Lecture Notes in Computer Science 4147

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Manfred Broy Ingolf H. Krüger
Michael Meisinger (Eds.)

Automotive Software – Connected Services in Mobile Networks

First Automotive Software Workshop, ASWSD 2004
San Diego, CA, USA, January 10-12, 2004
Revised Selected Papers

Springer

Volume Editors

Manfred Broy
Institut für Informatik
Technische Universität München
Boltzmannstr. 3
D-85748 Garching, Germany
E-mail: broy@informatik.tu-muenchen.de

Ingolf H. Krüger
University of California, San Diego
Computer Science and Engineering
9500 Gilman Drive
La Jolla, CA 92093-0404, USA
E-mail: ikrueger@cs.ucsd.edu

Michael Meisinger
Institut für Informatik
Technische Universität München
Boltzmannstr. 3
D-85748 Garching, Germany
E-mail: meisinge@informatik.tu-muenchen.de

Library of Congress Control Number: 2006932846

CR Subject Classification (1998): C.2.4, C.3, C.4, C.5.3, D.1.3, D.2.1, D.2.2, D.2.3, D.2.4, D.2.7, D.2.11, D.2.12, D.2.13, D.3.1, D.4, H.3-5, J.7

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-540-37677-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-37677-4 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11823063 06/3142 5 4 3 2 1 0

Preface

Software development for the automotive domain is currently subject to a silent revolution. On the one hand, software has become the enabling technology for almost all safety-critical and comfort functions offered to the customer. A total of 90 % of all innovations in automotive systems are directly or indirectly enabled by software. Today's luxury cars contain up to 80 electronic control units (ECUs) and 5 different, inter-connected network platforms, over which some 700 software-enabled functions are distributed.

On the other hand, the complexity induced by this large number of functions, their interactions, and their supporting infrastructure has started to become *the* limiting factor for automotive software development. Adequate management of this complexity is particularly important; the following list highlights three of the corresponding challenges:

First, the dependencies between safety-critical and comfort functions are rapidly increasing; a simple example is the interplay of airbag control and power seat control in the case of an accident. Careful analysis and design of these dependencies are necessary to yield correct software solutions.

Second, advances in wired and wireless networking infrastructures enable interconnection between cars and backend service providers (e.g., to call for help in cases of emergency), between cars and devices brought into the car by drivers and passengers (such as cell phones, PDAs, and laptops), and even among cars. This dramatically shifts the focus from the development of individual software solutions residing on dedicated ECUs to their distribution and interaction within and beyond car boundaries.

Third, the myriad of functions and services offered to the driver and passengers need to be effectively accessible without compromising traffic safety. This requires user interfaces addressing not only ease of use but also priority of information necessary for safe vehicle operation, and choice of interface modality (e.g., voice versus pushing of buttons for menu selection) for reasons of adequacy or user limitations.

These challenges are aggravated by demanding time-to-market requirements, short development cycles, rapid change of technological infrastructures, customer demands, and product lines. The silent revolution currently underway in the automotive domain thus consists of a shift of focus from hardware to software infrastructures and from ECUs to software services as the center of concern in the development process. This puts the *software architecture* for future generation automotive systems in the spotlight as a critical element both for enabling advanced services supporting drivers and passengers, and for managing the complexity of these functions amidst the high safety demands they are subject to.

The goal for the first Automotive Software Workshop, San Diego, ASWSD 2004, was to bring together experts from industry and academia, who work on highly

complex, distributed, reactive software systems related to the automotive domain, and to discuss and further the understanding of the following focus areas:

- Automotive Software and Software Architectures
- Automotive Domain Architectures
- Automotive Software Services
- Automotive Hardware, Middleware, and Software Platforms
- On- and Off-Board Ad-Hoc Networking
- Networked Automotive Services
- Mobile Sensor Networks
- Reliability, Security and Privacy for Automotive Software
- Enabling Technologies for Telematics Applications

The workshop, which took place during January 10–12, 2004 in La Jolla, CA, USA, contributed to fostering a deeper understanding of the research challenges and agendas in this area. Potentials for cross-disciplinary research, as well as pertinent curricula and training programs to address these challenges were identified and discussed.

The workshop program consisted of 4 keynote presentations, 22 technical paper presentations and 2 panel discussions. The workshop spanned two-and-a-half days and was divided into the following topical sessions: Quality Assurance (QA), Networking Infrastructures and Applications (NI), Real-Time Control (RT), Services and Components (SC), and Model-Based Development and Tools (MD). The pre-proceedings, consisting of the presentation slide sets, were made available at *http://aswsd.ucsd.edu/2004*.

To foster discussion on cross-cutting and interdisciplinary topics the organizers decided to have four keynote presentations (two from industry and two from academia) and two panel discussions as integral parts of the workshop program. Hans-Georg Frischkorn (then BMW Group), Hermann Kopetz (TU Vienna), K. Venkatesh Prasad (Ford Motor Company), and Janos Sztipanovits (Vanderbilt University) were recruited as keynote speakers. Professor Larry Smarr (Director, Calit2) delivered opening remarks on the first day of the workshop.

Hans-Georg Frischkorn (then Senior Vice President System Architecture and System Integration, BMW Group) delivered the opening keynote "Automotive Software – The Silent Revolution" in the Quality Assurance session, where he laid out BMW's automotive software vision to realize innovative software-based functionality in cars. He stressed the importance of software architectures and infrastructures, and promoted an open software platform providing extensibility, updatability and support for easy integration of new functionality. Frischkorn showed how the tasks of operating and maintaining future vehicle generations could be supported by system services provided on multiple layers of abstraction.

The keynote presentation for the session on Real-Time Control was given by *Hermann Kopetz* (Technical University of Vienna); he stressed the importance of having a fault-hypothesis for safety-critical real-time systems. Kopetz concluded that (a) the design of safety-critical computing systems requires a fault-tolerant architecture and a rigorous design methodology, (b) the precise specification of

the fault hypothesis is the key document in the design of fault-tolerant systems, and (c) the architecture of a safety critical application must tolerate the arbitrary failure of any single VLSI chip since one cannot assume that a chip contains two independent fault containment regions.

The Services and Components session started with the keynote by *K. Venkatesh Prasad* (Leader Ford Motor Company's Infotronics Technologies Group). He sketched the future of automotive product creation, involving the rapid convergence of enterprise and embedded computing and of portable-mobile, fixed-mobile and fixed wireless communications. Prasad argued that creating an automobile clearly calls for a series of innovations that in turn rely on a body of inventions, literature and competencies that are created and nurtured in academia and the broad industrial and public sector research & development base. This highlighted the emerging role of software technologies and processes in modern automotive product creation, and stimulated thinking in terms of how academic curricula might need to evolve and what types of new collaboration styles might be needed for the creation of sustainable mobility solutions in the future.

Janos Sztipanovits (Vanderbilt Univeristy) delivered the keynote speech, titled "Model-Integrated Computing", for the session on Model-Based Development, and Tools. Sztipanovits pointed out that despite it being a seemingly simple concept, building large systems from components is a very hard problem. In particular, the side-effects of component composition as manifested in component interactions often transpire only during system integration (compositionality problem), and the responsibility for design integrity lies with each system integrator (semantics problem). These problems are aggravated by physical requirements that cross-cut functional component-boundaries, and thus defy compositionality. Sztipanovits identified the following challenges for model-integrated computing: the creation and application of domain specific modeling languages (DSMLs), model synthesis, and model transformation. He then promoted the use of meta-modeling as a means for capturing the semantics of different target languages and execution models; he distinguished between domain models as capturing designs, and meta-models as capturing design invariants (such as types, constraints, and well-formedness rules). Sztipanovits also discussed the Generic Modeling Environment (GME), developed at Vanderbilt University, based on meta-modeling and model-transformation concepts. He concluded that domain-specific modeling languages and model-transformations are key technologies for future progress in embedded systems development, and that model-integrated computing is becoming a mature technology for the development of complex applications.

Two panel discussions complemented the keynote presentations. The first panel discussed "Research Challenges in Automotive Software" as well as the role of academia, industry and funding agencies in addressing these challenges. Panelists were *Hans-Georg Frischkorn* (then BMW Group), *K. Venkatesh Prasad* (Ford Motor Company), *Dev Kambhampati* (UC Discovery) and *Ramesh Rao* (California Institute for Telecommunications and Information Technology). Dis-

cussions emphasized the importance of software architectures in automotive software development and research, their integration into effective development processes, the availability of a defined middleware platform, the view of automotive software within broader system boundaries and the importance of user experience in the automotive design. The discussion also highlighted the need for an increased understanding of software as a product on the sides of both manufacturers and suppliers, the availability of business plans taking software into account, and access to engineers trained in system architecture and integration. Collaboration between industry and academia, as well as long-term fundamental research, is required to address these issues.

The second panel discussed "Challenges in Model-Based Design of Automotive Software". Panelists were *Werner Damm* (University of Oldenburg, Germany), *Edward C. Nelson* (Ford Motor Company), *Jürgen Bielefeld* (BMW Group) and *Carlo Ghezzi* (Politecnico di Milano, Italy). The discussion emphasized the importance of models that need to be kept consistent with software implementations and allow for incremental development and variant/product-line management. Models were also identified as a good means of communication between manufacturer and supplier. The importance of capturing system-wide views, such as the interactions of different system components, was also highlighted in the discussion. The utility of partial views, focused on separate services and addressing multiple levels of abstraction, was identified. Further research in developing modeling languages with thoroughly worked-out theories addressing the semantic level was suggested; a model repository was proposed as a valuable tool for the research community to compare modeling approaches and tools.

Selected Papers

This volume includes a selection of refereed technical and invited papers presented at the workshop. In the following we give a brief overview of the selected papers and their contents.

The contribution "Analyzing the Worst-Case Execution Time by Abstract Interpretation of Executable Code" by *Christian Ferdinand et al.*, addresses the validation of timing behaviors and memory usage as it occurs in embedded microprocessors by means of abstract interpretation of executable code.

In their paper "Quality Assurance and Certification of Software Modules in Safety Critical Automotive Electronic Control Units Using a CASE-Tool Integration Platform", *Klaus Müller-Glaser et al.* describe a CASE tool integration platform for quality assurance and certification of software modules.

In "On the Fault Hypothesis for a Safety-Critical Real-Time System", the paper accompanying his keynote presentation, *Hermann Kopetz* discusses the critical role of systematic failure management for systems prevelant in the automotive domain. This includes, in particular, the formulation of an explicit fault hypothesis, which has important consequences especially for the architecture design of safety-critical systems.

In "A Compositional Framework for Real-Time Guarantees", *Insup Lee et al.* describe a formal approach for establishing real-time properties for a composite system out of real-time properties for its parts.

Carlo Ghezzi et al. propose modeling component and service federations in "Validation of Component and Service Federations in Automotive Software Applications". Component federations can be modeled using statecharts (for individual components) and MSC variants (for interaction properties); service federations can be described using static architectural models, constraints on model transformations, and sequence diagrams for interaction properties.

The paper "Towards a Component Architecture for Hard Real Time Control Applications" by *Wolfgang Pree and Josef Templ* describes Giotto, a platform-independent, deterministic software model for embedded systems, with the goal of abstracting from the target hardware platform in the early development stages, and of supporting moving code modules from one ECU to another in the target system. One challenge addressed by Giotto is to specify timing-behavior independently from concrete scheduling algorithms and communication platforms.

In "Adding Value to Automotive Models", *Werner Damm et al.* describe advanced code-generation and validation techniques as a means for adding value to the models themselves. This can be accomplished by means of in-depth knowledge of the formal semantics behind the corresponding modeling tools, in-depth knowledge of the use of these tools, and extensive cooperation with the corresponding tool vendors.

Gabor Karsai, in "Automotive Software: A Challenge and Opportunity for Model-Based Software Development", identifies modeling and model-transformation as a common theme across the model-construction, analysis, and synthesis and integration phases of system development. According to Karsai, this makes the case for meta-programmable tools and corresponding tool-chains.

The paper "Software for Automotive Systems: Model-Integrated Computing" by *Sandeep Neema et al.* presents an exemplary design flow for automotive system development based on the Generic Modeling Environment (GME) and tool connectors for Simulink/Stateflow, Matlab, and specific code generators for the target platform.

Finally, in "Simulink Integration of Giotto/TDL", *Wolfgang Pree et al.* report on a case study carried out together with BMW on a throttle control; in this case study, the executable code was generated fully automatically from specifications of both the timing and the functional model. Using specialized translators the properties as specified in Simulink were transformed and simulated within a dedicated tool-set.

Outcome

The workshop clearly exhibited the state-of-the-art of automotive software engineering and pointed out various challenges in the area. This is also reflected by the papers selected for this volume. In particular, the idea of the workshop to bring together leading engineers from the Automotive domain with key researchers on an international level also stimulating the discussion between Europe and the US proved to be very fruitful and worked out perfectly. During the workshop significant progress was achieved towards developing a common understanding of the challenging problems in the automotive domain such as:

- standards for architectures and ways of structuring software systems in cars,
- a careful collection of significant data about the reliability of software in cars today and methodological steps to improve the reliability,
- better ways to model cars with respect to their software properties and structures during the development process,
- more sophisticated development processes incorporating recent scientific results from academia to improve the quality checking.

At the end of the workshop and also thereafter participants strongly expressed their satisfaction about the workshop and its usefulness to stimulate further research and progress in the area of automotive software engineering. Altogether the workshop was an overall success proving that the concept of the workshop accurately addressed the relevant issues and the appropriate community.

The organizers and editors extend their profound thanks to all workshop participants, authors, keynote speakers, panelists, reviewers, sponsors and members of the local organization team for their important contributions to the success of the workshop itself and of this post-proceedings volume.

March 2006

Manfred Broy
Ingolf H. Krüger
Michael Meisinger

Organizers

Manfred Broy
Ingolf H. Krüger

Referees

Scott Andrews
Luciano Baresi
Frederic Doucet
Carlo Ghezzi
Rajesh Gupta
Gabor Karsai
Luciano Lavagno
Insup Lee
Michael Meisinger
Massimiliano Menarini
Klaus D. Müller-Glaser
Edward C. Nelson
Wolfgang Pree
Insik Shin
Janos Sztipanovits

Local Arrangements

David Bareno
Martha Chavez
Diwaker Gupta
Ingolf H. Krüger
Jennifer Lee
Russell McClure
Don Peters-Coville
Sabine Rittmann
Marilyn Samms

Sponsors

California Institute for Telecommunication and Information Technology (Calit2)
National Science Foundation (NSF)
ARTIST, European Union (EU)
Deutsche Forschungsgemeinschaft (DFG)

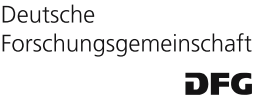

Table of Contents

Quality Assurance

Analyzing the Worst-Case Execution Time by Abstract Interpretation of Executable Code
Christian Ferdinand, Reinhold Heckmann, Reinhard Wilhelm 1

Quality Assurance and Certification of Software Modules in Safety Critical Automotive Electronic Control Units Using a CASE-Tool Integration Platform
Klaus D. Mueller-Glaser, Clemens Reichmann, Markus Kuehl, Stefan Benz 15

Real-Time Control

On the Fault Hypothesis for a Safety-Critical Real-Time System
Hermann Kopetz 31

A Compositional Framework for Real-Time Guarantees
Insik Shin, Insup Lee 43

Services and Components

Validation of Component and Service Federations in Automotive Software Applications
Luciano Baresi, Carlo Ghezzi 57

Towards a Component Architecture for Hard Real Time Control Applications
Wolfgang Pree, Josef Templ 74

Model-Based Development and Tools

Adding Value to Automotive Models
Eckard Böde, Werner Damm, Jarl Høyem, Bernhard Josko, Jürgen Niehaus, Marc Segelken 86

Automotive Software: A Challenge and Opportunity for Model-Based Software Development
Gabor Karsai 103

Software for Automotive Systems: Model-Integrated Computing
Sandeep Neema, Gabor Karsai 116

Simulink Integration of Giotto/TDL
Wolfgang Pree, Gerald Stieglbauer, Josef Templ 137

Author Index 155