Chapter XIV
Assuring Maintainability in Model–Driven Development of Embedded Systems

Stefan Wagner
Technische Universität München, Germany

Florian Deissenboeck
Technische Universität München, Germany

Stefan Teuchert
Durchstreichen, MAN Nutzfahrzeuge AG, Germany

Jean-François Girard
Durchstreichen, MAN Nutzfahrzeuge AG, Germany

ABSTRACT

In model-driven software development as much as in classical code-driven development maintenance costs make up the bulk of the total life cycle costs of a software system. However, as development methods in MDSD differ from classical methods, assuring the maintainability of systems built with MDSD requires companies to adjust their quality assurance to work with the new paradigm and the novel type of development artefacts. As the automotive industry has already applied model-driven approaches for some time (usually in the form of Matlab/Simulink) it proves to be a fertile ground to advance assurance methods for the maintainability of model-based systems. In this chapter we describe a two-dimensional quality metamodel and present an instance that defines maintainability for MDSD with Matlab/Simulink and TargetLink. We exemplify how such a model serves as the basis of all quality assurance activities and report on experiences made in an industrial case study with one of the leading international providers of commercial vehicles and transport solutions.
INTRODUCTION

Maintenance costs constitute the major part of the total life cycle costs of a software system (Lientz, Bennet, Swanson, & Burton, 1980; Boehm, 1981; Erlikh, 2000). Besides organisational issues such as knowledge management and labour turnover, the long-term maintenance costs are largely predetermined by various quality attributes of the software system itself, such as its comprehensibility and modifiability.

In model-driven software development (MDSD) as much as in classical code-driven development organisations need methods and processes to continuously monitor these quality attributes to ensure the maintainability of software systems. However, as development methods in MDSD differ from classical methods, assuring the maintainability of systems built with MDSD requires companies to adjust their quality assurance to work with the new paradigm and the novel type of development artefacts.

In the development of embedded systems in general and automotive systems in particular, model-driven approaches become more and more common. Up to 80% of the production code deployed on embedded control units today is generated from models specified using domain-specific formalisms (Beine, Otterbach, & Jungmann, 2004). Several major companies develop software with model-based tools like Matlab/Simulink and TargetLink. As these technologies enabled companies to apply model-based software development already some time ago, this field proves to be a fertile ground to advance assurance methods for the maintainability of model-based systems.

Although model-driven architecture (MDA) is often proposed to ease the maintenance of systems, maintainability is also an issue in MDSD (Seifert, Beneken, & Baehr, 2004). The MDA approach is mainly concerned with technology - especially platform – change. The problems connected with changing the underlying technologies are simplified by layering models that abstract from such technological details. However, as also stated in (Seifert et al., 2004) portability and hence changing the technology is only one of many challenges in maintenance. Therefore, the other issues need also to be dealt with in MDSD.

In this chapter we give a short introduction on model-based approaches, especially in the field of embedded systems development, and describe how the maintainability of such models can be assured. We introduce a unique quality metamodel that enables us to rigorously define maintainability and present a model instance that has been developed in an industrial case study with MAN Nutzfahrzeuge, a supplier of commercial vehicles and transport systems. We illustrate how such a model can be used as versatile basis for maintainability-related quality assurance techniques. These techniques include manual activities like model reviews as well as automated quality assessments like static model analyses. We conclude by highlighting the differences between quality assurance for MDSD and classical development.

EMBEDDED SYSTEMS DEVELOPMENT WITH MATLAB/SIMULINK/TARGETLINK

We investigate a slightly different flavour of MDSD than the MDA approach proposed by the OMG. In embedded systems development model-based tools such as Rhapsody, ASCET or Matlab/Simulink are commonly used. However, there is no explicit need to have different types of models on different levels and the modelling language is often not UML. Nevertheless, many characteristics are similar and quality-related results can easily be transferred to an MDA setting.

Matlab/Simulink is a tool commonly used in the automotive industry. It constitutes a representative example for a model-based tool-chain in embedded systems development. The original