14th International Conference on

RELIABLE SOFTWARE TECHNOLOGIES

ADA-EUROPE 2009

BREST, FRANCE, 8-12 JUNE 2009

ADVANCE PROGRAM

http://www.ada-europe.org/conference2009.html
The 14th International Conference on Reliable Software Technologies – Ada-Europe 2009 will take place in Brest, France, on 8-12 June 2009. The conference has established itself as an international forum for providers, practitioners and researchers into reliable software technologies. Following tradition, the conference will span a full week, with at its centre from Tuesday to Thursday a three-day technical program accompanied by vendor exhibitions, and at either end on Monday and Friday a string of parallel tutorials and workshops.

Brest enjoys a wonderful setting, on the shores of a very large bay, with inland beauty and coastal Brittany surrounding it. The city is a centre of commerce, combining the cobblestone streets and fortifications of the old port with all the attractions and facilities of a modern city. Brest is linked to the whole of Europe by an international airport (flights from Paris, London, Nantes, Lyon, Nice, Marseille, Toulon, Birmingham, Exeter and Southampton).

The information in this document is still preliminary – please refer to the conference web site for the latest details.
INVITED SPEAKERS

Three eminent keynote speakers have been selected to open each day of the core conference program:

- **John Benito** (Blue Pilot Consulting, USA), a leading member of the international standardization and programming languages community, will deliver a talk entitled “ISO JTC 1/SC 22/WG 23 Work on Programming Language Vulnerabilities”.

- **Pierre Sens** (LIP6, Université Pierre et Marie Curie, Paris), a researcher in software technologies for distributed systems will discuss fault tolerance technologies in a talk entitled “Fault Tolerance in Large Scale Distributed Systems”.

- **Peter H. Feiler** (CMU/SEI, USA), a worldwide expert in architecture modeling and verification, will elaborate on “Validation of Safety-Critical Systems with AADL”.

ISO JTC 1/SC 22/WG 23 Work on Programming Language Vulnerabilities

John Benito,
Blue Pilot Consulting,
USA
(Tuesday 9, 9:00 - 10:00)

Any programming language has constructs that are imperfectly defined, undefined, implementation-dependent, or difficult to use correctly. As a result, software programs can execute in a manner that is different than intended by the developer. In some cases, the unintended functionality can be exploited by hostile parties or can lead to failure when used in unforeseen circumstances. The result can be a compromise of safety, security, privacy, dependability or some other critical property. Security vulnerabilities are a particular concern because an adaptive adversary can use a compromise in any executing program—even a non-critical vulnerability—as a springboard to make additional attacks on other programs. This presentation describes an effort to develop an authoritative account of the known weaknesses in programming languages and how developers might avoid those weaknesses.

Presenter
John is an independent consultant providing software development, project management, and software testing. He is the current Convenor of ISO/IEC JTC 1/SC 22/WG14 (C), the Convenor of ISO/IEC JTC 1/SC 22 WG 23 (was OWG Vulnerabilities), the project editor for the Technical Report 24772 (Guidance to avoiding vulnerabilities in programming language through language selection and use for registration), the Vice chairman of INCITS PL22 and a member of the INCITS PL22.11 (ANSI C) technical committee. He previously was a member of INCITS PL22.16 (ANSI C++) and the ISO Java Study group. He has been in software development, project management, and testing for over 30 years. John has been participating in International Standard development for the past 19 years.

Fault Tolerance in Large Scale Distributed Systems

Pierre Sens,
LIP6, Université Pierre et Marie Curie, Paris,
France
(Wednesday 10, 8:30 - 9:30)

Scaling to large configurations is one of the major challenges addressed by the distributed system community lately. As the number of hosts increases, the probability of a host failure converges to one. Compared to classical distributed systems, failures are more common and have to be efficiently processed.

In this talk, we address the problem of fault tolerance in large configuration like Grid or peer-to-peer systems. We focus on two fundamental issues: the failure detection and reliable data storage. We introduce the problem of failure detection in large asynchronous network where the transmission delay is not known and present recent studies on failure detection in grid configuration. Then, we present DHT-based P2P systems provide a fault-tolerant and scalable mean to store data blocks in a fully distributed way. Unfortunately, recent studies have shown that if connection/disconnection frequency is too high in the system (churn), data blocks may be lost. This is true for most current DHT-based system’s implementations. To avoid this problem, it is necessary to build really efficient replication and maintenance mechanisms. In
this talk, we study the effect of churn on an existing DHT-based P2P system namely Past/Pastry.

**Presenter**

Pierre SENS received his Ph.D. in Computer Science in 1994, and the "Habilitation à diriger des recherches" in 2000 from the Université Pierre et Marie Curie - Paris 6, France. Currently, he is a Full Professor at Université Pierre et Marie Curie and co-director of the LIP6 - Computer Science Laboratory of University Paris 6. His research interests include distributed systems, peer-to-peer file systems, fault tolerance and resource management in grid configurations. Pierre Sens is heading the Regal group which is a joint research team between LIP6 and INRIA.

**Validation of Safety-Critical Systems with AADL**

![Peter H. Feiler, CMU/SEI, USA](image)

**Peter H. Feiler, CMU/SEI, USA**
(Thursday 11, 8:30 – 9:30)

As safety-critical systems have become increasingly software intensive the embedded software system has become an increasing risk factor. The SAE Architecture Analysis & Design Language (AADL) international standard has been developed to support model-based engineering of embedded and real-time software intensive systems.

In this presentation we examine how AADL contributes to model-based validation of systems, to consistency between different analytical models of the same system, and validation of the implementation against the validated models. We will illustrate model-based analysis throughout the life cycle of different degrees of fidelity and formality with examples in terms of security, latency, and model checking of redundancy logic. The presentation concludes with an illustration of challenges in an implementation against the model.

**Presenter**

Dr. Peter Feiler is a 24 year veteran at the Software Engineering Institute (SEI). He is a senior member of the Research, Technology, and Systems Solutions (RTSS) program of the SEI. His interests include dependable real-time systems, architecture languages for embedded systems, and predictable system engineering. He has been the technical lead and author of the Society of Automotive Engineers (SAE) Architecture Analysis & Design Language (AADL) standard, which was originally published in November 2004 with a revision in January 2009, and AADL Annex standards published in July 2006. Dr. Feiler recently received the Carnegie Science Award for Information Technology. Prior he was employed by Siemens Corp. in Princeton, NJ, in research and advanced product development. Dr. Feiler has a Ph.D. in Computer Science from Carnegie Mellon.
Two workshops will be organized in conjunction with the conference.

Workshop on Software Vulnerabilities  
Joyce Tokar  
tokar@pyrrhusoft.com  
(W1: Monday 8 June, full day)

Given the large focus on software vulnerabilities in the current market place, ISO has instantiated the “Other Working Group: Vulnerabilities” as ISO WG23. ISO WG23 has submitted a Technical Report that captures the current view of software vulnerabilities. This workshop will focus on the content of this Technical Report and its applicability to Ada and SPARK. An accepted position paper is required for admission to the workshop.

The objective of the workshop is to define the Ada Annex to the ISO WG23 Technical Report on Vulnerabilities, and to identify breaks and gaps in this Technical Report with respect to the programming language Ada.

Workshop on AADL  
Bruce Lewis  
bruce.a.lewis@us.army.mil  
(W2: Friday 12 June, full day)

The AADL workshop will propose a session that emphasizes learning AADL concepts. Then a series of tool demonstrations is proposed to help you to apply these concepts to industrial case studies. The AADL provides a means to precisely describe the hardware/software architecture to support model-based architectural engineering early and throughout the lifecycle, supporting quantitative analysis and automated system integration. Both high-level AADL modeling tools allow you to capture your system design in a domain specific way, and then map it to AADL to perform analysis, lowering the AADL learning curve and tools that allow directly working in the language will be demonstrated.
TUTORIALS

The conference schedule includes 2 full days of tutorials running on 3 parallel tracks on Monday and Friday. The program this year features 10 tutorials, of which 2 are full-day and the others half-day, all delivered by recognized domain experts addressing a variety of topics within the general scope of the conference.

Building Cross Language Applications using Ada

Quentin Ochem, AdaCore, France
(T1: Monday 8 June, full day)

Building complex applications often requires putting together pieces of software or requirements that have not been made to work together in the first place. Thinking of a project with a high integrity kernel written in Ada, using a set of low level libraries and drivers written in C or C++, with a graphical interface done in Java and unit tests driven by python is not thinking of science-fiction anymore. It’s actual concrete and day-to-day work. Unfortunately, having all of these technologies talking one to each other is not straightforward, and often requires a deep knowledge of both sides of the technology and extensive manual work. In this tutorial, we'll first study how to interface directly Ada with native languages, such as C or C++. We'll then have a deep look at communications with languages running on virtual machines, such as Java, Python and the .NET framework. Finally, we'll see how Ada can be interfaced with an arbitrary language using a middleware solution, such as SOAP or CORBA. We'll see how the communication can be manually done using low level features and APIs, and how a substantial part of this process can be automated using high level binding generators.

Presenter

Quentin Ochem has been working with the Ada language for the past eight years and joined AdaCore in 2005. He is one of the main programmers of the GNATbench Eclipse plugin for Ada, and implemented a binding generator from Ada to Java in this context, which is now distributed as a standalone product by AdaCore. He has been working with the GNAT Programming Studio scripting capabilities as well, which are based on an Ada/Python interfacing technology.

Why attend this tutorial?

This tutorial will give a broad overview of the possibilities of communications between Ada and other languages, presenting advantages and drawbacks of different solutions, plus state-of-the art information. Any developer facing this kind of issue will get a detailed view on different solutions currently available.

Introduction to Parallel and Real-Time Programming with Ada

John W McCormick, University of Northern Iowa, USA
(T2: Monday 8 June, morning)

This tutorial provides an in-depth description of the Ada features used in concurrent and real-time programming. Concurrent programming is the name given to programming notation and techniques for expressing potential parallelism and solving the resulting synchronization and communication problems (Moti Ben-Ari, 1982). The first two thirds of the tutorial will introduce you to Ada's support for concurrency. Ada has integrated concurrency and synchronization as part of the language. We will discuss the notion of the task, inter-task communication (entries, rendezvous, select) and protected objects. We will illustrate these constructs through classical and practical examples. In the final third of the tutorial, we will discuss Real-Time programming, that is concurrent programming with the addition of timing constraints. We will examine the aspects of Ada used to remove non-determinism from our concurrent programs. We will look at time and clocks, deterministic dispatching, and the resource

**Presenter**

John W. McCormick (PhD, University of California, Los Angeles, 1977) has been a Professor of Computer Science at the University of Northern Iowa since 1996. Before then, he was a Professor of Computer Science at the State University of New York (SUNY), Plattsburgh. In 1993, he received the SUNY Chancellor's Award for Excellence in Teaching. He is the primary author of two introductory Ada textbooks (both in second edition): Programming and Problem Solving with Ada and Ada Plus Data Structures: An Object-Oriented Approach. Most of his publications are in the area of computer science education. His papers on the use of model railroads in his undergraduate real-time systems courses received the award for best presentation at both the ACM SIGAda and Ada-Europe conferences. He received the ACM SIGAda Distinguished Service Award in 2002 and the ACM SIGAda Outstanding Ada Community Contributions award in 2008. He is currently Chair of ACM SIGAda.

**Why attend this tutorial?**

To learn the fundamentals of concurrent and real-time programming with Ada.

---

**Software Fault Tolerance**

Pat Rogers, AdaCore, USA

(T3: Monday 8 June, afternoon)

Software for current safety-critical applications – e.g., flight control systems – is both large and complex, such that full testing is not feasible. Furthermore, complete proofs of correctness are at best inherently limited by the potential for specification errors. The combination of potential specification errors and overall complexity defines a problem of handling unanticipated software faults. “Software fault tolerance” is the use of software mechanisms to deal with these unanticipated software faults. This half-day tutorial explores the software-based techniques and mechanisms available for tolerating unanticipated software design faults in safety-critical systems. We examine the rationale for tolerating software faults, the similarities to mechanisms for tolerating hardware faults, and the advantages and disadvantages of the common techniques. Special attention is paid to the concept of design diversity as the underlying theory for the most widely-used mechanisms (e.g., N-Version Programming) and, in particular, whether design diversity can achieve the extremely low failure rates required for safety-critical systems. The mechanisms explored are illustrated with concrete implementations using Ada 95 and Ada 2005. Participants will have an appreciation of the necessity for tolerating software faults as well as a firm foundation for further study and informed application of available mechanisms. A bibliography of suggested reading is provided to that end.

**Presenter**

Patrick Rogers is a senior Member of the Technical Staff with Ada Core Technologies, specializing in high-integrity and real-time application support. A computing professional since 1975 and an Ada developer since 1980, he has extensive experience in real-time applications in both embedded bare-board and POSIX-based environments. An experienced lecturer and trainer since 1981, he has provided numerous tutorials and courses in software fault tolerance, hard real-time schedulability analysis, object-oriented programming, and the Ada programming language. He holds B.S. and M.S. degrees in computer science from the University of Houston and a Ph.D. in computer science from the University of York, England, in the Real-Time Systems Research Group on the topic of software fault tolerance.

**Why attend this tutorial?**

Systems software architects and developers responsible for safety-critical software will gain an appreciation for software fault tolerance facilities, including their limitations, and will have a firm foundation for informed application of available mechanisms as well as further study.

---

**Software Measures For Building Dependable Software Systems**

William Bail, MITRE, USA

(T4: Monday 8 June, morning)

For any software development effort, being able to track progress in the development process, as well as monitoring the system’s...
projected quality attributes levels, is a crucial activity. For dependable systems, this is even more critical, because developing dependable software systems is a continuing challenge. It is important to be able to track the emerging attributes of the system so that corrective action can be performed early enough to avoid costly rework. The effective and efficient use of software measurement can assist in this goal. One key is to select and apply the right measures, and to avoid those that do not provide useful information. There is a temptation to collect many metrics with the idea that more data is better. Such a collection strategy results in wasted effort and at times misleading indicators such as false positives and false negatives. This tutorial surveys the range of popular and effective measures, and provides guidelines for their selection, application, and interpretation. It provides an assessment of the utility of many popular measures, and makes recommendations for effective subsets that provide cost-effective feedback and predictive information.

**Presenter**

Since 1990, Dr. Bail has worked for The MITRE Corporation in McLean VA as a Computer Scientist in the Software Engineering and Computing (SWEC) Department. MITRE is a not-for-profit corporation chartered to provide systems engineering services to the U.S. Government agencies, primarily the DoD, the FAA, and the IRS. Within MITRE, the SWEC supports customer programs, consulting on various aspects of software development, ranging from requirements elicitation and management, to design techniques and practice, and to testing. The SWEC particularly focuses on transitioning emerging technologies into practice. Dr. Bail’s technical areas of focus include requirements engineering, dependable software design and assessment, error handling policies, design methodologies, metric definition and application, and verification and validation. At MITRE, Dr. Bail is currently supporting the U.S. Navy, focusing on the practice of software engineering as applied to large real-time systems. Prior to 1990, Dr. Bail worked at Intermetrics Inc. in Bethesda MD. Since 1989 he has served as an Adjunct Professor at the University of Maryland University College where he develops instructional materials and teaches courses in software engineering, in topics such as Software Requirements, Verification and Validation, Software Design, Software Engineering, Fault Tolerant Software, and others. Previously, Dr. Bail taught part-time at The University of Maryland from 1983-1986 in the Computer Science Department for undergraduate courses in discrete mathematics, computer architecture, and programming language theory. Dr. Bail received a BS in Mathematics from Carnegie Institute of Technology, and an MS and Ph.D. in Computer Science from the University of Maryland. Dr. Bail has presented tutorials on Cleanroom Software Engineering, Semi-Formal Development Techniques, Statistical Testing, and Requirements Engineering for Dependable Systems at SIGAda, Ada-Europe, the NDIA Systems Engineering Conference, SSTC, and other conferences.

**Why attend this tutorial?**

Measurement of software systems is a key activity that aids in their development. Developing a complex software system without collecting appropriate metrics is like flying a plane in the fog without instruments and radar. It is crucial to collect metrics and to use them as an integral part of the development process. In order to avoid costly and time-intensive rework late in the development effort, collecting metrics allows developers to be able to track the emerging attributes of the system so that corrective action can be performed early. It is important however to select those measures that provide information that is useful for the development process. Mindless collection of numbers will only consume time and effort, and may distract developers from the true risks to the system. This tutorial will provide the attendee with the basic understanding of the role of software measures and how they can be used to support the management of a software development project. The tutorial will also discuss different categories of measures and how to select specific measures based on the nature of the development effort. The information presented will help in ensuring that a metrics program will focus on measures that provide utility and that are cost-effective, ensuring that any emerging issues are identified early providing an opportunity for correction.

**Modeling for Schedulability Analysis with the UML Profile for MARTE**

Julio Medina, Universidad de Cantabria, Spain.
Huascar Espinosa, CEA-List, France.

(T10: Monday 8 June, afternoon)

MARTE, the UML Profile for Modeling and Analysis of Real-Time and Embedded Systems, is the new standard provided by the Object Management Group to complement UML with modelling constructs useful for
the real-time and embedded systems communities. A challenging problem in model-based engineering is to integrate models that are commonly used for system production or software code generation with the information that is relevant to perform analysis. The goal is to reduce the time required to prepare a design model for performing analysis and to ensure greater accuracy of an analysis model by directly associating it with the actual system model. A large part of modelling constructs in MARTE are specific for analysis, and extend the capabilities of its predecessor, the "UML Profile for schedulability, Performance, and Time" specification by enlarging the kinds of resources, providing new mechanisms for the specification of non-functional properties, and additional modelling artifacts, as well as by specifying it for UML2, the latest version of UML. Considering its role for the enabling of schedulability analysis techniques in a model driven engineering approach, and being scheduling a key factor in the assessment and assurance of software reliability, this tutorial presents the capabilities in the MARTE profile that provide support for schedulability analysis. These annotations are used to add supplementary information to various kinds of UML elements that can then be interpreted by specialized tools or domain experts. For instance, fine-grained timing analyzers can help to determine the worst case execution times of relevant pieces of code, which are then used in schedulability analysis to predict end-to-end response times. These annotations are defined in the schedulability Analysis Modelling (SAM) sub-profile of MARTE.

Presenters
Julio Medina is graduated as Electronics Engineer from the Universidad Nacional de Ingeniería, Perú, in 1987. He obtained the Master in Real-Time Systems and the Doctorate in Telecommunications Engineering from the University of Cantabria, Spain, in 1993 and 2005 respectively. He developed electronics and embedded software for nuclear instrumentation in the Peruvian Nuclear Research Centre, he then worked as assistant professor in electronics instrumentation and software engineering, and did research on distributed real-time systems. During the doctorate period he concentrated on the modelling of real-time systems developed with object-oriented techniques, being later co-author of the UML Profile for MARTE of the OMG. Currently he is lecturer in the University of Cantabria and does research in a number of European projects. His main research topics include the modelling of real-time distributed systems for schedulability analysis, the UML representation of such models, and its usage for component-based development strategies and standardization. Huascar Espinoza is a researcher of the Model Driven Engineering Labs at CEA (French Atomic Energy Commission). He holds a Ph.D. in Computer Science from the University of Evry and CEA. His research interests centre on model-driven development of real-time embedded systems, architecture evaluation, and verification techniques. He is co-author of the OMG standard for Modelling and Analysis of Real-Time and Embedded systems (MARTE) with UML. His duties in MARTE included the leading of the non-functional properties and the schedulability analysis modelling frameworks. He currently participates in a number of European and French research projects for large automotive, avionics, and transport industries. He is member of the OMG’s finalization and revision task forces for MARTE and SysML.

Why attend this tutorial?
UML is a language; MARTE brings additional vocabulary to UML. This tutorial will help in identifying the precise semantics and modelling intent of this additional vocabulary. It is given by significant contributors to the standard, and will show its practical application in the utilization of UML models for schedulability analysis.

SPARK : The Libre Language and Toolset for High-Assurance Software

Roderick Chapman,
Praxis High Integrity Systems, UK
(T5: Friday 12 June, full day)

SPARK is an annotated sub-language of Ada which is unambiguous and suitable for rigorous static analysis. It has been extensively used in industrial applications where safety and security are paramount. The tutorial, which is extracted from the four-day "Software Engineering with SPARK" course will cover: The rationale and design goals of SPARK, the core SPARK language, and SPARK analyses including information flow, exception freedom, and formal verification. The tutorial will also cover the forthcoming "Libre" release of the SPARK technology and the research topics that this opens, such as more advanced forms of analysis and proof, model-checking and language extensions.
Presenter
The principal presenter will be Dr Roderick Chapman of Praxis High Integrity Systems. Rod is a well-known conference speaker. He has presented papers, tutorials and workshops at many international events including STC, NSA HCSS, and ACM SIGAda. He was the opening key-note speaker at Ada Europe 2006. Rod has been involved with the design of both safety and security-critical software with Praxis for many years, including significant contributions to many of Praxis' key-note projects such as SHOLIS, MULTOS CA. Tokeneer, and the development of the SPARK language and verification tools. Rod is a Chartered Engineer and a Fellow of the BCS.

Why attend this tutorial?
To learn about the trade-offs in the design of programming languages and static analysis tools. To learn about what such tools can and can't offer in terms of soundness, depth, efficiency and completeness of analysis. To appreciate the SPARK language, and the capabilities of the SPARK toolset, and how these "fit" into a larger high-assurance software process. To propose, discuss and learn about research themes offered by the Libre release of the SPARK technology.

Hard Real Time and Embedded Systems Programming
Pat Rogers, AdaCore, USA
(T6: Friday 12 June, morning)
Although the terms are often used interchangeably, real-time systems need not be embedded, and embedded systems need not have deadlines. However, applications in both domains are expensive and labor-intensive, especially because developers typically have only low-level tools available and must use techniques that are more ad hoc than analytical. Ada 2005 represents the state-of-the-art in real-time programming languages and offers a high-level model for low level programming that is unsurpassed in expressive power. This intensive course covers the modern analytical techniques for determining whether deadlines will be met, the Ada language facilities required to support those analyses, and the high-level model Ada provides for embedded systems development. In addition, the issues involved in storage management are covered so that, in addition to time,
equally important storage resources are available when required.

Presenter
See tutorial T3.

Why attend this tutorial?
Developers will understand the fundamentals of schedulability analysis and the unmatched real-time scheduling facilities provided by Ada. In addition, the low-level programming facilities, so often mis-understood, are covered in detail. This tutorial is intended for developers familiar with some of the more advanced features of Ada, including tasking and access types.

Designing Real Time, Concurrent and Embedded Software Systems with UML and Ada
Rob Pettit, The Aerospace Corporation, USA
(T7: Friday 12 June, afternoon)
The domain of real-time, concurrent, and embedded software is becoming increasing complex. To effectively develop these systems, greater care must be taken to construct adequate models of the software and to effectively analyze these designs prior to code development. In this tutorial, we will discuss modeling and analysis issues specific to real-time, concurrent, and embedded software systems. Specifically, this tutorial will present guidelines for modeling these systems using the Unified Modeling Language (UML) version 2. A case study will be introduced to solidify the concepts and participant interaction will be encouraged in the construction of the models. Analytical methods will also be discussed to verify that the UML-based designs will produce the desired behavior. Finally, we will discuss how to proceed from the UML models into an Ada (2005) implementation.

Presenter
Rob Pettit has over 20 years of experience in the software development industry and is an internationally recognized expert in the field of software design for object-oriented real-time and concurrent systems. In his
current position as Sr. Project Leader for The Aerospace Corporation, Dr. Pettit provides technical leadership and research direction for the Office of Software Modeling and Analysis (OSMA), which he helped to found in 2006. In addition to his leadership role for the OSMA, Dr. Pettit is responsible for oversight, consulting, and mentoring for a number of object oriented software development projects. Dr. Pettit has also served as the principle investigator on multiple Aerospace-funded research programs primarily focused on applying object-oriented technology to real-time embedded systems. Dr. Pettit has taught numerous industrial courses on software development and object oriented technology; has given numerous presentations to international conferences and workshops; and has been invited to speak at international venues related to real-time and embedded software. Dr. Pettit is also an Adjunct Professor of Computer Science at George Mason University where he teaches object-oriented software analysis and design for the software engineering master’s program and where he established a course on mode-driven real-time and embedded software. Dr. Pettit received his B.S., Computer Science degree in 1991 from the University of Evansville, his M.S., Software Systems Engineering degree in 1995 from the George Mason University, and his Ph.D. in Information Technology / Software Engineering from George Mason University in 2003. Dr. Pettit was the General Co-Chair for the Americas of IEEE’s 2007 and 2008 International Symposium on Object-oriented Real-time Computing (ISORC 2007 and ISORC 2008) and is the General Chair for the 2009 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS 2009). Dr. Pettit is a Senior Member of the IEEE; a member of ACM's SIGAda; and a co-author of Ada 95 Quality and Style.

Why attend this tutorial?
The domain of real-time, concurrent, and embedded software is becoming increasing complex. To effectively develop these systems, greater care must be taken to construct adequate models of the software and to effectively analyze these designs prior to code development. In this tutorial, we will discuss modeling and analysis issues specific to real-time, concurrent, and embedded software systems. Specifically, this tutorial will present guidelines for modeling these systems using the Unified Modeling Language (UML) version 2. A case study will be introduced to solidify the concepts and participant interaction will be encouraged in the construction of the models. Analytical methods will also be discussed to verify that the UML-based designs will produce the desired behavior. Finally, we will discuss how to proceed from the UML models into an Ada implementation.

This tutorial is structured for software and systems engineers responsible for specifying, modeling, and analyzing real-time, concurrent, and embedded software systems. The tutorial assumes basic knowledge of object-oriented design concepts as well as concepts surrounding real-time and embedded systems. Basic knowledge of the Ada programming language and its features for concurrency are also assumed.

Object-Oriented Programming in Ada 2005
Matthew Heaney,
On2 Technologies, USA
(T8: Friday 12 June, morning)

This tutorial will teach attendees how to use the new Ada 2005 language features for object-oriented programming. The tutorial will include information about annotating primitive operations, new distinguished-receiver call syntax, interface types, extended return statements, limited-with and private-with package dependencies, limited aggregates and box defaults, generic formal primitive operations, and additions to the predefined language environment. In particular, it will discuss the full integration of the language features for concurrency into the type model. The tutorial is a half-day in duration, and its intended level is intermediate. Attendees should have a modest experience with Ada programming in general, and Ada 2005 in particular. I have presented many tutorials at both SIGAda and Ada-Europe, on the subjects of object-oriented programming, design patterns, system design, and container libraries. I have been programming in Ada for almost 20 years, and have designed several large, real-time systems in Ada. I was the designer of the Ada 2005 container library, which was based on work I had done to port the C++ STL to Ada.

Presenter
Matthew Heaney is a developer at On2 Technologies, where he writes Flash Media authoring tools. He has had a long career using Ada technology, having designed several large real-time systems in Ada. Matthew participated in the standardization of the Ada 2005 container library (which was based on the Charles library, his port of the STL to Ada95), and more recently has been involved in development of bounded container forms. Matthew has presented many Ada tutorials at both SIGAda and Ada-Europe, on the subjects of object-
oriented programming, design patterns, system design, and container libraries.

Why attend this tutorial?
This tutorial will teach attendees about the Ada 2005 language features for object-oriented programming. The tutorial will include information about new procedure call syntax, interface types and tagged types, extended return statements, anonymous access return types, implicit aliasing and dereference, limited aggregates, and other improvements. It will also discuss new idioms for subsystem design (including a solution to the problem of mutual package dependencies), and demonstrate with many examples how the features for concurrency have now been fully integrated into the Ada type model.

**Execution Time : Analysis, Verification and Optimization in Reliable System**

**Ian Broster,**
**Rapita Systems, UK**
(T9: Friday 12 June, afternoon)

“How long does my software take to run and how can I reduce it?” are key issues for embedded engineers designing reliable systems. Understanding, verifying, and improving the timing performance of their real-time products gives successful companies a key competitive edge. This tutorial will cover key aspects on-target timing analysis: finding worst case execution times (WCET) and the worst case path, and why worst case optimization is not the same as average execution optimization. Practical issues of looking for the worst-case "hot-spots", identifying timing bugs and verifying optimisation opportunities will be explained with examples of worst case optimizations. The tutorial first focuses on understanding execution times of software, why they are important, why they vary and how to find them. We discuss different sorts of execution time: minimum, average, maximum measured, worst case and different ways of finding them. Low-level (e.g. hardware features) and high-level (e.g. path through the software) issues are discussed and the impact that they have on finding execution times. Tool support for understanding the timing behaviour, including worst case execution times is explained. Finally, methods for reducing (optimization) of the worst-case execution time are explored. We make the distinction between average case optimizations and worst-case optimizations and how worst-case optimizations are important for reliable systems. A simple procedure for managing and performing software optimization in reliable systems is suggested with examples and results of optimization of the worst-case execution times in a large Ada project.

**Presenter**
Dr Ian Broster is a founder and Director of Rapita Systems Ltd, a company specializing in software timing analysis. He is an experienced, lively lecturer who has given numerous training courses, lectures and presentations on this and other topics. He has been involved with Ada for several years. He earned his PhD at the Real-time Systems group of University of York.

**Why attend this tutorial?**
Today, software timing analysis doesn't have to be guesswork. This tutorial addresses two key aspects of real-time systems performance: (1) how to gain a clear, detailed, and accurate understanding of the execution time and worst case execution time (WCET) behaviour of real-time embedded software, (2) how to target optimisation effort precisely where it will have the maximum benefit in improving system timing behaviour, for the minimum cost.
<table>
<thead>
<tr>
<th>Time</th>
<th>Tuesday 9 June</th>
<th>Wednesday 10 June</th>
<th>Thursday 11 June</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:30 – 9:00</td>
<td>Welcome and Opening Address</td>
<td>Keynote Talk</td>
<td>Keynote Talk</td>
</tr>
<tr>
<td>9:00 – 9:30</td>
<td>Keynote Talk</td>
<td>Pierre Sens</td>
<td>Peter H. Feiler</td>
</tr>
<tr>
<td></td>
<td>John Benito</td>
<td>Fault tolerance in large scale distributed systems</td>
<td>Validation of Safety-Critical Systems with AADL</td>
</tr>
<tr>
<td></td>
<td>JTC 1/SC 22/WG 23 work on Programming Language Vulnerabilities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:30 – 10:00</td>
<td>Coffee &amp; Exhibition</td>
<td>Coffee &amp; Exhibition</td>
<td>Coffee &amp; Exhibition</td>
</tr>
<tr>
<td>10:00 – 10:30</td>
<td>Real Time</td>
<td>Model-Driven Engineering and AADL</td>
<td></td>
</tr>
<tr>
<td>10:30 – 11:00</td>
<td>High Integrity</td>
<td>Implementation of the Ada 2005 Task Dispatching Model in MaRTE OS and GNAT, Mario Aldea Rivas, Michael Gonzalez Harbour and José F. Ruiz</td>
<td>Modeling AADL Data Communication with BIP, Lei Pi, Jean-Paul Bodeveix and Mamoun Filali-Amine</td>
</tr>
<tr>
<td>11:00 – 12:30</td>
<td>Requirements on a Target Programming Language suited for a High-Integrity MDE Environment, Alessandro Zovi and Tullio Vardanega</td>
<td>Combining EDF and FP: Analysis and Implementation in Ada 2005, Alan Burns, Andy Wellings and Fengxiang Zhang</td>
<td>Formal Verification of AADL Specifications in the Topcased Environment, François Vernadat, Bernard Berthomieu, Jean-Paul Bodeveix, Christelle Chaudet, Silvano Dal Zilio and Mamoun Filali</td>
</tr>
<tr>
<td></td>
<td>A restricted middleware profile for high-integrity distributed real-time systems, Santiago Urueña, Juan Zamorano and Juan Antonio de la Puente</td>
<td>Predicated Worst-Case Execution-Time Analysis, Amine Marref and Guillem Bernat</td>
<td>Process-Algebraic Interpretation of AADL Models, Oleg Sokolsky, Insup Lee and Duncan Clarke</td>
</tr>
<tr>
<td></td>
<td>Validating safety and security requirements for partitioned architectures, Julien Delange, Laurent Pautet and Peter Feiler</td>
<td></td>
<td>OCARINA: An Environment for AADL Models Analysis and Automatic Code Generation for High Integrity Applications, Lasnier Gilles, Zalila Bechir, Pautet Laurent and Hugues Jérôme</td>
</tr>
<tr>
<td>12:30 – 14:00</td>
<td>Lunch &amp; Exhibition</td>
<td>Lunch &amp; Exhibition</td>
<td>Lunch &amp; Exhibition</td>
</tr>
<tr>
<td>Time</td>
<td>Tuesday 9 June</td>
<td>Wednesday 10 June</td>
<td>Thursday 11 June</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>14:00 - 15:00</td>
<td>Testing</td>
<td>Model-Driven Engineering</td>
<td>Vendor session</td>
</tr>
<tr>
<td></td>
<td>Model Checking Techniques for Test Generation from Business Process Models, Didier Buchs, Levi Lacio and Ang Chen</td>
<td>Modelling and Evaluating Real-Time Software Architectures, Jose L. Fernandez and Gloria Marmol</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A Formal Foundation for Metamodeling, Liliana Favre</td>
<td>A Formal Foundation for Metamodeling, Liliana Favre</td>
<td></td>
</tr>
<tr>
<td>15:00 - 15:30</td>
<td>Coffee &amp; Exhibition</td>
<td>Model-Driven Engineering</td>
<td>Vendor session</td>
</tr>
<tr>
<td>15:30 - 16:00</td>
<td>Education</td>
<td>Coffee &amp; Exhibition</td>
<td>GAP meeting</td>
</tr>
<tr>
<td>16:00 - 16:30</td>
<td>Vendor session</td>
<td>Education</td>
<td>Coffee break</td>
</tr>
<tr>
<td></td>
<td>Using Java or C# Monitor for a concurrency kernel implies defensive multithreading programming, Claude Kaiser and Jean-François Prudat-Peyre</td>
<td>Pattern-Based Refactoring Shrinks Maintenance Costs, John S. Harbaugh</td>
<td>Ensuring Software Integrity</td>
</tr>
<tr>
<td>17:00 – 17:30</td>
<td>Welcome Reception</td>
<td>Ada-Europe General Assembly</td>
<td>Coffee break</td>
</tr>
<tr>
<td>17:30 - 18:00</td>
<td></td>
<td></td>
<td>Closing Address and Best-Presentation Award</td>
</tr>
<tr>
<td>18:00 - 19:00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>From 19:00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An informal GNAT Academic Program (GAP) meeting will be held during the afternoon Wednesday break (15h30-16h30) to update members on the latest news and exchange views and ideas on the future development of the program. All Ada academics and students are welcome to attend.
# Tutorial and Workshop Schedule

<table>
<thead>
<tr>
<th>Day</th>
<th>Time</th>
<th>Tutorial/Workshop</th>
<th>Presenter/Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 June</td>
<td>W1</td>
<td>Full day</td>
<td>Joyce Tokar (Pyrrhus Software, USA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Workshop on Software Vulnerabilities</td>
</tr>
<tr>
<td></td>
<td>T1</td>
<td>Full day</td>
<td>Quentin Ochem (AdaCore, France)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Building Cross Language Applications using Ada</td>
</tr>
<tr>
<td></td>
<td>T2</td>
<td>Morning</td>
<td>John Mc Cormick (University of Northern Iowa, USA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>An Introduction to Parallel and Real-Time Programming with Ada</td>
</tr>
<tr>
<td></td>
<td>T3</td>
<td>Afternoon</td>
<td>Pat Rogers (AdaCore, USA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Software Fault Tolerance</td>
</tr>
<tr>
<td></td>
<td>T4</td>
<td>Morning</td>
<td>William Bail (MITRE, USA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Software Measures for Building Dependable Software Systems</td>
</tr>
<tr>
<td></td>
<td>T10</td>
<td>Afternoon</td>
<td>Julio Medina (Universidad de Cantabria, Spain), Huascar Espinoza (CEA-List, France)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Modeling for Schedulability Analysis with the UML Profile for MARTE</td>
</tr>
<tr>
<td>Friday</td>
<td>W2</td>
<td>Full day</td>
<td>Bruce Lewis (US Army, USA)</td>
</tr>
<tr>
<td>12 June</td>
<td></td>
<td></td>
<td>Workshop on AADL</td>
</tr>
<tr>
<td></td>
<td>T5</td>
<td>Full day</td>
<td>Roderick Chapman (Praxis High Integrity Systems, UK)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SPARK - The Libre Language and Toolset for High-Assurance Software</td>
</tr>
<tr>
<td></td>
<td>T6</td>
<td>Morning</td>
<td>Pat Rogers (AdaCore, USA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Hard Real-Time and Embedded Systems Programming</td>
</tr>
<tr>
<td></td>
<td>T7</td>
<td>Afternoon</td>
<td>R. D. Pettit (The Aerospace Corporation, USA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Designing Real-Time, Concurrent, and Embedded Software Systems using UML and Ada</td>
</tr>
<tr>
<td></td>
<td>T8</td>
<td>Morning</td>
<td>Matthew Heaney (On2 Technologies, USA)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Object-Oriented Programming in Ada 2005</td>
</tr>
<tr>
<td></td>
<td>T9</td>
<td>Afternoon</td>
<td>Ian Broster (Rapita Systems, UK)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Execution Time: Analysis, Verification, and Optimization in Reliable System</td>
</tr>
</tbody>
</table>

Morning tutorial sessions will start at 9:30 and end at 13:00. Afternoon sessions will start at 14:30 and end at 18:00. Coffee breaks will be at 11:00 - 11:30 and at 16:00 - 16:30.
EXHIBITION

The exhibition will open at the morning break on Tuesday and run continuously until the end of the afternoon break on Thursday. It takes place in the heart of the conference centre just next to the meeting halls. The coffee breaks are held in the exhibition area. The breaks on Tuesday to Thursday span one hour so as to allow the attendees comfortable time to visit the exhibition.

Companies that have already booked their exhibition space include: AdaCore, Aonix, Ellidiss Software, IBM and Rapita Systems. Others have shown interest. Exhibitors will also deliver a presentation in the vendor session.

SOCIAL PROGRAM

Welcome Reception

The social program of the conference will open with a welcome reception at Oceanopolis: Brittany’s sea park by the Marina in Brest. The Sea Park is organized around three buildings which highlight the diversity and the habitat of the seas around the world: a temperate building which presents sea life around Brittany, the tropical building displays tropical and colorful fishes and sharks, the polar building is home to about 40 penguins. The 50 aquariums make it possible to observe the various animals and plants in their habitat. Ellidiss Technologies sponsors the reception at Oceanopolis.

Conference Banquet

The conference banquet will take place by the seaside, in the charming village of Porspoder, located 25 km northwest of Brest: this area provides a spectacular landscape of rough cliffs and sandy coves, and impressive marine streams due to the collision of the Channel and the Atlantic waters. AdaCore sponsors the banquet.

Additional tickets for the welcome reception on Tuesday 9 and for the excursion and banquet on Wednesday 10 can be purchased on site at the registration desk.

La Pointe du Raz (a famous rocky peak)
REGISTRATION AND ACCOMMODATION

Conference Registration
The registration fee for the three days of the technical program (June 9-11) includes one copy of the proceedings, coffee breaks, lunches, welcome reception on Tuesday 9 June evening, banquet on Wednesday 10 June evening. The registration fee for a single day of the technical program includes one copy of the proceedings, two coffee breaks, and the lunch on the day of the registration.

<table>
<thead>
<tr>
<th>Member of Ada-Europe or ACM SIGAda</th>
<th>Non-member</th>
<th>Student(**)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Academia</td>
<td>Non academia</td>
<td>Academia</td>
</tr>
<tr>
<td>Early registration (payment by 22 May)</td>
<td>480 €</td>
<td>540 €</td>
</tr>
<tr>
<td>Late/on-site registration (payment after 22 May)</td>
<td>600 €</td>
<td>660 €</td>
</tr>
<tr>
<td>Day registration (single day)</td>
<td>300 €</td>
<td>330 €</td>
</tr>
</tbody>
</table>

(*) Student discount: please check the Registration page at http://www.ada-europe.org/conference2009.html for the eligibility conditions to student discounts.

Tutorial and Workshop Registration
The fee is per tutorial, including tutorial notes and coffee breaks. Lunches are only included when registered for a full day tutorial or two half day tutorials on the same day.

<table>
<thead>
<tr>
<th>Early registration (payment by 22 May)</th>
<th>Half day</th>
<th>Full day or two half days on the same day</th>
<th>Full day Workshop (**)</th>
</tr>
</thead>
<tbody>
<tr>
<td>130 €</td>
<td>260 €</td>
<td>75 €</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Late/on-site registration (payment after 22 May)</th>
<th>Half day</th>
<th>Full day or two half days on the same day</th>
<th>Full day Workshop (**)</th>
</tr>
</thead>
<tbody>
<tr>
<td>145 €</td>
<td>290 €</td>
<td>75 €</td>
<td></td>
</tr>
</tbody>
</table>

(**) Workshop participant: please check with the workshop organizers for the participation requirements.

No registration request will be confirmed until the payment has been processed. Substitutions will be accepted. To save on administrative costs and postage, receipts will be handed out at the conference. Cancellation must be always confirmed in writing: don't forget to include all your banking information. Refund of fees, with deduction of € 25.00 for administrative duties, will be made for cancellations received before 22 May 2009. After that date, no refunds will be possible. Refunds will be processed and paid after the Conference.

For latest information, see http://www.ada-europe.org/conference2009.html. For additional information, contact Yvon Kermarrec, tel: +33-229-001285, fax: +33-229-001030, e-mail: ae2009-org@mlistes.telecom-bretagne.eu.

Accommodation
Information on hotel accommodation is available at the conference website. Brittany is a tourist attraction at all times, especially during Spring and Summer. Therefore, book your accommodation as soon as possible. If you want to book the resort at Le Trez-Hir, please check the price on the conference website, contact the local chair for availability (e-mail: ae2009-reg@mlistes.telecom-bretagne.eu).
Please return this form by fax or e-mail to:
Yvon Kermarrec
Telecom Bretagne
Brest (France)
tel. +33 2 29 00 12 85
fax. +33 2 29 00 10 30
e-mail: ae2009-reg@mlistes.telecom-bretagne.eu

Participant Registration Form
14th International Conference on Reliable Software Technologies – Ada-Europe 2009
8-12 June 2009, Brest, Brittany, France
http://www.ada-europe.org/conference2009.html

Please use block capitals

<table>
<thead>
<tr>
<th>Participant</th>
<th>Ms [ ] Mr [ ] Title: ____________</th>
</tr>
</thead>
<tbody>
<tr>
<td>First name:</td>
<td>Last name: _______________________</td>
</tr>
<tr>
<td>Affiliation/Organization:</td>
<td>________________________________</td>
</tr>
<tr>
<td>Address:</td>
<td>________________________________</td>
</tr>
<tr>
<td>City: ____________________________</td>
<td>State: ____________________________</td>
</tr>
<tr>
<td>Country: _________________________</td>
<td>Zip/Postal code: __________________</td>
</tr>
</tbody>
</table>
| Phone: __________________________ | Fax: _____________________________ | Email: ___________________________

Special requirements (e.g. diet): ________________________________

There are a number of other “in cooperation” conferences in this area (such as ACM SIGAda). Indicate whether you agree to receive information about such conferences: [ ] Yes [ ] No

Reduced registration fee
[ ] Member Ada-Europe; national organization / direct / etc. _____
[ ] Academia : institute _________________________________
[ ] Member ACM; membership number _______________________

Additional Comments: ____________________________________________

Registration Type: Early (payment by 22 May) [ ] Late or on site (payment after 22 May) [ ]

Registration Fees (see table on previous page or web site)
Conference registration fee
- Three days registration: _____________ EUR
- Individual days (Tue [ ] Wed [ ] Thu [ ]): _____________ EUR

Tutorial registration (please indicate the tutorials for which you want to register):
- Monday, 8 June: T1 [ ] T2 [ ] T3 [ ] T4 [ ] T10 [ ]
- Friday, 12 June: T6 [ ] T7 [ ] T8 [ ] T9 [ ] T5 [ ]

Tutorial registration fee: _____________ EUR

Workshop: Monday on Software vulnerability [ ] Friday on AADL [ ] 75 Euro each

Extra tickets for welcome reception (@ 50 EUR each): _____________ EUR
Extra tickets for banquet (@ 80 EUR each): _____________ EUR
Extra tickets for lunch (@ 30 EUR each): _____________ EUR
Extra proceedings (@ 30 EUR each): _____________ EUR

Accomodation at Le Trez Hir / Plougonvelin
- indicate the number in your party [ ] and check the price on the web _____________ EUR

Total Payment Due: _____________ EUR
The conference web site at http://www.ada-europe.org/conference2009.html gives full and up to date details of the program. The web site also provides details of the venue, including travel advice, instructions to reserve hotel accommodation in the conference venue, maps and a list of hotels close by.

Exhibiting and Sponsoring details are also on the web site; a sliding scale of sponsorship provides a range of benefits. All levels include display of your logo on the conference web site and in the program. The lowest level of support is very affordable.
ORGANIZATION

Conference Chair:
Frank Singhoff, UBO/LISyC, France
Frank.Singhoff@univ-brest.fr

Program Co-Chairs:
Yvon Kermarrec, Telecom Bretagne, France
Yvon.Kermarrec@telecom-bretagne.eu
Fabrice Kordon, Université Paris 6, France
Fabrice.Kordon@lip6.fr

Tutorial Chair:
Jérôme Hugues, Télécom Paris-Tech, France
Jerome.Hugues@telecom-paristech.fr

Exhibition Chair:
Pierre Dissaux, Ellidiss Technologies, France
Pierre.Dissaux@ellidiss.com

Publicity Chair:
Dirk Craeynest, Aubay Belgium and K.U.
Leuven, Belgium
Dirk.Craeynest@cs.kuleuven.be

Local Chairs:
Alain Plantec, UBO/LISyC, France
Alain.Plantec@univ-brest.fr
Mickael Kerboeuf, UBO/LISyC, France
Mickael.Kerboeuf@univ-brest.fr

The organizers are grateful to the exhibitors and sponsors of the conference (preliminary list)

Springer publishes the proceedings of the conference as volume 5570 in Lecture Notes in Computer Science