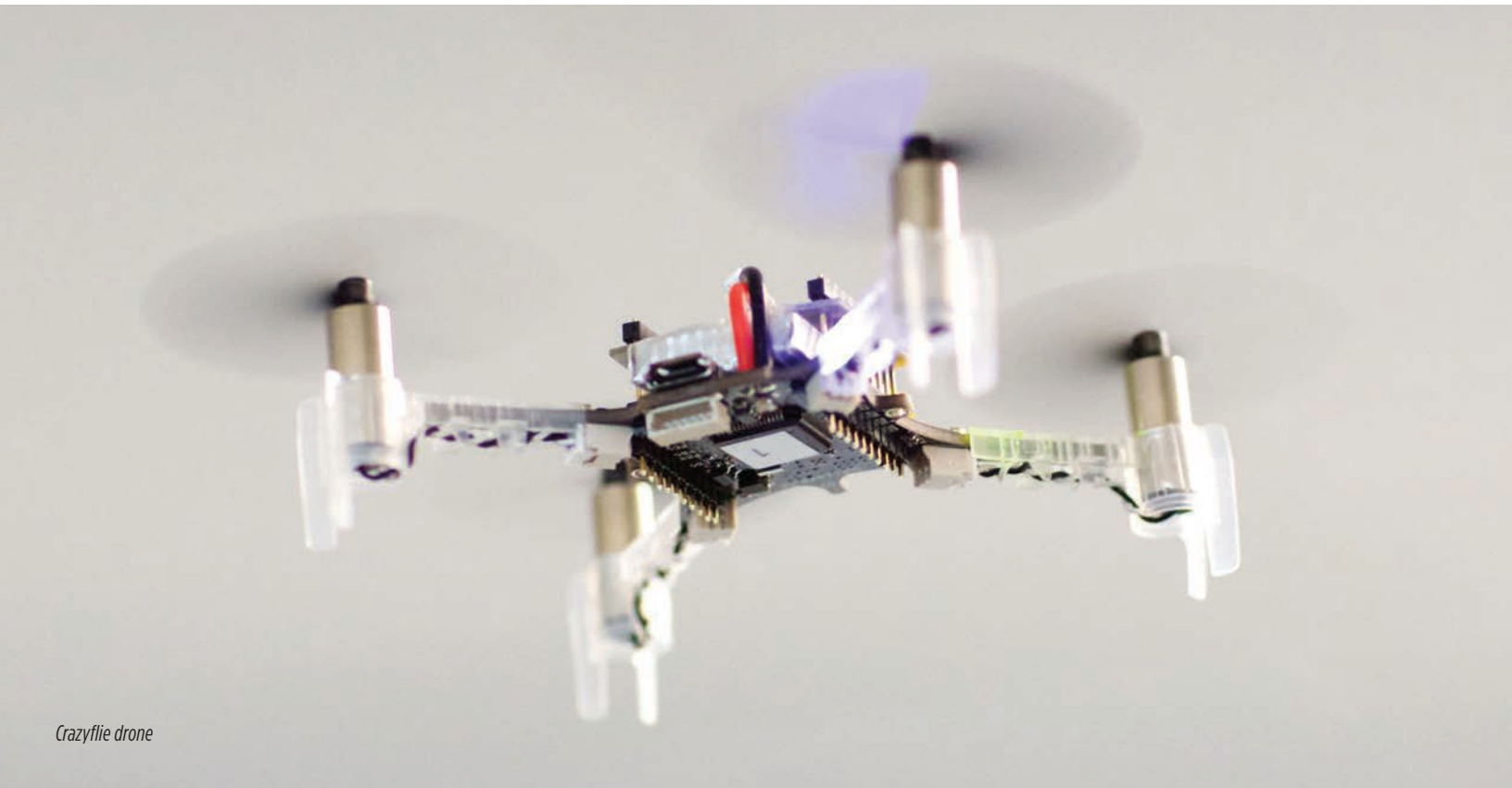
AdaCore intern Anthony Leonardo Gracio has rewritten the stabilization firmware for the Crazyflie drone in SPARK 2014, proving that it was free of run-time errors and also uncovering several bugs in the original C version.

The Crazyflie is a very small quadcopter sold as an open source development platform: both electronic schematics and source code are directly available from a GitHub repository, and its architecture is very flexible. However, the original software was not developed with safety in mind. Because the Crazyflie is small and lightweight with plastic propellers, it will not hurt anyone if it crashes. But if the propellers were made of carbon fiber, and tapered into sharp edges for better performance, then a software error in the flight control system could lead to serious injury or worse.

To gain higher confidence in the system's safety, Anthony re-implemented the stabilization software in SPARK 2014, making use of features such as constrained subtypes, state abstraction, and generics. One of the issues that arose was the prevention of overflow, since an out-of-range value used as input to a control command could damage a motor. The SPARK version of the software replaces out-of-range values by the highest or lowest permissible values, a style known as saturation. The saturation technique also helped the SPARK tools prove the desired properties of certain floating-point computations.

With drone safety becoming an increasingly important requirement, the Crazyflie project shows how the choice of a safe programming language can have major benefits.

For further information (including a link to the GitHub repository for the source code) please visit **blog.adacore.com/how-to-prevent-drone-crashes-using-spark/**. The SPARK tools are available from **libre.adacore.com/tools/spark-gpl-edition/**.



Crazyflie drone

Martyn Pike

Technical Account Manager

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

I graduated from Oxford Brookes University in England in the mid 1990's and then did graduate work in the area of high-integrity embedded software for military avionics; it was during my student days that I first encountered Ada. After graduate school I worked for the Embedded Software Division of Mentor Graphics, and then became an independent consultant where I specialized in high-integrity software development with a particular focus on Ada.

The subsequent years were dominated by work on military avionics and railway signaling systems, which I thoroughly enjoyed. Then in the Spring of 2014 I was approached by Quentin Ochem from AdaCore. Quentin had a vision of bringing to life an e-learning center to teach the Ada language and the use of related technologies: AdaCore University (see **u.adacore.com**).

I joined AdaCore in mid-2014 and started by performing an English language voice-over of the AdaCore University lectures. This was quite daunting to start with, as I'd never done anything like that before. I was completely out of my comfort zone. After a while I started to get the hang of it and eventually I authored some lectures and quiz questions myself.

This work made me feel an integral part of the AdaCore family—and I use that word deliberately, because the company after 20 years still has that start-up culture and overall fresh feeling. Even as it is growing and evolving in new directions I still see the core values and principles of the company being carefully preserved.

In my current role I am a Technical Account Manager based in the UK, which involves helping the sales team to deliver the right technical solutions to meet customer requirements.

You've had significant experience with embedded systems development in various languages including Ada, C, and C++. Any comments on how these languages compare, for example in ease of development, performance, etc.?

I first came across Ada while studying for my degree and oddly fell in love with strong domain typing almost instantly. Until this point I had been a C programmer, and at the time C++ was being touted as the next step in the evolution of C-based languages—however as soon as I saw domain typing, packages to aid program abstraction, and language level support for tasking, I was hooked on Ada.

It was during my time at Mentor Graphics that I really started to understand the integration between hardware and software and the unique challenges this presents. My eyes were opened to the key reasons why the C programming language was so popular in embedded systems development—the rawness of the language and its power over the hardware were almost on par with assembler languages. But clearly the deficiencies in the language were placing a large and entirely unnecessary responsibility on the programmer when it came to working at higher levels of abstraction and in producing reliable code. I felt this was being ignored by the industry, even as the technological earthquake brought on by the Internet meant that a local software problem could cause worldwide woe. It was frustrating to see all the attention being paid to new languages such as Java and later C# when I knew of a mature programming language called Ada with a strong pedigree that could provide all the abstraction and ease of use required by Internet-connected software. But things are changing.

The arrival of the 2012 revision of the Ada language has changed the perception of Ada within the software development community. I am extremely pleased that domains which previously had no knowledge of Ada's strong support for software engineering are learning about them thanks to features like contract-based programming, features designed to improve the economics of software development, especially during the verification phase of the software life cycle.

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

I am a huge Sci-Fi fan both as a keen reader and an avid watcher of films, both of which come in very handy when travelling for my work with AdaCore.

My children are still young so we are at the start of showing them the world through travel, which I greatly enjoy.

When I get the chance I like to take country walks and take in the beautiful sights of England's green and pleasant land.

Celebrating Ada Web Page Launched

December 10, 2015, marked the 200th anniversary of the birth of Lady Ada Lovelace, the world's first programmer. To help commemorate the occasion AdaCore served as a sponsor of a symposium in her honor at Oxford University (UK), and at that event announced a web page highlighting her achievements. The "Celebrating Ada" page shows timelines summarizing her life as well as key milestones for the language that bears her name.

For information about the Oxford symposium please visit **blogs.bodleian.ox.ac.uk/adalovelace/symposium/** and for AdaCore's web page celebrating Ada please visit **www.celebratingada.com/**.

VxWorks 7 in GNAT Pro 7.4

Support for Wind River's VxWorks 7 real-time operating system will be fully integrated into GNAT Pro 7.4. AdaCore engineers worked closely with Wind River on this product, ensuring that it would support both single- and multicore systems as well as other architectures. A completely reengineered open source debugger protocol and smoother integration with Wind River Workbench are some of the enhancements over earlier GNAT Pro versions for Wind River platforms, and the development environment handles both all-Ada and multi-language applications.

French DGA selects GNAT Pro and AdaSquore

The Direction Générale de l'Armement (DGA)—the French government agency that oversees military procurements—has selected AdaCore and SQUORING Technologies to provide a graphical tool, AdaSquore, for military and civil avionics software audits. Featuring a smart dashboard that monitors the quality metrics of critical components, AdaSquore will make it easier for developers and managers to analyze existing and new code in Ada and SPARK. The tool will also support the processes defined in the Formal Methods (DO-333) and Object-Oriented Technology (DO-332) supplements to the DO-178C avionics software standard.

New *Safe and Secure Software* booklet available

Safe and Secure Software: an Invitation to Ada 2012 has been updated to include a summary of SPARK 2014. Written by Ada luminary John Barnes, this 150-page booklet describes and illustrates the software engineering principles underlying Ada 2012 that are especially relevant for safety-critical and high-security systems. The booklet can be downloaded from **www.adacore.com/safesecure2012/**, or hard copies may be requested by sending email to **info@adacore.com**.

Online demos available

To help users gain a better understanding of AdaCore tools, several demos are available online. A demo of the GPS IDE introduces the main product features and shows how to use smart completion. The CodePeer demo consists of several sections on specific topics. One covers the tool's main features, another shows how to deal with race conditions and run-time checks, a third describes how the tool relates to the CWE (Common Weakness Enumeration) and floating point, a fourth illustrates how CodePeer is tested in the context of tool qualification, and a fifth summarizes the new features. To view these and other demos, please visit **www.adacore.com/knowledge/demos/**.

Here is a sampling of product enhancements that will be available in the next major release cycle, in Q1 2016.

GNAT Pro 7.4

GNAT Pro continues to build upon the strong foundation of gcc 4.9 while upgrading to the gdb 7.10 debugger technology. It will support Windows 10 as well as several new target platforms, in particular VxWorks 7 (ARM, e500v2, PPC, x86_64), VxWorks 653 3.0, and PikeOS (PowerPC). Among the more than 120 new features are the following enhancements:

- Generating C headers from Ada package specifications, which complements the existing facility (-fdump-ada-spec) for deriving Ada package specs from C header files
- Detecting invalid memory access via libsanitizer on Linux
- Enabling SSE floating point extensions by default on all x86 native ports
- Better performance for the Ada.Containers library, for example in the implementation of "for…of" loops and iterations
- New pragmas to support low-level programming on bareboard targets
- New pragmas to ease porting existing codebases from other Ada compiler environments

For further information about GNAT Pro please visit **www.adacore.com/gnatpro/**.

CodePeer 3.1

Codepeer 3.1 will include a "static debugger", supporting backtraces on precondition-related messages and displaying the possible values for any variable, and will provide persistent annotations with the resulting ability to perform faster and more precise incremental analysis for large projects. It will also correlate detected errors with the relevant weaknesses in the Common Weakness Enumeration (CWE), and allow integration with the Jenkins continuous integration server and the SonarQube open source platform for monitoring code quality. For further information about CodePeer please visit **www.adacore.com/codepeer/**.

SPARK Pro 16.0

SPARK Pro 16.0 will enhance both its SPARK language coverage and its GNATprove tool capabilities. This release will introduce support for the Ravenscar tasking profile and type predicates, generate counterexamples for unproved checks, and improve the handling of bitwise (modular) operations in proof. It will also include Alt-Ergo 0.99.1, CVC4, and C3 provers, and generate a global summary table. For further information about SPARK Pro please visit **www.adacore.com/sparkpro/**.

QGen 2.1

AdaCore's QGen tool, a qualifiable and customizable code generator from Simulink® and Stateflow® models, was launched in early 2015 and currently supports essentially all constructs used for modeling safety-critical control systems. QGen 2.1, which is compatible with Matlab 2015a/b, will add a number of enhancements. The new version will optimize code for switch blocks, allow adding external code for Lookup tables and Prelookup blocks, support commented-out and commented-through blocks, and support factoring of code for reference models and model libraries.

An important new capability—model-level debugging—is in the product pipeline in conjunction with a subsequent QGen release. Using the GPS IDE, developers will be able to debug both "pure" Simulink®/Stateflow® models and applications that combine manually prepared and autogenerated code. Host (native) debugging, and cross-debugging with any target supported by gdb, will be provided. For further information about QGen, please visit **www.adacore.com/qgen/**.

IDE tools

GPS 6.2 will include an improved code intelligence engine for C and C++ based on libclang, which will help with code completion, navigation, and block folding. GPS will also have a number of GUI enhancements (support for HiDPI monitors, TextMate color themes, and custom fonts), and performance improvements. Additionally, the user interface for the CodePeer plugin to GPS will relate error diagnostics to the Common Weakness Enumeration.

GNATbench 2.10 will support Eclipse Mars, 64-bit Windows, and Wind River Workbench 4 and VxWorks 653 (3.0). GNATdashboard 1.1 will support GNATcoverage and GNATstack, and it will be able to upload data to SonarQube 5. For further information on these IDE tools please visit **www.adacore.com/gnatpro/toolsuite/gps/**, **www.adacore.com/gnatpro/toolsuite/gnatbench/**, and **www.adacore.com/labs/gnatdashboard/**.

Automating Test Case Generation in Ada 2012 with GNATtest

Testing is an essential part of the software life cycle, but setting up the test harnesses and constructing appropriate test cases can be tedious and error prone. The GNATtest tool, in conjunction with the GNAT-defined Ada 2012 aspect Test_Case (also available as a pragma) expedites this process by allowing the developer to annotate subprogram specifications with testing-related information. GNATtest thereby makes it easier to generate test cases and to check whether they run successfully.

GNATtest is a source code analyzer. It creates a test driver infrastructure and (by default) generates one test case skeleton per analyzed subprogram. The test case author will need to flesh out the test case by selecting appropriate subprogram input data and determining the expected outputs through white-box code analysis. To simplify this effort the subprogram author can supply information about the test cases that will exercise the subprogram, including whether to use expected (nominal) or out-of-range (robustness) input values. This information is conveyed through the Test_Case aspect or pragma, which can be associated with a subprogram in a package specification. Here is an example using the aspect syntax:

```
package Math_Lib is
   ...
   function Sqrt (Arg : Float) return Float with
     Test_Case =>
      ( Name     => "Test 1",
        Mode     => Nominal,
        Requires => Arg = 49.0,
        Ensures  => Sqrt'Result = 7.0 ), -- Return exact result
     Test_Case =>
      ( Name     => "Test 2",
        Mode     => Robustness,          -- Reject negative argument
        Requires => Arg < 0.0 );
   ...
end Math_Lib;
```

The Name string appears in GNATtest messages and in test skeleton comments.

When GNATtest encounters a Test_Case aspect it generates an intermediate subprogram within the test case skeleton; this intermediate subprogram acts as a wrapper to the actual subprogram under test. For simplicity the intermediate subprogram is given the same name as the actual subprogram, and an Ada "renames" is used to make it appear that the subprogram under test is being invoked directly.

Based on the "Requires" Boolean expression from the Test_Case aspect, the intermediate subprogram checks the input values given by the user-written part of the test case, and then on return from the tested subprogram checks the "Ensures" Boolean expression. If either Boolean expression evaluates to False, then the test will automatically fail, thus prompting further investigation. The Requires and Ensures expressions are independent of any pre- and postconditions that may be supplied for the subprogram.

The Test_Case aspect extends the concept of programming by contract to entail dynamic verification by contract, reducing the amount of code to be hand written and making the test infrastructure more maintainable. Through this aspect a subprogram author can also specify which subprogram inputs exercise "Nominal" paths of execution and which exercise "Robustness" paths. The information in the subprogram specification thus permits a hybrid black-box / white-box approach to dynamic verification that is consistent with the objectives in software standards such as DO-178B/C.

For further information please visit **www.adacore.com/gnatpro/toolsuite/gnattest/**.

Check Out the AdaCore Blog

The company's blog site offers up-to-date news, "how-to" stories, and opinions from AdaCore experts, and allows readers and authors to interact through the comment sections. Here is a sampling from the posts that have appeared since Summer 2015:

- In **"Make with Ada: from Bits to Music"** Raphaël Amiard presents a sound synthesis library written in Ada that can be used on small bareboard computers such as the Raspberry Pi 2.
- Emmanuel Briot's article **"Traits-Based Containers"** proposes a new generics-based API for containers (bounded and unbounded lists, etc.). It takes a different approach than the standard Ada library in meeting the goals of provability, efficiency, safety, and reusability.
- The article **"Using Reference Types to Handle Persistent Objects"**, by Maxim Reznik and Nicolas Setton, illustrates how to use the Ada 2012 user-defined reference facility to simplify the programming of applications (such as database interfacing) that deal with persistent objects.
- Fabien Chouteau's **"Make with Ada: Formal Proof on my Wrist"** describes how he generated an Ada binding to the Pebble OS for the Pebble Time smartwatch and ported the Tetris game app (renamed to Patris) to run on the watch. The app is written in SPARK, with a formal proof of program properties including freedom from run-time errors and assurance of valid game states.
- Another article by Emmanuel Briot, **"Calling Inherited Subprograms in Ada"**, presents an Ada Object-Oriented Programming idiom that can be used when an overriding subprogram needs to invoke the parent type's version of the subprogram.

For these and other entries please visit: **blog.adacore.com**

professional training

Learn Ada from AdaCore Expert Instructors

Public courses in UK and US during April 2016

A 5-day course on Ada fundamentals will be held during April 11–15, 2016, in Yeovil, UK, and during that same week at AdaCore's New York headquarters. Consisting of lectures and hands-on workshops using the latest AdaCore tools, the course covers language basics as well as topics relevant to embedded systems applications, and it explains contract-based programming as well as other important Ada 2012 features. The course is open to the public and does not require previous Ada experience. For registration information and a detailed schedule please visit **www.adacore.com/public-ada-training/**.

Tech Days in Paris and Boston

During Autumn 2015 AdaCore hosted two customer-centric events, one in Paris and the other in the Boston area. These conference-style meetings featured presentations and demos by AdaCore staff, offered project success stories from GNAT Pro users, and most importantly allowed customers and AdaCore personnel to meet and talk with one another directly and in an informal social setting.

The AdaCore Tech Day conference in Paris, known as "GNAT Industrial User Day" in past years, was held on October 1. Topics included an AdaCore product roadmap, dynamic software verification with GNAT, static analysis with CodePeer, the QGen model-based development tool, SPARK 2014, a summary of the company's embedded products, and multi-language programming with GPRbuild. Live demonstrations of a Crazyflie 2.0 drone and a railway simulation, both of which used software written in SPARK and Ada 2012, added to the day's highlights. And customer presentations from the financial software domain, which is not an area where Ada has traditionally seen heavy use, sparked several discussions on the role of Ada on mission-critical systems.

For up-to-date information on dates and locations for AdaCore Tech Days in 2016, please visit www.adacore.com/techdays/.

The AdaCore Tech Days conference in the US was held in metropolitan Boston (Burlington, Mass.) during November 3 and 4. With the two-day duration the event expanded on the Paris agenda with demos and tutorials, treating subjects including the GPS and GNATbench IDEs, Ada 2012 language features, and ARM programming with Ada. A user presentation from a GNAT Academic Program (GAP) member described a space project—CubeSat software and hardware—which served as an Ada/SPARK success story and showed that undergraduate students could easily learn and become proficient with the SPARK language.

Both events were well attended and well received, and plans are moving forward to organize similar AdaCore Tech Days conferences in Europe and the US in 2016.

To view the slides that were presented at the 2015 events, please visit **www.slideshare.net/adacore/**.

calendar highlights / November 2015–February 2016

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

ISSRE 2015: International Symposium on Software Reliability Engineering
November 2–5, 2015 / Gaithersburg MD, USA

Quentin Ochem and Eric Perlade presented a talk: "Formal Methods for Informal Developers - A case-study driven by the French defense agency (DGA)."
issre.net

AdaCore Tech Days 2015, Boston
November 3–4, 2015 / Boston MA, USA

Please see a summary of this event in a companion article on this page.
www.adacore.com/techdays/us/

High Integrity Software 2015
November 5, 2015 / Bristol, UK

AdaCore was a major sponsor of this conference.
his-2015.co.uk

ARM TechCon
November 10–12, 2015 / Santa Clara CA, USA

AdaCore was an exhibitor at this event, and Pat Rogers presented a tutorial: "The Practical Development of Safe, Secure, and Reliable Embedded Software."
www.armtechcon.com

NBAA2015 – Business Aviation Convention & Exhibition
November 17–19 / Las Vegas NV, USA

AdaCore was an exhibitor at this show.
www.nbaa.org/events/bace/2015/

Ada Lovelace Symposium
December 9–10, 2015 / Oxford, UK

AdaCore was a sponsor of this event.
blogs.bodleian.ox.ac.uk/adalovelace/symposium/

HASE 2016: 17th IEEE International Symposium on High Assurance Systems Engineering
January 7–9, 2016 / Orlando FL, USA

Tucker Taft presented a paper "High-Integrity Multitasking in SPARK: Static Detection of Data Races and Locking Cycles", co-authored by Florian Schanda (Altran) and Yannick Moy.
hase2016.org

ERTS² 2016: Embedded Real Time Software and Systems
January 27–29, 2016 / Toulouse, France

AdaCore is a Gold Sponsor and exhibitor for this conference, and Thomas Quinot is presenting a paper: "Structural Coverage Criteria for Executable Assertions".
www.erts2016.org/about.html

Embedded World 2016
February 23–25, 2016 / Nürnberg, Germany

AdaCore is exhibiting at this event in Booth 4-149, Hall 4.
www.embedded-world.de/en/

Inside AdaCore 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