Chapter XV
Quality Improvement in Automotive Software Engineering Using a Model-Based Approach

Tibor Farkas
Fraunhofer Institute FOKUS, Germany

ABSTRACT

Premium quality and innovation are the cornerstones of the leading positions of car manufacturers and suppliers in the world market. The permanently increasing complexity of in-car electronics and the rapidly growing amount of automotive software running on embedded electronic control units, places higher demands on quality assurance for the future. Quality cannot be implemented into software on embedded control units after their development. Methods for defects detection have to be constituted to automatically stop development to fix a problem before the defect continues downstream. In addition preventive actions have to be taken in respect of front-loading quality and reliability. An automatic and tool independent check of custom development rules, quality standards and enterprise wide guidelines can support the quality assurance process in the development of automotive control software. In the domain of automotive software engineering there is a lack of automated checking for standard conformance. Especially, a formal and tool independent notation of rules to follow is missing. In this chapter, the model-based design of automotive vehicle functions is taken as an example to show how textual rules describing development standards to be met can be transformed into a formal notation using the open standards Meta Object Facility and Object Constraint Language. Thereafter these rules can be checked automatically. The feasibility of this approach is shown by a software demonstrator.
1. INTRODUCTION

Premium quality and innovation are the cornerstones of the leading positions of car manufacturers and suppliers in the world market. Quality assurance starts in early development phases and is a joint responsibility of both, the car manufacturers and their suppliers. The use of electronic control units (ECU) has grown rapidly in modern vehicles. This has gone along with an increase in variety and complexity of these electronic systems and their networking over different busses. New functions are preferably implemented in embedded software that is distributed on a rising number of networked control devices (VDI, 2005). Having nowadays approximately 10 to 20 different ECUs on a single vehicle network, the integration of software from many different suppliers is a difficult task (Mercer, 2001).

X-by-wire systems are an upcoming technology in the automotive industry that replaces the traditional mechanical and hydraulic control systems with electronic control systems using electromechanical actuators and human-machine interfaces. They constitute the basis for vehicle control systems and assistance systems that support and relieve the driver during his driving assignment. Purely mechanical or hydraulic systems will be replaced by mechatronic systems. They are integrated into the vehicle environment intelligently. As a result, software will become a technology that is critical for the business competition especially in the automotive manufacturing domain (Jackman, 2005). High demands for quality of these systems and the great complexity as well as the rapidly growing interaction between single subsystems lead to strong requirements on development methods and development tools. As an example, Figure 1 shows the increasing number of software inside telephone-hands-free equipment (an automotive telematics application) over a period of 10 years (Form, 2006).

While simple functions like displaying the digits were realized on an embedded display device, nowadays rich functions such as voice dialing and phone book synchronization are controlled by embedded software.

Coping with future system complexity under stringent quality requirements on the one hand and the expected lead in innovation on the other hand are important factors of business success for automotive manufacturers and their suppliers (Liggesmeyer, 2005). Nevertheless, the quality losses of the electronic devices in the vehicle are still engraving. According to the current

Figure 1. Amount of embedded software in an automotive telephone hands-free equipment over a period of ten years (Form, 2006)
Related Content

A Quantitative Risk Assessment Model for the Management of Software Projects
[www.igi-global.com/chapter/quantitative-risk-assessment-model-management/28113?camid=4v1a](www.igi-global.com/chapter/quantitative-risk-assessment-model-management/28113?camid=4v1a)

A Case Study of Dynamic Analysis to Locate Unexpected Side Effects Inside of Frameworks
[www.igi-global.com/article/a-case-study-of-dynamic-analysis-to-locate-unexpected-side-effects-inside-of-frameworks/126614?camid=4v1a](www.igi-global.com/article/a-case-study-of-dynamic-analysis-to-locate-unexpected-side-effects-inside-of-frameworks/126614?camid=4v1a)

On the Study of Complexity in Information Systems
[www.igi-global.com/chapter/study-complexity-information-systems/38172?camid=4v1a](www.igi-global.com/chapter/study-complexity-information-systems/38172?camid=4v1a)

Resolving Conflict in Code Refactoring
Lakhwinder Kaur, Kuljit Kaur and Ashu Gupta (2013). *Designing, Engineering, and Analyzing Reliable and Efficient Software* (pp. 149-161).
[www.igi-global.com/chapter/resolving-conflict-code-refactoring/74879?camid=4v1a](www.igi-global.com/chapter/resolving-conflict-code-refactoring/74879?camid=4v1a)