

Application of MDS D for the development of automotive software development tools

Uwe Honekamp, Michael Hoffmann

PND – Tools for Networks and distributed Systems
Vector Informatik GmbH
Ingersheimer Straße 24
70499 Stuttgart
uwe.honekamp@vector-informatik.de
michael.hoffmann@vector-informatik.de

1 About MDS D Activities at Vector

Since 2006, Vector provides an authoring and code generation tool [RV05] for the development of automotive embedded software. This tool has been created using Eclipse and the JDT and has been heavily based upon model-driven software development (MDS D) approaches, namely open Architecture Ware [oA06].

The main purpose of this tool is to provide means for formally modelling automotive application software according to the AUTOSAR standard [AR04] and then on user's demand automatically generating a lean software layer (the so called AUTOSAR run-time environment, or RTE) located between application software and basic software.

The usage of open Architecture Ware is supposed to ease and accelerate changes in the tool meta-model and code generation facility. In this case open Architecture Ware is not only used to support the creation of a specific tool but is also a fundamental part of the tool itself because the tool also provides a model-based code generation facility.

Although being relatively new to the approach of open Architecture Ware to MDS D, the authors already have a background in providing tools that (to some extent) also implement concepts addressed by MDS D. For example, Vector already developed and released a tool suite called DaVinci [HW04].

Since its first release in 2003, Vector's DaVinci tool suite supports users at model-based design of automotive embedded systems on many levels. One of the main goals of the DaVinci project has been and continues to be the formal modelling of automotive application software and its interfaces to the basic software.

One of the key concepts of the DaVinci modelling approach is the formal definition of so-called software-components and their interfaces to each others. The definition of software-components leads to a strongly decomposed application software that supports reusability across different platforms as well as distributed development and testing of application software.

Based on models defined by means of DaVinci it is possible to automatically generate a lean software layer located between the basic software of an automotive ECU and the application part. With the advent of the AUTOSAR initiative [AR04], this software layer is now called the AUTOSAR run-time environment (RTE).

2 Outlook and future work

The authors are convinced that MDSO is about to become one of the key technologies for the future of software tool development. However, in our opinion the technology is obviously still in an emerging phase that requires serious consideration of stable key concepts that can already be used for industrial-level development of software tools.

Our roadmap for applying MDSO concepts primarily foresees the improvement of the existing MDSO-based tool but also a consideration whether to carry over MDSO concepts to tools already existing in Vector's tool landscape, most prominently the DaVinci tool suite.

In general, it would be conceivable to improve several aspects of the DaVinci tool suite by using MDSO techniques:

- Automatic derivation of meta-model implementation based on the formally defined meta-model.
- Definition of import and export filters for AUTOSAR specification formats [AR04].
- Definition of import and export filters for Simulink [TM05]. This allows for the specification of functional behaviour.
- Generation of source code for the AUTOSAR RTE.

3 About us

Uwe Honekamp received his engineering diploma and doctorate degree from Paderborn University. He has a 20 years background in the development of software tools for modelling and code generation for distributed mechatronic and automotive embedded systems. He also contributes to the specification of the AUTOSAR standard.

Michael Hoffmann received his diploma degree from university of applied science Trier. He has 10 year experience in tool development and 4 years in design of distributed automotive systems.

4 Bibliography

- [HW04] Honekamp, U; Wernicke, M.: Development of distributed automotive software – the DaVinci Methodology. Proc. IFIP Working Conference on Distributed and Parallel Embedded Systems, Toulouse, France, August 23-26, 2004.
- [TM05] The Mathworks, Inc.: Using Simulink ®, Version 6, 2005.
- [AR04] Heinecke, H. et al.: AUTomotive Open System ARchitecture - An Industry-Wide Initiative to Manage the Complexity of Emerging Automotive E/E-Architectures, Convergence, October, 18-20 2004, Detroit, 2004.
- [RV05] Rudorfer, M.; Völter, M.: Domain-specific IDEs in embedded automotive software, EclipseCon, Burlingame, CA, February 28 - March 03, 2005.
- [oA06] www.openarchitectureware.org