Software Engineering for Automotive Systems at ICSE 2004
Workshop Summary

Alexander Pretschner
ETH Zurich, Departement Informatik, Information Security,
Haldeneggsteig 4, CH-8092 Zurich, Switzerland
pretscha@inf.ethz.ch

Christian Salzmann, Thomas Stauner
BMW Car IT, Petuelring 116, D-80809 Munich, Germany
{christian.salzmann,thomas.stauner}@bmw-carit.de

Abstract

This paper provides a summary of the ICSE 2004 workshop on software engineering for automotive systems. We outline the main characteristics of automotive software engineering and indicate how the workshop papers relate to these characteristics. Furthermore, we summarize the discussion at the workshop.

Keywords: automotive software engineering, ICSE workshop.

Introduction

Software is taking a leading role in automotive development and innovation. It is commonly estimated that software and electronics account for ca. 90% of all innovations. Already today cost for software and electronics in current premium cars amount to up to 40% of the overall cost. The applications are not anymore limited to classical embedded control systems, such as airbag control software, but cover a broad range from mission critical embedded systems in the X-by-wire field, driver assistance to infotainment and personalization in the MMI (Man Machine Interface) area.

The next generation of premium cars is expected to host a cumulated amount of up to one gigabyte of binary code of software deployed via a set of embedded platforms. To design, implement and manage a complexity of such a huge, heterogeneous distributed system with increasingly short innovation cycles, neither the techniques and methods of classical embedded systems are suitable, nor the known ones in the desktop and business software domains. To tackle this challenge, we furthermore need new adapted software engineering methods for the automotive domain that allow to design the different software types specifically, corresponding to their requirements, and later on integrate the system parts into one reliable and manageable system. We therefore see the discipline of automotive software engineering (ASE) as a massively emerging research field with heavy impact in industry.

This background already motivated a panel discussion on automotive software engineering at ICSE 2003 which was well attended. This year we therefore decided to organize this workshop with a special focus on the adoption of established software engineering concepts to the automotive domain. This can be a building block towards automotive software engineering as an explicit discipline of software engineering including tailored techniques and methods.

The rest of the paper is organized as follows. In the following section we outline the main characteristic of ASE. The paper then lists the workshop presentations and relates them. Finally, we summarize the concluding discussion at the workshop.

Characteristics of Automotive Software Engineering

It is vital to understand the basic characteristics of a domain in order to introduce domain specific engineering techniques. In this section we sketch these characteristics for the automotive domain. A more detailed description is given in [2].

Heterogeneity. Automotive software is very diverse ranging from entertainment and office related software to safety-critical real-time control software. We cluster automotive software into three groups: *Infotainment & Comfort Software* (including system services with soft real-time requirements), *Safety or Security Critical Software* and *(hard) real-time Software.* In the infotainment domain, traditional IT concepts play an important role, while for control systems, which usually are hard real-time systems, techniques form control theory must be intermixed with methods from computer science.

Emphasis on Software Integration. Automotive software is developed in tight collaboration between OEMs (original equipment manufacturers) and suppliers. As a result, OEMs have to integrate independently developed systems or software. While this always is a difficult task for a complex system, the situation in automotive software engineering is even worse, because suppliers usually have a lot of freedom in how they realize a solution. (In business IT the client would often strongly constrain the technologies that may be used by a supplier to facilitate integration and maintenance.) Furthermore, the OEM usually only has black-box specifications of the subsystems to be integrated which makes successful testing and error localization more difficult. In addition, distributed development in itself makes it difficult or impossible for the OEM to modify parts of the subsystems in order to localize errors.

Unit based cost structure. Due to the large produced quantities, production and material cost by far outweigh engineering cost in the automotive domain. Traditionally, the cost per produced unit plays the decisive role for all decisions on how car functions are realized. For automotive software engineering this is problematic, because it drives system engineers to design hardware that is as cheap as possible for the required function. This usually means that performance and memory is small. Thus, the software engineer is confronted with a hardware that is only sufficient for the function to be implemented if the implementation is highly optimized. In particular this means that abstraction layers possibly have to be sacrificed to performance, and modern development

methodologies and programming languages often cannot be used. This of course makes developing complex software that is correct and delivered in time a very difficult task. In the recent past important automotive system development projects were delayed because of software problems.

Further main drivers of complexity are a large number of product variants, depending on the marketing strategy of the OEM, and long product life cycles. Service for the product typically has to be available for 15+ years which also affects automotive software.

Summary of workshop presentations

The introduction to the workshop was the invited talk on *Challenges in Automotive Software Engineering* by Jürgen Bortolazzi (DaimlerChrysler AG) which gave a very good summary of the field and of typical problem areas.

The papers and their authors were as follows.

1. D. Nyström, M. Nolin (Mälardalen University), A. Tesanovic, C. Norström, J. Hansson (Linköping University). COMET: A Component-Based Real-Time Database for Automotive Systems
2. S. Buehne, K. Lauenroth, K. Pohl (University of Duisburg-Essen), M. Weber (DaimlerChrysler AG). Modeling Features for Multi-Criteria Product-Lines in Automotive Industry
3. M. Conrad, I. Fey (DaimlerChrysler AG), S. Sadeghipour (ITPower Consultants). Systematic Model-Based Testing of Embedded Control software: The MB3T Approach
4. M. Horstmann, E. Schnieder (TU Braunschweig), P Mäder, S. Nienaber, H.-M. Schulz (EXTESSY AG). A framework for interlacing Test and Design
5. I. Krüger, D. Gupta, R. Mathew, P. Moorthy, W. Phillips, S. Rittmann (University of California, San Diego). Towards a Process and Tool-Chain for Service-Oriented Automotive Software Engineering
6. H. Lonn, T. Saxena (Volvo Technology Corp.), M. Nolin (Mäardalen University), M. Torngren (Royal Institute of Technology, Sweden). FAR EAST: Modeling an Automotive Software Architecture Using the EAST ADL
7. L. Vitkin, T.K. Jestin (Delphi Electronics & Safety). Incorporating Autocode Technology into Software Development Process
8. G. Costagliola, S. Di Martino, F. Ferrucci, M. Risi (University of Salerno), G. Oliviero, D. Freni (Elasis Scpa). Towards a Design Pattern for Automotive Telematics Systems

Papers' Summaries. Papers 1 and 2 are in the context of two specific topics in ASE, real-time databases for the continuously growing amount of information to be managed in cars, and product line engineering for automotive systems. Papers 3 and 4 address automotive specific testing techniques. Papers 5-7 are concerned with (model-based) specification and development methodologies for embedded software in the automotive domain. A design pattern for infotainment systems was introduced in paper 8.

Overall the range of papers very well reflects the heterogeneity of the domain influenced by classic IT concepts, like the design pattern in paper 8, as well as by specific engineering methods for reactive systems, as e.g. in papers 6 and 7. The specification and testing techniques in papers 3 to 7 provide a basis to support the integration of independently developed software/systems, which is a characteristic of the automotive domain.

All papers are published in the workshop proceedings [1].

Workshop discussion and outlook

The discussion after the workshop focused around two topics. The papers following model-based engineering approaches claimed they introduced a systematic way of performing the considered process phase/phases. This stimulated discussion whether a process phase can be called systematic just because of the use of models or whether systematic development and model-based development rather are orthogonal to each other.

The second point of discussion was whether the complexity in automotive software engineering and the domain characteristics are inherent to the (technical) domain, or to what extent they are a result of business decisions, like the desire to offer many product variants that can be shaped more or less freely.

As a whole the workshop stimulated the discussion between researchers and practitioners in automotive software engineering. The workshop contributions reflected that domain specific techniques, like in paper 6, are about to emerge, thereby creating the basis for future tailored techniques and tools. This success and the positive feedback of the workshop participants motivates our plans to continue this workshop at next year's ICSE.

Acknowledgements. We thank the PC members for their paper reviews, and the ICSE organization committees for the excellent organization.

References

[1] A. Pretschner, C. Salzmann and T. Stauner (2004): Proc. of the ICSE 2004 Workshop on Software Engineering for Automotive Systems. Edinburgh, May 25, 2004.

[2] C. Salzmann and T. Stauner (2003): Automotive Software Engineering – An emerging application domain for software engineering. In *Proc. of FDL'2004 – Forum on Specification and Design Languages, Frankfurt, Sept. 23-26, 2003*.