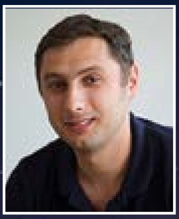
AdaCore

Space Sector

Interview: Eric Perlade, Technical Account Manager, AdaCore.



Are there any specific requirements for space applications?

Space applications are either safety critical or mission critical, and most have hard real-time constraints. Guiding a launcher or delivering positioning data requires precision and high reliability. The software has to be maintained for 10 years for satellites and 20 or more years for launchers with different versions. What's more, the environment in which onboard software is running does not allow the usual patching methods. Patching is either impossible or else only achievable with limited capabilities.

Why is the Ada language suitable for space applications?

All those ingredients make Ada not only suitable for space applications but a very good fit, and it is widely used to develop onboard software. What we see in the space industry is the use of Ada for bare metal applications, that is, where the target platform does not have an underlying operating system. Ada also offers the flexibility and portability to run on native platforms and on embedded real-time operating systems, with advanced tasking capabilities.

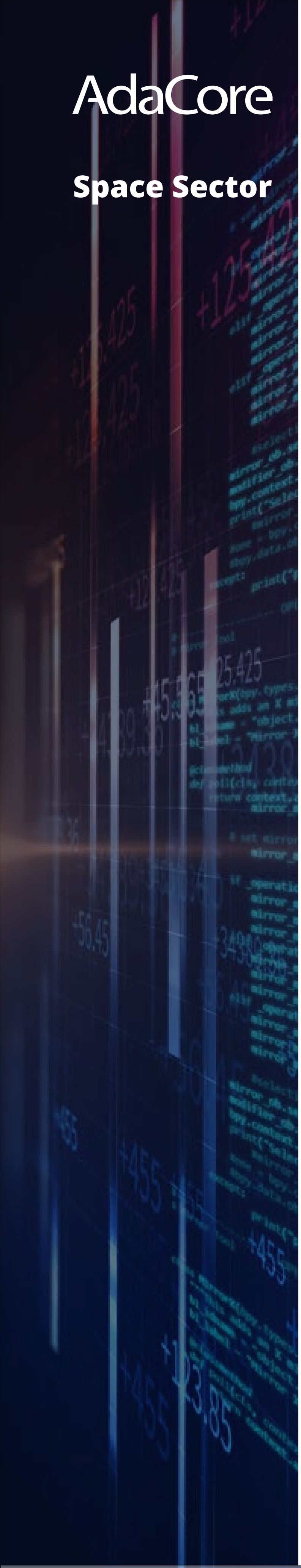
Because Ada is a general-purpose imperative programming language, you can use it to write software similarly to how you would do it in C or C++. You can write sequential programs (with a single thread of control), or, in a more advanced way, write concurrent programs with features that are high-level but efficient. You can develop software in Ada using object-oriented programming, or even contract-based programming and formal methods.

What are Ada's main attributes?

The language has been designed with reliability and maintainability in mind. It's intended for engineers who aren't necessarily specialists in programming language technology, but rather experts in their industry domain such as flight control systems for launchers. It provides both a high level of abstraction and an access to low-level hardware interfaces when required.

Ada offers a strong typing mechanism that allows users to detect problems at compile time. The language also defines a list of run-time checks that are automatically added by the compiler. Any run-time check violation will raise an exception instead of smashing the stack with a buffer overflow or causing an invalid pointer reference; the benefit of these checks is less time debugging, and more time to focus on other life cycle activities.

Using Ada in your workflow allows you to detect bugs as early as possible, first while compiling and then during testing. This helps to increase product reliability while reducing the development and verification costs.



What advantages does Ada have over other languages?

Ada is readable, with clear syntax that conveys the program intent and avoids traps and pitfalls, and with well-defined semantics for run-time behavior that other languages leave unspecified. Ada supports modular software architectures, with its package facility separating specification from implementation. For low-level programming where some of Ada's checks have to be disabled, the language features have an explicit syntax (an "Unchecked" prefix) that makes their usage apparent to the human reader. All of this makes Ada a very good choice when projects have to be maintained for many years.

Ada also provides a flexible and tailorable run-time environment, with different subsets that can be selected depending on a project's needs. In the space industry, typical options are the Zero FootPrint (ZFP) run-time and the Ravenscar Small FootPrint (SFP) run-time. Both can be run on bare metal, providing a safe subset of Ada. The Ravenscar SFP run-time supports deterministic scheduling for hard real-time applications with one or multiple tasks. Ada is a safe but expressive language for critical onboard software for space hardware targets like ERC32, LEON and ARM.

What types of tools does AdaCore offer for developers of software for space vehicles and satellites?

AdaCore provides a full complement of software development and verification tools. Our core GNAT Pro Ada development environment includes cross compilers hosted on Linux or Windows for all the usual hardware targets used in the space industry. GNAT Pro compilers undergo rigorous Quality Assurance, with nightly regression tests encompassing the official ACATS (Ada Conformance Assessment Test Suite) as well as extensive internal test suites.

In addition to the compiler, GNAT Pro Ada supplies a variety of utilities. These include the GNATcheck coding standard compliance checker, the GNATmetric program metrics tool, the GNATtest test harness generator, and the GNATstack stack usage computation tool. The GNATcoverage tool, available as a supplement, performs coverage analysis at levels up to MC/DC. Ada projects typically involve software not just in Ada but also in C, and AdaCore offers GNAT Pro C Development Environments to help provide "one-stop shopping" for customers with requirements for multiple language support.

For advanced static analysis detecting run-time/logic errors, we offer CodePeer, a comprehensive static analysis tool-suite for Ada. This is useful for detecting latent bugs and vulnerabilities in existing code bases and can also be used effectively during initial development. Another AdaCore product is SPARK Pro, a toolset that brings mathematics-based confidence to software verification, based on the formally analysable SPARK subset of Ada. And we also offer QGen, a model-based engineering tool for a safe subset of Simulink® and Stateflow® models, which generates code in SPARK or MISRA C. All of AdaCore's tools are fully integrated into the graphical development environment, GNAT Studio.

What about services?

AdaCore can supply a customized run-time with the delivery of an associated qualification package following the ECSS-E-ST-40C and ECSS-Q-ST-80C standard. We also provide all the tools required to build a qualified space application. All the phases are covered: from writing formal requirements, to the implementation, followed by ISVV (Independent Software Verification & Validation) activities - requiring code review and testing tools - to the final qualification phase.

More generally, all AdaCore product subscriptions come with expert support services supplied by the product developers themselves, with timely and accurate responses to any question about the technology.

Can you tell us more about the GNAT Pro Assurance for critical systems?

GNAT Pro Assurance is a complete Ada solution for projects with the most stringent requirements for reliability, long-term maintenance or certification.

The product supports all versions of the Ada language standard (from Ada 83 to Ada 2012), with C support as an optional add-on. It includes a full tool suite as well as a configurable run-time library and several specific run-times that are especially suited to high-assurance systems.

Unique to the GNAT Pro Assurance product is specialized service known as sustained branches, which allows a project to continue its use of a specific version of the technology, including upgrades to repair critical issues. This is the key feature of the product, since any project in the space industry has to freeze the version of the tools to maintain the qualification of the final application.

Is it cost effective?

Migrating to a newer version of the product is costly, as all testing and quality assurance activities have to be repeated. However, GNAT Pro Assurance allow users to reduce this cost while maintaining the same level of support through a yearly subscription.

AdaCore has a long history of working with customers who need to meet certification requirements, and GNAT Pro Assurance offers a number of options that can simplify the development and verification effort. These include certification material for the run-time libraries, as well as qualification material for several of the tools.

Can you help customers meet the required standards?

AdaCore has extensive experience helping our aerospace customers meet the European Space standards ECSS-E-ST-40C and ECSS-Q-ST-80C and has already qualified several versions of the run-time libraries.

We have worked on many projects with many different customers over the last decade (see the list below for further information).

Customers and projects:

ESA (2019)

Selected AdaCore to provide a qualified multitasking solution for spacecraft software development to support multiple ongoing and future ESA contracts.

AVIO (2019)

Selected GNAT Pro Assurance Ada Development Environment to implement the on-board software for its Vega-C launch vehicle which handles guidance, navigation and control. European Space Agency sponsored.

MDA (2017)

Selected GNAT Pro Assurance Ada Development Environment for its LEON3 target processor to produce software for a new International Space Station communication subsystem.

Astrium (2013)

Selected GNAT Pro and PolyOrb toolset for the Core Ground System on the International Space Station. Operates reliable communication between the Columbus laboratory and onboard applications.

Cubesat (2013 press release, 2019 blog)

Vermont Technical College in the U.S. selected the GNAT Pro Ada Development Environment and adopted the SPARK language for its NASA-sponsored satellite which launched into orbit in 2013 to test systems used for lunar missions. After ten years the CubeSat Laboratory at Vermont Technical College continues to use the SPARK technology for its rigor and sound formal verification framework.

Thales (2011)

Selected GNAT Pro Assurance to develop onboard instrument software for the Argos Satellite project - a worldwide tracking and environmental monitoring system.