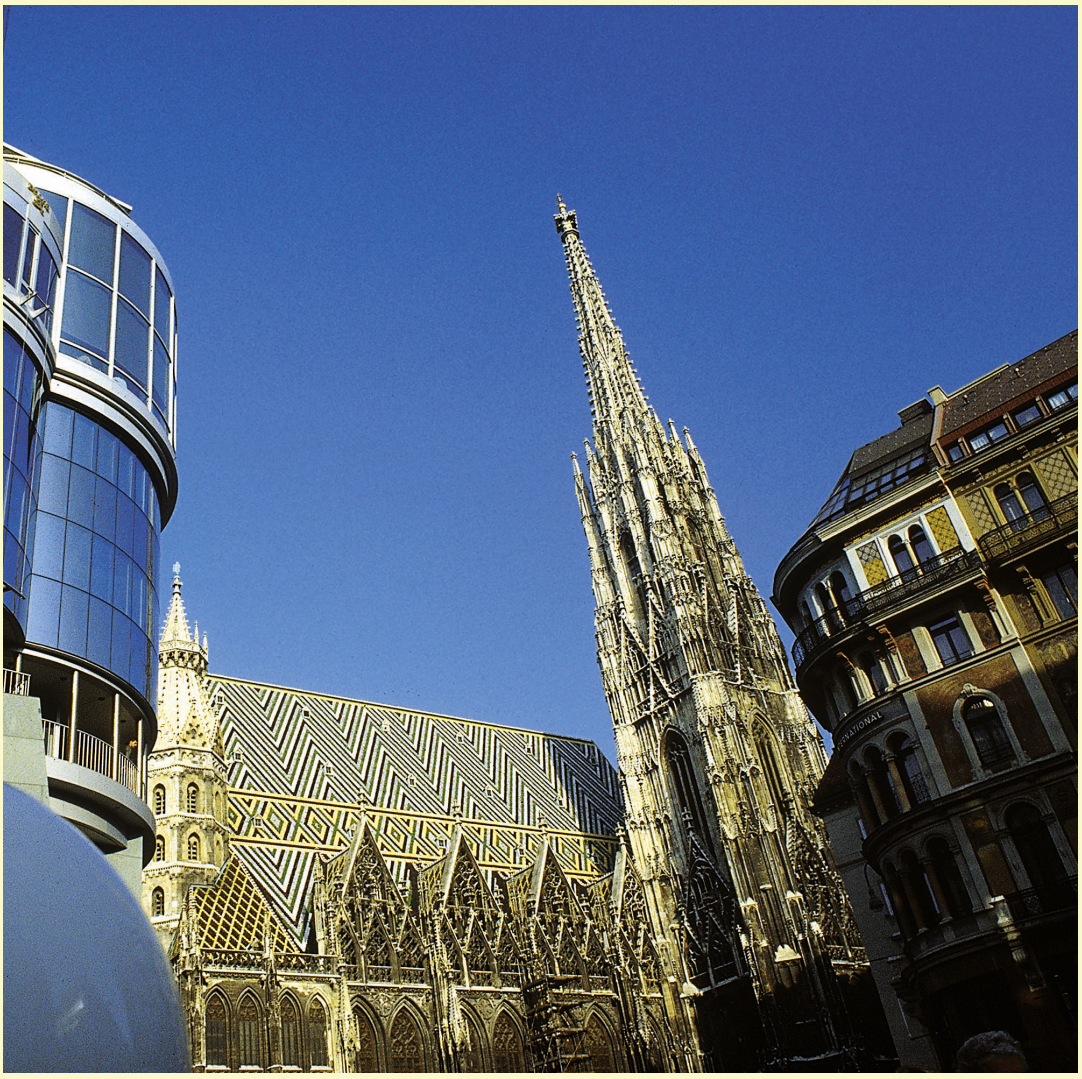
7th International Conference on
Reliable Software Technologies
Ada-Europe 2002



Vienna, Austria, June 17-21, 2002

ADVANCE PROGRAM

http://www.ada-europe.org/conference2002.html

SIGAda

Preliminary Program

The information presented here is preliminary - please refer to the conference web site for the latest details.

In 2002, the 7th International Conference on Reliable Software Technologies will take place in Vienna, Austria, from June 17th to June 21st. The conference offers a technical program and exhibition, plus a series of tutorials and a workshop.

The conference provides an international forum for researchers, developers and users of reliable software technologies. Presentations and discussions cover applied and theoretical work currently conducted to support the development and maintenance of software systems.

Vienna, a city with about 2 million inhabitants is situated in the heart of Europe. It is a city on which its ever-changing history has left an indelible mark, manifested also in the rich cultural heritage. Shaped by its hundreds of years as capital of an empire, the city's ultimate fascination nowadays stems from combining imperial grandeur with explosive modernity.

The conference will take place in the Parkhotel Schönbrunn which originated in 1907 as the guest house of Emperor Franz Josef I. The newly renovated hotel is located in the immediate vicinity of the "Schönbrunn Palace" and its beautiful surrounding park, situated close to the center of Vienna.

Overview of the Week

<table>
<tr><th></th><th>Morning</th><th>Late Morning</th><th>After Lunch</th><th>Afternoon</th></tr>
<tr><td rowspan="3">Monday
June 17th
Tutorials</td><td colspan="4">SPARK, an "Intensive Overview"
P. Amey and J. Barnes</td></tr>
<tr><td colspan="4">MaRTE OS: Bringing Embedded Systems and Real-Time POSIX Together
M. González Harbour and M. Aldea Rivas</td></tr>
<tr><td colspan="2">Principles of Physical Software Design in Ada95
M. Heaney</td><td colspan="2">Implementing Design Patterns in Ada95
M. Heaney</td></tr>
<tr><td rowspan="2">Tuesday
June 18th
Sessions &
Exhibition</td><td rowspan="2">Embedded Systems Unsuitable for Object Orientation
Maarten Boasson</td><td>Embedded Systems</td><td>Real-Time Systems</td><td rowspan="2">High-Integrity Systems</td></tr>
<tr><td>Case Studies</td><td>Vendor presentations</td></tr>
<tr><td rowspan="2">Wednesday
June 19th
Sessions &
Exhibition</td><td rowspan="2">On Architectural Stability and Evolution
Mehdi Jazayeri</td><td rowspan="2">Ada Language Issues</td><td>Program Analysis</td><td rowspan="2">Tools</td></tr>
<tr><td>Vendor presentations</td></tr>
<tr><td rowspan="2">Thursday
June 20th
Sessions &
Exhibition</td><td rowspan="2">Indulgent Algorithms: Virtues and Limitations
Rachid Guerraoui</td><td>Distributed Systems</td><td>Libraries</td><td rowspan="2">Contextware: Bridging Physical and Virtual Worlds
Alois Ferscha</td></tr>
<tr><td>Vendor presentations</td><td>Object-Orientation</td></tr>
<tr><td rowspan="3">Friday
June 21st
Tutorials &
Workshop</td><td colspan="4">CORBA 3 and CORBA for Embedded Systems
S. Ron Oliver</td></tr>
<tr><td colspan="2">Using Open Source Hardware and Software to Build Reliable Systems
J. Sherrill and J. Gaisler</td><td colspan="2">Cleanroom Software Engineering: An Overview
W. Bail</td></tr>
<tr><td colspan="2">Workshop: A Standard Container Library for Ada
E. Lamm</td><td colspan="2">Exceptions – What You Always Wanted to Know about Exceptions, But Were Afraid to Ask
C. Colket</td></tr>
</table>

Invited Speakers

Embedded Systems Unsuitable for Object Orientation

Maarten Boasson, University of Amsterdam, The Netherlands

Tuesday June 18th, 9:30

It will be argued that the current focus on object technology is detrimental to progress in embedded systems. The core of the problem is that OO is fine for analysis but does not answer the design needs. The difference between analyzing a system and designing one is enormous: during analysis, the focus is on understanding the various functional components of a system and their interactions, whereas designing is about implementing the desired functionality under the constraint that all quality goals for the system are achieved. The way that object technology forces the software to be structured establishes relationships between objects that do not necessarily reflect the way real-world entities are related. In the real world, the entities that have been modeled in the system very often have no hierarchical relationships, but have their own autonomous behavior. This results in awkward programming constructs necessary for overcoming the idiosyncrasies of the object paradigm. This talk will outline a different approach and will discuss benefits of that approach relative to object orientation.

Short Biography

Maarten Boasson studied mathematics in Groningen, The Netherlands. He became involved in advanced studies aiming at control of complexity, both of the development process and of the system under development itself. This resulted in the creation of a novel architecture for distributed reactive systems, that has been applied successfully in numerous systems and is, more than 10 years after its introduction, still unsurpassed in its support for integration, fault tolerance and component reuse. In 1996 Boasson was appointed professor of computer science at the University of Amsterdam. He has been involved in the organization of numerous international conferences, played a major role in establishing a dutch national research program in embedded systems, and is currently associate editor-in-chief of IEEE Software.

On Architectural Stability and Evolution

Mehdi Jazayeri, Technical University of Vienna, Austria

Wednesday June 19th, 9:30

Many organizations are now pursuing software architecture as a way to control their software development and evolution costs and challenges. A software architecture describes a system's structure and global properties and thus determines not only how the system should be constructed but also guides its evolution. An important challenge is to be able to evaluate the "goodness" of a proposed architecture. The talk will propose stability or resilience as a primary criterion for evaluating an architecture. The stability of an architecture is a measure of how well it accommodates the evolution of the system without requiring changes to the architecture. As opposed to traditional predictive approaches to architecture evaluation, this talk will suggest retrospective analysis for evaluating architectural stability by examining the amount of change applied in successive releases of a software product. A case study of twenty releases of a telecommunication software system containing a few million lines of code will be used to demonstrate how retrospective analysis may be performed. The talk will also present the challenges in software evolution and conclude with recommendations for future research. This work was part of the project "Architectural Reasoning for Embedded Systems" (ARES).

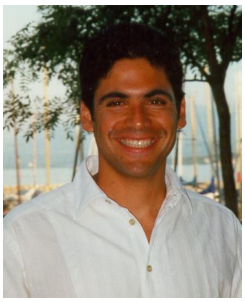
Short Biography

Mehdi Jazayeri is a professor of computer science at the Technical University of Vienna where he holds the chair of distributed systems. He spent many years in software research and development at several Silicon Valley companies, including ten years at Hewlett-Packard Laboratories in Palo Alto, California. His recent work has been concerned with component-base software engineering of distributed (web-based) systems. He is a coauthor of Programming Language Concepts (John Wiley, 1998), Fundamentals of Software Engineering (Prentice-Hall, 2002), and Software Architecture for Product Families (Addison-Wesley, 2000).

Indulgent Algorithms: Virtues and Limitations

Rachid Guerraoui, Swiss Federal Institute of Technology in Lausanne (EPFL)

Thursday June 20th, 9:30



An indulgent algorithm is a reliable distributed algorithm that does not need reliable failure detection. In particular, with an indulgent algorithm, no process ever needs to know if some other process is up or down. This talk will discuss some practical and theoretical ramifications of indulgent algorithms, and will give some new results on indulgent algorithms in the context of the consensus problem.

Short Biography

Rachid Guerraoui is professor in computer science at the Swiss Federal Institute of Technology in Lausanne where he leads the Distributed Programming Laboratory (lpdwww.epfl.ch). He is interested in classic and less classic music, triathlons, object-oriented programming and distributed algorithms.

Contextware: Bridging Physical and Virtual Worlds

Alois Ferscha, University of Linz, Austria

Thursday June 20th, 16:30

Today a variety of terms like Ubiquitous Computing, Pervasive Computing, Invisible Computing, Ambient Intelligence, Sentient Computing, and Post-PC Computing refers to new challenges and paradigms to the interaction between users and mobile and embedded computing devices. Fertilized by a vast quantitative growth of the Internet over the past years and a growing availability of wireless communication technologies, an ubiquitous use of "embedded" information society technologies is evolving. Most of the services delivered through those new technologies are services adapted to context, particularly to the person, time and location of their use. The aim for seamless service provision to anyone (personalized services) at any place (location based services) and at any time has brought the issues of middleware to a new discussion: it is expected that context-aware services will evolve, enabled by wirelessly networked computing devices. Services with explicit user input and output will be replaced by a computing landscape sensing the physical world via all different kinds of sensors, and controlling it via actuators in such a way that it becomes merged with the virtual world.

This talk will explore the software engineering issues, challenges and enabling technologies associated with the provision of context-aware services. In analogy to the term "middleware"—generally understood as software technologies that serve to mediate between two or more separate and usually already existing software components—the term "contextware" is introduced as the core of software technologies mediating services and the context of their use, thus bridging virtual and physical worlds.

Short Biography

Alois Ferscha received the Mag. degree in 1984, and a PhD in business informatics in 1990, both from the University of Vienna, Austria. From 1986

through 2000 he was with the Department of Applied Computer Science at the University of Vienna at the levels of assistant and associate professor. In 2000 he joined the University of Linz as full professor. He published more than 60 technical papers on topics related to parallel and distributed computing. Currently his research interests are in the areas of Pervasive Computing, Embedded Software Systems, Wireless Communication, Multiuser Cooperation, Distributed Interaction and Distributed Interactive Simulation.

Exhibition

The exhibition opens in the mid-morning break on Tuesday and runs until after the Thursday afternoon break. It takes place in the Parkhotel Schönbrunn's "Kaisersalon", where the coffee breaks are held.

The mid-morning and almost all mid-afternoon breaks are 60 minutes to give attendees ample opportunity to visit the exhibition.

Each exhibitor will have at least one half hour presentation slot during the vendor track; the program for the vendor presentations is still being worked out.

At the time of writing seven exhibitors: ACT, Aonix, Raincode, TNI, DDC-I, Green Hills Software Inc., and Rational have come forward, others have expressed interest.

Tutorials

SPARK, an "Intensive Overview"

Peter Amey & Janet Barnes, Praxis Critical Systems, UK

Monday June 17th, all day

SPARK is an annotated sub-language of Ada which is unambiguous and suitable for rigorous static analysis. The tutorial, which is extracted from the four-day "Software Engineering with SPARK" course will provide an intensive introduction to SPARK and the static analysis that is performed by the SPARK Examiner.

The tutorial is intended primarily for those with current or recent experience of software development in Ada, especially those who will work on or lead safety critical or other high integrity developments. Attendees will be encouraged to bring laptop computers on which the SPARK Examiner will be installed.

Outline

1. The rationale of SPARK: major design objectives, relationship with Ada, "why annotations?", tool support.
2. The core SPARK language: types, expressions, statements, subprograms, packages.
3. Data and information flow analysis (including practical).
4. Design building blocks: abstract state machines, abstract data types, interfacing with the environment.
5. SPARK and program design.
6. Formal verification.
7. Exception freedom.
8. Effective SPARK use, project organization.

Presenters

Peter Amey is an aeronautical engineer by original professional training and achieved Chartered

Engineer status through the Royal Aeronautical Society. He served as an engineering officer in the Royal Air Force and spent several years at the Boscombe Down test establishment working on the certification of aircraft armament systems. Peter joined Program Validation Limited to develop SPARK and the SPARK Examiner and continues that work today with Praxis Critical Systems. As well developing SPARK he has used it on major programs including Tornado, Eurofighter and the Lockheed C130J.

Peter teaches SPARK and Ada on a regular basis and has lectured widely on the development of critical systems. Most recently this has included the keynote address "Logic versus Magic" at Ada-Europe 2001 and the paper "A Language for Systems not Just Software" at SIGAda 2001.

Janet Barnes received a BA degree in Mathematics from Cambridge University in 1988 and MSc and Phil degrees in Computation from Oxford University in 1990 and 1993 respectively. For the last eight years Janet has specialised in design and implementation of high-integrity real-time embedded systems. As a user of SPARK on five development programmes she has extensive experience and understanding of the SPARK language.

Janet has taught the SPARK training course offered by Praxis Critical Systems on a number of occasions. Dr. Barnes is a member of the British Computer Society and is a Chartered Engineer.

MaRTE OS: Bringing Embedded Systems and Real-Time POSIX Together

Michael González Harbour & Mario Aldea Rivas, University of Cantabria, Spain

Monday June 17th, all day

MaRTE OS is a free software implementation of the POSIX minimum real-time system profile. It is designed for embedded systems and provides a development environment for Ada, C, or mixed language real-time applications. The tutorial will describe the main features of MaRTE OS, its architecture and performance, and the details on its development environment. An example will be used to demonstrate these concepts.

In addition, the tutorial will discuss the main real-time operating system services of the POSIX.13 minimum real-time profile. These services allow application developers to write portable applications that meet their real-time requirements, and that may be implemented on small embedded systems.

Outline Part I: Real-Time POSIX

1. Introduction. Real-Time operating systems
2. The POSIX standards and their Ada bindings
3. POSIX subsets: Application environment profiles
4. Thread management
5. Priority scheduling
6. Synchronization
7. Signals
8. Time management

Outline Part II: MaRTE OS

1. Introduction. MaRTE OS Design requirements
2. Architecture
3. Implementation details and performance
4. Development environment
5. Case study
6. Conclusions

Presenters

Michael González Harbour is a Professor in the Department of Electronics and Computers at the University of Cantabria. He works in software engineering for real-time systems. He is a co-author of *"A Practitioner's Handbook on Real-Time Analysis"*. He has been involved in several projects using Ada to build real-time controllers for robots. He is an active member of the POSIX real-time working group.

Mario Aldea Rivas received his Bachelor degree in Physics (Electronics) from the University of Cantabria (Spain). He is an Assistant Professor in the Department of Electronics and Computers at the University of Cantabria. His previous research activity was centered on the development of real-time embedded industrial robot controllers. He is currently finishing his PhD thesis on task scheduling in real-time operating systems, and as a part of that research activity he has developed MaRTE OS: an Ada real-time operating system for embedded applications that implements the POSIX.13 minimal real-time profile.

Principles Of Physical Software Design in Ada95

Matthew Heaney, On2 Technologies, US

Monday June 17th, morning

The tutorial addresses issues concerning the compilation of large software systems and presents many techniques for ameliorating the problems.

Most texts on software design concentrate almost exclusively on logical design, and provide only a cursory explanation of physical design. Discussions about types and objects are important, but there are also many pragmatic compilation issues that cannot be ignored. Unless care is taken, dependencies among modules often force a substantial recompile when seemingly innocuous changes are made. This can stymie development, especially for large systems that require hours (or even days) to rebuild.

Outline

- Physical versus logical design
- Coupling and cohesion
- What is the unit of decomposition?
- Decoupling of components
- Subsystems and package hierarchies
- Package idioms
- Type declarations - how, where, and how many
- Static versus dynamic polymorphism
- Aggregation versus inheritance
- Composition using access discriminants
- The adapter pattern: software glue
- Iterators as a decoupling mechanism
- Revisiting the Wegner taxonomy
- C++ friendship versus Ada95 child units
- How to emulate Java-style interfaces
- How to prevent unnecessary compiles
- Compiler support for minimal recompiles
- Hiding static values used to constrain a type or initialize an object
- Static versus dynamic linking
- COM: a binary standard for reusable components
- Private children versus subunits
- Factory functions
- Handle-body idiom and deferred type definition
- Using class-wide types to remove compilation dependencies
- How to break mutual dependency between specs

- Categorization pragmas
- Package elaboration issues
- Access types versus the Rosen Trick
- The nature of a subsystem interface
- The facade pattern
- Degrees of type safety
- The pitfalls of a Common_Types package
- How much should be declared in a generic package
- Where does the data go: module versus instance state
- Logical versus physical types
- What "maintainability" really means

Presenter

Please find Matthew Heaney's biography at the end of the description of the following tutorial also presented by him.

Implementing Design Patterns in Ada95

Matthew Heaney, On2 Technologies, US

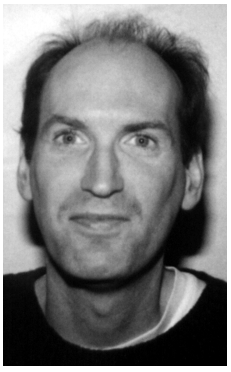
Monday, June 17th, afternoon

This tutorial addresses the question of what "design patterns" are and presents many advanced idioms for object-oriented programming in Ada95.

Outline

- Idioms for class-wide programming and memory management
- Interpreter pattern
- Flyweight pattern
- Template method (class-wide operations)
- Adding controlled-ness to a (tagged) type already in a class
- Auto pointers and smart pointers: to eliminate memory leaks
- Factory function: like a constructor in C++
- Factory method: factory function that dispatches
- Observer pattern: getting notified when the state of another object changes
- Multiple views idiom: for emulating Java-style interfaces
- Rosen Trick: to convert a constant view of an object into a variable view
- Parameter passing and tests for equality
- Decorator pattern: for adding behavior to an abstraction dynamically
- Dispatching through a generic formal subprogram or an access-to-subprogram object

Presenter



Matthew Heaney has been using Ada since 1987 to successfully build large, real-time systems in the areas of simulation and electronic intelligence. His interest in software design and object-oriented programming led to his work on design patterns. He learned about design patterns by converting all the C++ examples in the Gamma book to Ada95, and has since used them on real projects. Matthew Heaney has already given several tutorials at SIGAda, Ada-Europe, and AdaUK.

CORBA 3 and CORBA for Embedded Systems

S. Ron Oliver, Top Graph'X, US

Friday, June 21st, all day

This tutorial starts with an overview of CORBA 3 with emphasis on changes from CORBA 2. It will include a brief introduction to distributed computing, in general, including fundamentals of concurrent and real-time systems, and of computer networks. Thereafter it addresses CORBA principles, the Interface Definition Language (IDL), client programs, object (server) programs, CORBA Services, CORBA Facilities, and the CORBA Component Model (CCM). Several advanced features of CORBA 3, including Minimum CORBA and Real-Time CORBA, are also discussed. These topics are of particular interest when using CORBA in the area of embedded systems.

All examples for the tutorial will be based on the TopGraphX product, ORBAda, and the Ada95 programming language. Attendees need not have prior knowledge of CORBA.

Presenter

S. Ron Oliver has been in the software industry since 1968, with an equal mixture of industrial and academic experience. Since 1974 he has specialized in concurrent and real-time software engineering. During 9 years with the computer science and computer engineering programs at Cal Poly, he led many research and student projects in object-oriented design and implementation for concurrent and real-time systems, and in the distributed computing environment. Most recently he has been focusing on CORBA technology.

Conference Schedule

	Tuesday 18th		Wednesday 19th	Thursday 20th	
9:30–10:30	Invited Talk: **Embedded Systems Unsuitable for Object Orientation** *Maarten Boasson, University of Amsterdam, The Netherlands*		Invited Talk: **On Architectural Stability and Evolution** *Mehdi Jazayeri, Technical University of Vienna, Austria*	Invited Talk: **Indulgent Algorithms: Virtues and Limitations** *Rachid Guerraoui, Swiss Federal Institute of Technology, Lausanne, Switzerland*	
10:30–11:30	**Exhibition Opening & Coffee**		**Exhibition & Coffee**	**Exhibition & Coffee**	
	Embedded Systems	**Case Studies**	**Ada Language Issues**	**Distributed Systems**	**Vendor Session I**
11:30–12:00	Evaluating Performance and Power of Object-oriented vs. Procedural Programming in Embedded Processors *A. Chatzigeorgiou, G. Stephanides*	Software Development Reengineering – An Experience Report *Adrian Hoe*	How to use GNAT to Efficiently Preprocess New Ada Sentences *J. Miranda, F. Guerra, E. Martel, J. Martín, A. González*	Modelling & Schedulability Analysis of Hard Real-Time Distributed Systems based on Ada Components *J. L. Medina, J. Javier Gutiérrez, J. M. Drake, M. González Harbour*	11:30–12:00
12:00–12:30	OMC-INTEGRAL Memory Management *Jose Manuel Pérez Lobato, Eva Martín Lobo*	Development of a Control System for Teleoperated Robots using UML and Ada95 *F. J. Ortiz, A. Martínez, B. Alvarez, A. Iborra, J.M. Fernández*	Exposing Uninitialized Variables: Strengthening and Extending Run-Time Checks in Ada *Robert Dewar, Olivier Hainque, Dirk Craeynest, Philippe Waroquiers*	Transparent Environment for Replicated Ravenscar Applications *Luís Miguel Pinho, Francisco Vasques*	12:00–12:30
12:30–13:00	Language Issues of Compiling Ada to Hardware *M. Ward, N. C. Audsley*	Using a Secure Java Micro-kernel on Embedded Devices for the Reliable Execution of Dynamically Uploaded Applications *W. Binder, B. Lichtl*	Adding Design By Contract to the Ada Language *Ehud Lamm*	Concurrency Control in Transactional Drago *M. Patiño-Martínez, R. Jiménez-Peris, J. Kienzle, S. Arévalo*	12:30–13:00
13:00–14:30	**Lunch & Exhibition**		**Lunch & Exhibition**	**Lunch & Exhibition**	

Conference Schedule (cont.)

	Tuesday 18th		Wednesday 19th		Thursday 20th	
	Real-Time Systems	**Vendor Session II**	**Program Analysis**	**Vendor Session III**	**Libraries, APIs, and Bindings**	**Object-Orientation**
14:30–15:00	A POSIX-Ada Interface for Application-Defined Scheduling *Mario Aldea Rivas, Michael González Harbour*	14:30–15:00	Static Dependency Analysis for Concurrent Ada 95 Programs *Zhenqiang Chen, Baowen Xu, Jianjun Zhao, Hongji Yang*	14:30–15:00	An Ada Binding to the IEEE 1003.1q (POSIX Tracing) Standard *Agustín Espinosa Minguet, Ana García Fornes, Alfons Crespo i Lorente*	Using Object Orientation in High Integrity Applications: A Case Study *A. Alonso, R. López, T. Vardanega, J. A. de la Puente*
15:00–15:30	The Formal Development of a Real Time Kernel: Kernel Modelling *Stephen G. Michell, Douglas J. Howe*	15:00–15:30	DataFAN: A Practical Approach to Data Flow Analysis for Ada 95 *K. Czarnecki, M. Himsolt, E. Richter, F. Vieweg, A. Rosskopf*	15:00–15:30	GNAT Ada Database Development Environment *Michael Erdmann*	Ada, Interfaces and the Listener Paradigm *J-P. Rosen*
15:30–16:00	**Coffee & Exhibition**		Prioritization of Test Cases in MUMCUT Test Sets: An Empirical Study *Y. T. Yu, M. F. Lau*	15:30–16:00	**Coffee & Exhibition**	
16:00–16:30			**Coffee & Exhibition**			
	High-Integrity Systems		**Tools**			
16:30–17:00	Closing the Loop: The Influence of Code Analysis on Design *Peter Amey*		About the Difficulties of Building a Pretty-Printer for Ada *Sergey Rybin, Alfred Strohmeier*		Invited Talk: **Contextware: Bridging Physical and Virtual Worlds** *Alois Ferscha, Universität Linz, Austria*	
17:00–17:30	High-Integrity Systems Development for Integrated Modular Avionics using VxWorks and GNAT *Paul Parkinson, Franco Gasperoni*		A Tailorable Distributed Programming Environment *E. Martel, F. Guerra, J. Miranda*			
17:30–18:15	Ada-Europe General Assembly				Closing Session Awards: best paper & best presentation	
	Guided City Tour & Town-Hall Reception		**Banquet (Heuriger)**		**Schönbrunn Palace "Grand Tour"**	

TUTORIALS (cont.)

Using Open Source Hard- and Software to Build Reliable Systems

Joel Sherrill, OAR Corporation, US, & Jiri Gaisler, Gaisler Research, Sweden

Friday, June 21st, morning

A framework for the development of embedded systems based solely on open-source components is presented. The framework is based on the LEON SPARC-V8 processor, RTEMS real-time operating system, and the GNU Ada toolchain. The tutorial includes a discussion of the implications of applying the open source model to hardware and embedded systems software. An overview of the characteristics of real-time embedded systems, the cross development process, and the features of Ada95 that aid the development of real-time embedded systems is presented. A demonstration is made on how to configure the target processor, adapt the RTEMS operating system to custom boards, and develop Ada applications.

Outline

The LEON processor and the RTEMS real-time operating system are open source solutions targeting the two sides of embedded systems—hardware and software. Both were initially developed by organizations responsible for the development, deployment, and maintenance of custom systems with high safety and reliability requirements and extremely long lifespans. While the LEON was initially developed for space applications and RTEMS for military applications, both have been made available to the general embedded systems community. Interestingly, both have found use in a variety of domains not envisioned by their original sponsors.

In this tutorial, LEON and RTEMS and the implications for using open source hardware and software in embedded systems are examined. The features of both LEON and RTEMS along with the open source tools that support them are presented. This is followed by an examination of the adaptation of the GNAT compiler to this target environment and the features of the Ada programming language that are particularly useful to real-time embedded systems developers. The tutorial concludes with a demonstration of the GNAT/RTEMS-LEON toolchain on both a simulator and real hardware.

Presenters

Dr. Joel Sherrill is Director of Research and Development for OAR Corporation with 15 years experience in the design, development, and fielding of real-time embedded applications in a variety of military, commercial, and research domains. As a principal author and current maintainer of the open-source real-time operating system RTEMS, he has

been deeply involved in numerous RTEMS related efforts including the GNAT/RTEMS validation. In addition to regularly teaching courses on RTEMS and other real-time topics for OAR Corporation, Dr. Sherrill is a part-time instructor at the University of Alabama in Huntsville. He is a founding member of the Steering Committee for the Free Software Foundation's GNU Compiler Collection. As an experienced software developer, Dr. Sherrill has also focused on the practical application of software engineering research to the everyday life of the developer. In this light, he is a member of OAR's SEPG and has led process improvement and implementation efforts on a number of projects.

Jiri Gaisler is the founder of Gaisler Research and

the designer of the ERC32 and LEON processors. He has 15 years experience in the design and implementation of fault-tolerant processors and digital systems for space applications, most of it gained as staff researcher at the European Space Agency. Current activities involve simulator design and data-handling systems development for future space platforms. Mr. Gaisler has presented several papers at international conferences, and regularly holds courses on LEON architecture and utilization.

Cleanroom Software Engineering: An Overview

William Bail, MITRE & University of Maryland, US

Friday, June 21st, afternoon

Cleanroom Software Engineering is an approach to the development of software that is strongly rooted in formal methods and mathematics. Developed by Harlan Mills and his colleagues at IBM in the late 1970s, and officially named Cleanroom in 1987, this technique emphasizes defect avoidance. While not gaining the notoriety that other techniques have en-

joyed, projects that have applied Cleanroom have experienced significant benefits, including low defect rates. Cleanroom emphasizes multiple builds in an incremental model, with each build constructed using forms known as box structures. Verification of the structures is accomplished using correctness proofs, while software certification is based on usage models which facilitate statistical testing. Recent work has integrated Cleanroom with object-oriented models. In addition the SEI has released a Cleanroom Software Engineering Reference Model, providing an integrated set of work products and processes for organizations wishing to apply this technique.

Presenter

William Bail

BS, Carnegie Institute of Technology, Mathematics MS, University of Maryland, Computer Science PhD, University of Maryland, Computer Science

Work experience focuses on aiding large scale software development projects in the areas of software methodologies, principally for the US Government (DoD and FAA). Teaching focuses on requirements engineering, IV&V practice, and software design. William Bail has a total of 35 years of software engineering experience.

Exceptions — What You Always Wanted to Know about Exceptions, But Were Afraid To Ask

Currie Colket, MITRE & ACM SIGAda, US

Friday, June 21st, afternoon

Exception processing has the power to detect serious problems in the execution of a program and return one back to a known safe state with high integrity. As such, it can be a very powerful tool for developing high quality software. Unfortunately many developers do not use the full power of exceptions. Frequently the use of exceptions is to simply log the problem and continue execution, allowing things to gracefully degrade. In the case of Ariane 5, exceptions were raised appropriately, but the result had not been well thought out, resulting in a disaster.

This tutorial will start at the basics, discussing the Ada 83 concept of exceptions. To be effective, exceptions and their handling must be addressed at the design level and not at the code level where it is frequently performed today. This presentation will discuss several alternative approaches to addressing error handling in the design using exceptions. Ada 95 introduced some important changes to the exception area making them more effective. In particular, the addition of package Ada.Exceptions provides excellent facilities to support debugging and provides a mechanism to eliminate erroneous mapping of raised exceptions.

The use of exceptions can be assessed via automated tools. Several analyses that can be performed on a program via automated tools so the program quality can be improved will be discussed. The tutorial will conclude by addressing proposed needs for exceptions resulting from the Exception Workshop held at Ada-Europe 2001.

Outline

1. Introduction
2. Major Benefits of Exceptions
3. Ada 95 RM Exception View & Ada 83 deltas
4. Designing Software With Exceptions
5. Design Considerations
6. Ada 95 Quality and Style Guide
7. Design Examples
8. Issues Using Exceptions—Exception Gotchas
9. Future of Ada Exceptions
10. Conclusion

Presenter

Mr. Currie Colket is the Chair of ACM SIGAda, the Chair of the SIGAda Ada Semantic Working Group, and the Chair of the ISO WG9 ASIS Rapporteur Group. He recently retired from the DoD where he served in the Air Force as an Airborne Surveillance Officer on AWACS and a computer scientist for the United States Navy. Mr. Colket is currently a software systems engineer for MITRE. His current tasks involve code analysis using ASIS-based tools. In this capacity, he has addressed the use of Ada exceptions for both code development and operational use. Prior to his affiliation with MITRE, he was a consultant for the Software Program Manager's Network (SPMN). He has a Bachelor of Science from Case Institute of Technology, a Master of Business Administration from the University of Southern Mississippi and a Master of Science in Computer Science from the Ohio State University.

Workshop

A Standard Container Library for Ada

Friday, June 21st, morning. Both contemporary dominant general purpose programming languages, Java and C++, come equipped with a standard set of reusable containers, such as Maps and Sets. There are quite a few Ada libraries for these purposes, but there is little agreement on the exact details of a standard container library. There is however a general feeling, as can be witnessed on recent discussions on comp.lang.ada, that such a library is important for Ada's future. A standard container library is important for achieving many of Ada's goals, top among them the use of reusable components for efficient software engineering. Other important goals that can be served by a standard container library are educational uses and efficient implementation of common algorithms and data structures, which is important for real-time systems.

Designing a useful standard container library for Ada is a difficult task, as the language is used in a wide variety of different domains, with different and at times conflicting demands. Hence the need for debating and elaborating the issues among a group of interested Ada users.

Aims of the Workshop

Workshop participants should attempt to come up with two deliverables: a set of core requirements that are deemed by the majority to be critical for applicability of the standard container library to their needs, and a high level design for the library itself.

The workshop should help in forming a working group dedicated to implementing a reference implementation of the proposed library.

The results of the workshop should furthermore form the basis for a recommendation which would lead to the adoption of a standard container library as part of the Ada standard library, in the next revision of the Ada language.

Workshop Co-Chairs

Ehud Lamm (ehudla@openu.ac.il), The Open University of Israel

John English (je@brighton.ac.uk), University of Brighton

Participation

Participation to the workshop is limited to 25–35 individuals and is by invitation upon acceptance of a submission. Participants should submit brief position papers (1 or 2 pages) to Ehud Lamm, including link to relevant source code if at all possible. The workshop will include talks based on the submitted papers and intensive shepherded design sessions.

The deadline for the submission of position papers is stated on the conference web site.

See the conference web site for more details (http://www.ada-europe.org/conference2002.html).

Social Program

Tuesday Evening: Civic Reception

On Tuesday we will enjoy a cocktail reception at the historic town hall by invitation of the Mayor of Vienna. Before that a guided tour by bus will provide a first impression of the city and several of its well-known sights, among them the magnificent buildings along the Ringstraße, such as the State Opera House, the Museum of Fine Arts, the Museum of Natural History, the Hofburg Palace, the Parliament, the City Hall, the Burgtheater, and the University as well as the Votivkirche Church.

Wednesday Evening: Banquet

Wednesday evening the conference banquet will take place at a famous "Heurigen" in Grinzing. Heuriger is a word that the Viennese use both for the wine of the latest grape harvest and for the establishments at which it is served. Grinzing is probably Austria's best known wine-growing district, and the Heuriger "Altes Presshaus" has a tradition that dates back as far as 1527. Over a glass of wine and traditional Viennese cuisine we will have the opportunity to experience several of the mundane ingredients such as "Schrammel-Musik" and "Wiener Gemütlichkeit" that add to the flair of this city.

Thursday Evening: Schönbrunn Palace "Grand Tour"

A short walk from the conference venue "Parkhotel Schönbrunn" through the adjacent baroque-style park of the Schönbrunn Palace will take us to the palace itself, where we will enjoy an exclusive evening tour through this former summer residence of the Habsburg family.

7th International Conference on Reliable Software Technologies - Ada-Europe 2002
Vienna, Austria, June 17-21, 2002

REGISTRATION FORM

PARTICIPANT *Please use block capitals*

Ms/Mrs [ ] Mr [ ] Title _____

Family name _____ First name _____

Affiliation/Organization _____

Street _____

City _____ Post/Zip code _____ Country _____

Telephone _____ Fax _____ Email _____

Special requirements (e.g. diet) _____

Reduced registration fee

[ ] member Ada-Europe; national organization _____ [ ] academia

[ ] member ACM; membership number _____

Additional Comments _____

Registration time Early registration (by May 25th) [ ] Late or on site (after May 25th) [ ]

REGISTRATION FEES

Conference registration fee (see table on next page)

Three day conference ..... EUR _____

Individual days (Tue [ ] Wed [ ] Thu [ ]) ..... EUR _____

Tutorial/Workshop registration (see table on next page)

Please indicate tutorials/workshop for which you want to register:

Monday, June 17th [ ] T1 [ ] T2 [ ] T3 [ ] T4

Friday, June 21st [ ] T5 [ ] T6 [ ] T7 [ ] T8 [ ] W

Tutorial/Workshop registration fee ..... EUR _____

Extra Banquet ticket: _____ tickets @ 53 EUR ..... EUR _____

Extra proceedings: _____ proceedings @ 30 EUR ..... EUR _____

TOTAL PAYMENT DUE ..... EUR _____

PAYMENT METHOD

By bank transfer [ ] **By cheque** [ ] **By credit card** [ ]

By bank transfer to account number 0130-30655/00, "TU Wien – Ada-Europe 2002". The account is at the CA-BV Austria whose bank identifier (swift) code is CABVATWW (Please mention "Ada-Europe 2002" and your name and attach proof of payment, e.g., a copy of the bank draft, to this form).

By cheque drawn on an Austrian Bank and made payable to:

TU Wien Reliable Software Technologies—Ada-Europe 2002.

By credit card MasterCard [ ] Visa [ ]

Card _____ Expiration Date _____

Name as shown on credit card _____ Signature:

Mail or fax this form to:

AE2002 Registration, CON.ECT, Event Management GesmbH,
Kaiserstr. 14, A-1070 Vienna, Austria
Fax ++43 1 522 36 36-10

Document Version 1.4

Conference Registration Fee:

Three days of conference (June 18th–June 20th) including one copy of the proceedings, coffee breaks, lunches, visit and reception in town hall on Tuesday 18th, and the Schönbrunn Palace "Grand Tour" on Thursday 20th.

	member Ada-Europe or ACM SIGAda		non member	
	non academia	academia	non academia	academia
Early registration (by May 25th)	530 EUR	470 EUR	590 EUR	530 EUR
Late registration (after May 25th)	590 EUR	590 EUR	650 EUR	650 EUR
Individual day registration (per day)	270 EUR	270 EUR	300 EUR	300 EUR

Tutorial Registration Fee:

Prices are per tutorial, including tutorial notes and coffee breaks.
Lunches are only included when registered for full day tutorial or two half day tutorials on the same day.

	half day	full day or two halves on same day	Workshop (by invitation only)
Early registration (by May 25th)	120 EUR	230 EUR	50 EUR
Late registration (after May 25th)	150 EUR	290 EUR	70 EUR

Overview of Tutorials:

Monday June 17th	**T 1**	full day	SPARK, an "Intensive overview" – *Amey/Barnes*
	T 2	full day	MaRTE OS: Bringing Embedded Systems and RT POSIX Together – *Gonzalez/Aldea*
	T 3	morning	Principles of Physical Software Design in Ada 95 – *Heaney*
	T 4	afternoon	Implementing Design Patterns in Ada 95 – *Heaney*
Friday June 21st	**T 5**	full day	CORBA 3 and CORBA for Embedded Systems – *Oliver*
	T 6	morning	Using Open Source Hardware and Software to Build Reliable Systems – *Sherrill/Gaisler*
	T 7	afternoon	Cleanroom Software Engineering: An Overview – *Bail*
	W	morning	Workshop: Standard Container Library for Ada – *Lamm* (by invitation only)
	T 8	afternoon	Exceptions – What You Always Wanted to Know About Exceptions, But Were Afraid to Ask – *Colket*

Note: No registration request will be confirmed until payment has been received. CANCELLATIONS must be in writing. A Cancellation fee of 120 EUR will be applied to all cancellations. No refunds will be given for cancellations postmarked after June 1st. Substitutions will be accepted. The hotel information can be found through the web page of the conference. Additional lunch tickets will be on sale throughout the conference.

For latest information see the web page at `http://www.ada-europe.org/conference2002.html`, or send email to ae2002-info@auto.tuwien.ac.at.

For any information, please contact:

Bettina Hainschink (Conference Secretariat), CON.ECT
CON.ECT, Event Management GesmbH
Kaiserstr. 14, A-1070 Vienna, Austria

Tel: ++43 1 522 36 36
Fax: ++43 1 522 36 36-10
Email: events@conect.at

Program Committee

Ángel Álvarez, Technical University of Madrid, Spain

Lars Asplund, Uppsala University, Sweden

Neil Audsley, University of York, UK

John Barnes, UK

Guillem Bernat, University of York, UK

Maarten Boasson, University of Amsterdam, The Netherlands

Ben Brosgol, ACT, USA

Bernd Burgstaller, TU Vienna, Austria

Ulf Cederling, Vaxjo University, Sweden

Roderick Chapman, Praxis Critical Systems Limited, UK

Paolo Coppola, INTECS HRT, Italy

Dirk Craeynest, Offis nv/sa & K.U.Leuven, Belgium

Alfons Crespo, Universidad Politécnica de Valencia, Spain

Peter Dencker, Aonix GmbH, Germany

Raymond Devillers, Université Libre de Bruxelles, Belgium

Brian Dobbing, Praxis Critical Systems Limited, UK

Wolfgang Gellerich, IBM, Germany

Jesús M. González-Barahona, ESCET, Universidad Rey Juan Carlos, Spain

Michael González Harbour, Universidad de Cantabria, Spain

Thomas Gruber, ARC Seibersdorf research, Austria

Helge Hagenauer, University of Salzburg, Austria

Andrew Hately, Eurocontrol, Belgium

Günter Hommel, TU Berlin, Germany

Wolfgang Kastner, TU Vienna, Austria

Jan van Katwijk, Delft University of Technology, The Netherlands

Hubert B. Keller, Forschungszentrum Karlsruhe, Germany

Yvon Kermarrec, ENST Bretagne, France

Jörg Kienzle, Swiss Federal Institute of Technology Lausanne, Switzerland

Albert Llamosí, Universitat de les Illes Balears, Spain

Kristina Lundqvist, Massachusetts Institute of Technology, USA

Franco Mazzanti, Istituto di Elaborazione della Informazione, Italy

John W. McCormick, University of Northern Iowa, USA

Pierre Morere, Aonix, France

Laurent Pautet, ENST Paris, France

Erhard Plödereder, University Stuttgart, Germany

Juan A. de la Puente, Universidad Politécnica de Madrid, Spain

Gerhard Rabe, TÜV Nord e.V., Hamburg, Germany

Jean-Marie Rigaud, Université Paul Sabatier, Toulouse, France

Alexander Romanovsky, University of Newcastle, UK

Jean-Pierre Rosen, Adalog, France

Bo Sanden, Colorado Technical University, USA

Bernhard Scholz, TU Vienna, Austria

Edmond Schonberg, New York University & ACT, USA

Tullio Vardanega, Dept. of Pure and Applied Math., Univ. of Padova, Italy

Stef Van Vlierberghe, Eurocontrol CFMU, Belgium

Andy Wellings, University of York, UK

Ian Wild, Eurocontrol CFMU, Belgium

Jürgen Winkler, Friedrich-Schiller-Universität, Jena, Germany

Thomas Wolf, Paranor AG, Switzerland

Further Information

The conference web site gives up-to-date details of the program. Also on the web site are details on the venue including travel information and *hotel accommodation*. In case of unavailable web-access please contact the *Conference Secretariat* regarding hotel accommodation and to receive further information.

Bettina Hainschink
Ada-Europe 2002 Conference Secretary
CON.ECT Event Management GesmbH
Kaiserstr. 14, A-1070 Vienna, Austria
Tel: ++43 1 522 36 36 Fax: ++43 1 522 36 36-10
Email: events@conect.at

http://www.ada-europe.org/conference2002.html

Organization

Conference Chair

Gerhard H. Schildt
Technical University of Vienna
Department of Computer-Aided Automation
Schildt@auto.tuwien.ac.at

Program Co-Chairs

Johann Blieberger
Technical University of Vienna
Department of Computer-Aided Automation
Blieberger@auto.tuwien.ac.at

Alfred Strohmeier
Swiss Fed. Inst. of Technology Lausanne
Software Engineering Lab
Alfred.Strohmeier@epfl.ch

Tutorial Chair

Helge Hagenauer
University of Salzburg
Dept. Comp. Science & System Analysis
hagenau@cosy.sbg.ac.at

Exhibition Chair

Thomas Gruber
ARC Seibersdorf research GmbH
thomas.gruber@arcs.ac.at

Publicity Chair

Dirk Craeynest
Offis nv/sa & K.U.Leuven
Dirk.Craeynest@cs.kuleuven.ac.be

Local Organization Chair

Bernd Burgstaller
Technical University of Vienna
Department of Computer-Aided Automation
Burgstaller@auto.tuwien.ac.at

In cooperation with

SIGAda

The organizers thank the exhibitors (preliminary list)

and the supporters (preliminary list) **of the conference.**

The City of Vienna