SPARK 2014:何が新しくなったか

SPARK 2014とは、SPARK 2005からの数多くの重要な改善が施された新たな版本です。その中からいくつかの新機能を挙げてみましょう。

1. **SPARK 2014**:
   - 新機能を追加: Increment に新しいパラメーターを追加。
   - 前処理: Global, Control State, Increment。
   - ポスト処理: Control State, Increment。
   - モード: Accelerating。

2. **SPARK 2005**:  
   - 新機能を追加: Increment に新しいパラメーターを追加。
   - 前処理: Global, Control State, Increment。
   - ポスト処理: Control State, Increment。
   - モード: Accelerating。

SPARK 2014がSPARK 2005とどのように異なっているかを詳しく説明します。新機能と改善点について、詳しく説明をします。
SPARK Pro 11.1
This new release incorporates a number of enhancements to the SPARK Pro toolset as well as some minor bug fixes, and also brings the technology to several new platforms. To ease the transition from SPARK 2005 to SPARK 2014, the Examiner can now optionally ignore SPARK 2014 aspects. Various improvements to the proof system mean that users should see more verification conditions being discharged automatically. The toolset is now available for 64-bit Windows, and Victor is now available on Solaris.

SPARK Pro 14.0 Beta
A beta release of the SPARK 2014 toolset is now available to SPARK Pro customers. This is a major upgrade to the SPARK Pro toolset, with a ground-up rewrite based on the GNAT Pro front end and an entirely new flow analysis engine. Verification Condition generation and proof are now handled by GNATprove (using Why3 technology) with proof results presented at the source code level in the GDS IDE. For more details of the SPARK 2014 language please see the Technology Corner article on Page 4 of this newsletter.

GNAT Pro for Wind River Linux
GNAT Pro is now available on the Wind River Linux platform. GNAT Pro can be used in conjunction with Wind River products to compose multi-language applications combining Ada, C, and C++, and Ada code can be manipulated and analyzed through Wind River’s Linux browser tools. GNAT Pro for Wind River Linux supports all versions of the Ada language standard from Ada 83 through Ada 2012, and is also supported on the Wind River Workbench development environment.

Interview with Ben Brolsgod, Senior Technical Staff, AdaCore US
Tell us about your background and how you came to be involved with Ada and AdaCore. What is your current role?
I was a grad student in Applied Math and Computer Science at Harvard back in the late 1960s and early 1970s, and that’s where I became intrigued by programming languages and compilers. This was a golden age for research in software methodology, and new ideas like structured programming, encapsulation (information hiding), object orientation, and concurrency primitives (semaphores, monitors) were revolutionizing the programming world. After receiving my Ph.D. for thesis work in which my subject was Deterministic Translation Grammars—I joined Intelerratics in 1973 and soon got involved with the project that would lead to Ada. I was in charge of the “Real” language in the design competition, but we ultimately lost out to the “Green” language from Jean Kilby at CII-Hewlett-Packard in France. Ada has pretty much been my career since then. In early 1983 I joined Jean Kilby’s company, Alsys, and helped found its US branch. I did some independent consulting in the early 1990s, and a good chunk of that work was on the Ada 95 project under the direction of Tucker Taft. After another stint with Alsys (then Arrows), I joined AdaCore in March 2010. My work has been a good mix of technical and marketing writing, presenting papers and tutorials at conferences, conducting professional Ada courses, preparing press releases, and so on. On occasion I’ve managed development projects, including a current project to produce qualification evidence for GNATcoverage as a verification tool under DO-178B. Oh yes, I also edit the GNAT Pro Insider newsletter, which makes this interview an interesting exercise.
I’ve been involved with the ACM SIGAda organization since its inception, including a long stretch as Vice Chair for Meetings and Conferences and two terms as Chair. I guess I can take the credit (or blame) for launching the TRI-Ada series of conferences/trade-shows back in the late 1980s, which helped put Ada on the map.
Occasionally I’ve branched out into other areas. I was a member of the Expert Group that produced the Real-Time Specification for Java, and that was both fun and challenging, although it hasn’t had much impact in the embedded systems marketplace. Hard real-time was not what Java was originally designed for, so we had to extend Java’s thread scheduling and memory management models to get the necessary predictability. I’ve also been following the DO-178C effort and have created several tutorials and papers on that topic.

You’ve been working in the computer software industry for a long time. Any insights or predictions?
It’s depressing that the term “software crisis” has been kicking around for forty-five years but seems just as timely today as when it first appeared. So this is a chronic condition, not a crisis. Unfortunately, some of the solutions that have appeared over the years—such as fancy tools that make it easy to write and run programs—may be part of the problem, since they can encourage a “plug something together and debug it” approach to software development. But debates such as Knight Capital Group’s $440 million “glitch” last year tend to get people’s attention, so there’s hope for progress. Tools and methods that focus on the front end of the development cycle (requirements capture/analysis) or on the interface between the software and the overall system are especially valuable, since that’s where the subtle issues tend to lurk. Languages are important of course; they can help in expressing assumptions as part of the source code and detecting them early. We’ve seen this with SPARK and, more recently, Ada 2012 with contract-based programming. And combining formal methods with traditional testing can be very effective.
Software does not have to be buggy. Industries with a rigorous certification process and a long-standing culture that focuses on safety and reliability, such as commercial aviation, have shown that the problems are solvable.

Any hobbies or outside interests that you’d like to share?
I’m a bit of a movie buff, with a particular fondness for some of the early films of Orson Welles. The Third Man, one of my favorite films, with brilliant performances from Laurence Olivier, Joseph Cotten, and Judi Dench. It won the Best Picture award in 1941, but the director, Welles did not win as Best Director. Another gem from later in that decade was Carol Reed’s The Third Man, with its memorable zither music and Orson Welles at his cynical best. Without CGI special effects to attract an audience in those days, movies had to focus on character and story. The good ones did this well.

GNAT Industrial User Day
AdaCore hosted the 2013 GNAT Industrial User Day in Paris on September 25, with customers, partners, and GNAT Academic Program members attending. With the theme of “Meet, Learn, Share”, the event featured presentations from company staff, partners, and very importantly, AdaCore customers. The company’s presentations covered a range of topics including the role of formal verification, the new AdaCore University initiative, Ada on ARM® for automotive and safety-critical embedded systems. The AdaCore team received much excitement around GPS 6.0, GNATdashboard, GNAT Tracker 3.0, and the more general roadmap. 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 2014.

AdaCore University Launched
To make it easier for developers to quickly come up to speed in Ada, a free, web-based educational resource center known as AdaCore University has been launched by AdaCore. The site has flexible controls for navigating through the lessons, including pausing, rewinding, fast forwarding, etc. The courses educate through examples, allowing students to see, understand and experiment with most features of the Ada programming language, and comparisons with C, C++, and Java are used to illustrate particular points of semantics. Each course includes one or more interactive tools for writing and running example programs. Students at all levels of experience and expertise can begin writing programs quickly and can be manipulated and analyzed through Wind River’s Wind River Workbench development environment.

AdaCore University Launched
To make it easier for developers to quickly come up to speed in Ada, a free, web-based educational resource center known as AdaCore University has been launched by AdaCore. The site has flexible controls for navigating through the lessons, including pausing, rewinding, fast forwarding, etc. The courses educate through examples, allowing students to see, understand and experiment with most features of the Ada programming language, and comparisons with C, C++, and Java are used to illustrate particular points of semantics. Each course includes one or more interactive tools for writing and running example programs. Students at all levels of experience and expertise can begin writing programs quickly and can be manipulated and analyzed through Wind River’s Wind River Workbench development environment.

You’ve been working in the computer software industry for a long time. Any insights or predictions?
It’s depressing that the term “software crisis” has been kicking around for forty-five years but seems just as timely today as when it first appeared. So this is a chronic condition, not a crisis. Unfortunately, some of the solutions that have appeared over the years—such as fancy tools that make it easy to write and run programs—may be part of the problem, since they can encourage a “plug something together and debug it” approach to software development. But debates such as Knight Capital Group’s $440 million “glitch” last year tend to get people’s attention, so there’s hope for progress. Tools and methods that focus on the front end of the development cycle (requirements capture/analysis) or on the interface between the software and the overall system are especially valuable, since that’s where the subtle issues tend to lurk. Languages are important of course; they can help in expressing assumptions as part of the source code and detecting them early. We’ve seen this with SPARK and, more recently, Ada 2012 with contract-based programming. And combining formal methods with traditional testing can be very effective.
Software does not have to be buggy. Industries with a rigorous certification process and a long-standing culture that focuses on safety and reliability, such as commercial aviation, have shown that the problems are solvable.

Any hobbies or outside interests that you’d like to share?
I’m a bit of a movie buff, with a particular fondness for some of the early films of Orson Welles. The Third Man, one of my favorite films, with brilliant performances from Laurence Olivier, Joseph Cotten, and Judi Dench. It won the Best Picture award in 1941, but the director, Welles did not win as Best Director. Another gem from later in that decade was Carol Reed’s The Third Man, with its memorable zither music and Orson Welles at his cynical best. Without CGI special effects to attract an audience in those days, movies had to focus on character and story. The good ones did this well.

GNAT Industrial User Day
AdaCore hosted the 2013 GNAT Industrial User Day in Paris on September 25, with customers, partners, and GNAT Academic Program members attending. With the theme of “Meet, Learn, Share”, the event featured presentations from company staff, partners, and very importantly, AdaCore customers. The company’s presentations covered a range of topics including the role of formal verification, the new AdaCore University initiative, Ada on ARM® for automotive and safety-critical embedded systems. The AdaCore team received much excitement around GPS 6.0, GNATdashboard, GNAT Tracker 3.0, and the more general roadmap. 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 2014.
AdaCore University Launched

To make it easier for developers to quickly come up to speed in Ada, a free, web-based educational resource center known as AdaCore University has been launched by AdaCore. The site contains pre-recorded courses and other learning materials and provides access to AdaCore’s GNAT Ada tools for writing and running example programs. Students at all levels of experience and expertise can begin writing programs quickly and can proceed at their own pace. The site has flexible controls for navigating through the lessons, including pausing, rewinding, fast-forwarding, etc.

The courses educate through examples, allowing students to see, understand and experiment with most features of the Ada programming language, and comparisons with C, C++, and Java are used to illustrate particular points of semantics. Each course includes one or more interactive quizzes, with detailed explanations of the solutions. Drawing on the experience and teaching credentials of Ada experts such as AdaCore founders and New York University Emeritus Professors Robert Dewar and Edmond Schonberg, the courses explain Ada’s technical concepts with insight into the rationale and usage of particular features.

The initial curriculum includes two courses, presented by AdaCore University project leader Quentin Ochern:

- Ada D01 (Overview) – a module that presents an overall picture of the language and shows how to build and run programs
- Ada D02 (Basic Concepts) – a module that comprises a series of lessons on Basic Types, Statements, Arrays, Records, Subprograms, and Packages

These courses are complemented by a set of lab exercises to reinforce the concepts covered by the lectures. The courses cover the latest version of Ada, which introduces new features such as concurrent-based programming (preconditions, postconditions, invariants). The AdaCore University website also hosts a number of technical papers on Ada, offering insight into particular aspects of the language’s design and usage.

AdaCore University is an ongoing, live project that will be expanded to include more advanced courses on Ada, as well as training materials for SPARK 2014—a new version of the Ada-based SPARK programming language for high-integrity software. For more information, please visit ada.adacore.com.

Interview with Ben Brosol
Senior Technical Staff, AdaCore US

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

I was a grad student in Applied Math and Computer Science at Harvard back in the late 1960s and early 1970s, and that’s where I became intrigued by programming languages and compilers. This was a golden age for research in software methodology, and new ideas like structured programming, encapsulation (information hiding), object orientation, and concurrency primitives (semaphores, monitors) were revolutionizing the programming world. After receiving my Ph.D. for thesis work on Ada, I put Ada on the map.

I’ve been involved with the ACM SIGAda organization since its inception, including a long stretch as Vice Chair for Meetings and Conferences and two terms as Chair. I guess I can take the credit or blame for launching the TRI-Ada series of conferences/trade-shows back in the late 1980s, which helped put Ada on the map.

My current role is a GNAT Pro insider.

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

I’m a bit of a movie buff, with a particular fondness for some of the early classics like Citizen Kane, The Maltese Falcon, and Brief Encounter. I also enjoy playing chess and am an avid reader, particularly of science fiction. I’ve been a long-time fan of the Hitchcock films and particularly like his noir thrillers like Shadow of a Doubt and Strangers on a Train, which features a plot device of a hero pursued by both the bad guys and the law. And Rebecca is one of my favorite films, with brilliant performances from Laurence Olivier, Joan Fontaine, and Judith Anderson. It won the Best Picture award in 1940, but my vote did not win as Best Director. Another gem from later in that decade was Carol Reed’s The Third Man, with its memorable zither music and Orson Welles at his cynical best. Without CGI special effects to attract an audience in those days, movies had to focus on character and story. The good ones did this well.

GNAT Industrial User Day

AdaCore hosted the 2013 GNAT Industrial User Day in Paris on September 25, with customers, partners, and GNAT Academic Program members attending. With the theme of “Meet, Learn, Share”, the event featured presentations from company staff, partners, and very importantly, AdaCore customers. The company’s presentations covered a range of topics including the role of formal verification, the new AdaCore University initiative, Ada on ARM 2014, GNATprove, GNATdashboard, and GNAT 2014. The event concluded with a raffle focused on GNAT graphics and software development tools.
What’s New in SPARK 2014?

The SPARK language has a long history of successful usage in safety-critical and high-security software, but it supports statically proving properties such as freedom from run-time errors or completeness with a formal specification of a program’s requirements. And because SPARK is a subset of Ada, SPARK programs can be compiled by any standard Ada compiler. A major new version of the language, known as SPARK 2014, preserves these benefits while introducing several significant innovations.

Consistency with Ada 2012 and Hybrid Verification

Since Ada 2012 provides contract based programming features (e.g., pre- and post-conditions, and quantified and conditional expressions) and a uniform framework for aspects, SPARK’s “#” annotations as-structured-comment syntax has been replaced with constructs that use valid Ada semantics. This is not just a cosmetic change. The new syntax is at the heart of SPARK 2014’s hybrid verification, a novel approach to safely combining formally verified and traditionally verified code in one program based on on-time checking of contracts at their boundary. For example, if a formally verified subprogram is called by code that is not subject to formal verification, the precondition of the called subprogram can be checked when the call is executed. If the precondition fails, then the called subprogram’s proofs are inapplicable. Using hybrid verification, developers can selectively apply formal verification to only those portions of a program where it is feasible and cost-effective. Executable checking contracts also allow contracts to be derived more easily and more quickly; this is valuable for both formally and traditionally verified code.

Larger Language Subset

SPARK 2014 supports a richer subset of Ada (now Ada 2012) than did earlier versions of SPARK. The fundamental restrictions that the proof technology relies upon are still enforced (no side effects in functions, no exception handlers, no tasks, no controlled types), but many other restrictions are now relaxed. SPARK 2014 allows discriminated types, dynamic constants, dynamic subtypes, recursion, and generic packages and subprograms. A predefined SPARK container library (generic packages implementing maps, sets, vectors, and lists) provides much of the essential functionality of access types for managing flexible data structures, but without the verification problems that arise with access types having associated access types.

Easier Ada-to-SPARK Transition

SPARK 2014 makes it easier to transition from non-SPARK Ada to SPARK. Part of a subprogram’s contract is the set of global variables that it depends upon (directly or as callbacks to other subprogram inputs and outputs). SPARK 2014 offers an alternative where data dependencies for a given subprogram are determined automatically by analyzing the body. Non-SPARK APIs for SPARK interfaces (e.g., an SPARK Ada body for a non-SPARK Ada subprogram declarator) are also supported.

For more information, please see www.spark-2014.org/about.

Conferences / Events  October 2013–April 2014

23rd Digital Avionics Systems Conference (DASC) October 6–10, 2013 / Syracuse NY, USA Tucker Taft is giving a talk on integrating testing and proof with SPARK and dasconline.com

IET International System Safety Conference October 15–17, 2013 / London, England AdaCore is a sponsor and exhibitor conferences.iet.org/systemsafety/

ACM SIGPLAN’s SPLASH 2013 Systems, Programming, Languages and Applications: Software for Humanity October 26–31, 2013 / Indianapolis USA Tucker Taft is presenting a tutorial and a "tech talk" on contracts in multi-core complex systems splash2013.acm.org/program/tutorials-tech-talks

RTBCE Seattle Real-Time & Embedded Computing Conference November 5–7, 2013 / Seattle WA, USA AdaCore is exhibiting at this event. rtec.com/events/details/tid1211city=Seattle

ACM SIGAda’s HILT 2013 High Integrity Language Technology November 10–14, 2013 / Pittsburgh PA, USA Tucker Taft is presenting a tutorial on proving safety of concurrent programs. AdaCore is a Platinum sponsor and exhibitor. sigAda.org/conf/hilt2013/

Safety-critical Systems Symposium 2014 February 4–6, 2014 / Brighton, UK AdaCore is a major sponsor of this event. www.gnat-sys.org/safety/

ERTS’ 2014 Embedded Real Time Software and Systems February 5–7, 2014 / Toulouse, France AdaCore is a Gold Sponsor of this event. www.erts.org

Embedded World 2014 February 24–27, 2014 / Nuremberg, Germany AdaCore is exhibiting at this event. www.embedded-world.de/

The GNAT Pro Insider is published twice a year straddling New York in New York and Paris by AdaCore.

The GNAT Pro Insider is published twice a year straddling New York in New York and Paris by AdaCore.

Design West March 31–April 3, 2014 / San Jose CA, USA AdaCore is exhibiting at this event - Booth 7106. www.udemidesign.com/sanjose

GPS 6.0 Released

New Major Revision Upgrades IDE’s Look and Feel

For this new GNAT Programming Studio (GPS) release, the AdaCore development team has conducted a complete review of the existing interface and made it cleaner and easier to use. Based on customer feedback, the improvements include a major upgrade for editing and a number of design changes to bring program-related information within easy reach. This is supported by a new relational database at the heart of the GPS engine to make searches much more efficient. The guidelines underlying the GPS 6.0 revision help the IDE achieve its main goal: to serve as a customizable platform for multi-language multi-tool integration, usable by developers at all experience levels.

The improvements to the IDE’s look and feel take advantage of the latest Gtk+/GtkAda graphical toolkit and fall into several categories:

• Interface reorganization
• More economic usage of screen real estate, for example by placing tabs on top of the view
• Improved toolbar with monochromatic icons
• Ability to position tabs on the side, for better appearance on wide screens
• Better handling of documents, for example with editors and views clearly separated
• Actions
• Ability to search in all contexts, including source filenames, file contents, and code entities, directly from the main toolbar
• Views
• Ability to directly access view preferences from a local toolbar
• Support for “compact mode” (specification and body on same line) and “flat mode” for view
• Colors
• Ability to change the color of many GPS elements, including selection of white foreground and dark background
• Support for color themes, to switch quickly among preselected settings

GPS 6.0 also brings improved performance and new capabilities:
• Use of a relational database for a more efficient (and persistent) storage mechanism for cross-tool integration
• Language support for SPARK 2014, plus syntax highlighting and toolips for Ada 2012 and SPARK 2014 aspects
• Several editor enhancements, ranging from completion and autobox to text formatting
• A number of additions to the scripting API

The GPS 6.0 enhancements have received an enthusiastic response from the product’s beta sites. An IDE is more than a tool, it is a platform and framework that developers depend on to organize and conduct their daily work. A major effort has thus been invested in improving the overall usability of GPS based on customer comments, giving it an intuitive and visually attractive look and feel. In short, and somewhat subjectively, the idea is to provide users with a “pleasant experience”. This has been a major objective for GPS 6.0 and will continue to drive product development.

Future enhancements to GPS are planned in several areas, including debugger integration, documentation generation, and support for aggregate projects.

A GPS 6.0 demo will be available at www.adacore.com/gps-demo. For further information, please contact info@adacore.com.