Chapter 7.6
Integrating Quality Criteria and Methods of Evaluation for Software Models

Anna E. Bobkowska
Gdańsk University of Technology, Poland

ABSTRACT

Successful realization of the model-driven software development visions in practice requires high quality models. This chapter focuses on the quality of models themselves. It discusses context-free and context-dependent quality criteria for models and then moves on to methods of evaluation which facilitate checking whether a model is good enough. We use linguistic theories to understand groups of criteria and their impact on other models, software product and the process of software development. We propose a strict distinction of the impacts of visual modeling languages, models of the system and tools for quality criteria. This distinction is helpful when designing the methods of evaluation and making decision about the point in time, scope and personnel responsible for quality assessment. As the quality criteria and several methods of evaluation has usually been considered separately we propose a methodology which integrates them. Such an integrated approach provides the following benefits. It allows for designing methods of evaluation based on quality criteria and elements of the model (or modeling language) in the context of specific needs. It can be applied for management of the scope of evaluation with quality criteria as well as configuration of the method to a specific situation. It allows for flexible and efficient conduct of the evaluation with selection of the methods of evaluation. Finally, this chapter presents case studies which illustrate the approach.

INTRODUCTION

Models play the central role in model-driven software development approaches. However, only good models really support work of analysts, managers and developers. Only good models have the potential to facilitate dealing with complexity, direct micro-process of software development, enable more efficient performance of the difficult tasks, facilitate management and increase satisfaction at work. On the opposite, “bad” models may cause mistake, waste time and be annoying in use. Despite that, much research has been done in the area of technology of model transformations and tools supporting this task. We argue that there is a need to focus on the quality of visual modeling languages (VML), which are used for creating the models as well as quality of models themselves.

This chapter takes a practical perspective to quality in modeling. In order to improve the quality of models one needs to understand quality criteria for models, know efficient methods for conducting evaluation and understand relations between them for effective configuration and management. The level of maturity of the technology is not satisfactory nowadays and one can observe several problems from the point of view of project and general knowledge.

From the point of view of the software project, there are several unrelated methods with limited scope of application. It is unclear what could be benefits of using them and how they could be integrated. Perhaps the most popular method for improving model quality are inspections or reviews with checklists. However, one often needs to customize the checklist provided with methodologies for a specific project. It is easy to remove unnecessary checkpoints, but the problem remains when deciding which checkpoints need to added. The problem of inflexible methods of evaluation appears even more clearly in the perspective of the current trend of software process configuration. It assumes that companies do not choose a process but rather they make a configuration of roles, artifacts and tasks for a given type of project within a framework. It means that model artifacts, modeling activities and model checking activities are objects of configuration. Thus, we need objective criteria to make decisions about the scope of modeling as well as methods of evaluation for software models.

From the point of view of maturity of modeling technology, some objective criteria and practical methods of evaluation are needed in order to provide frameworks for customization of the concrete software projects. There is a need to understand criteria why modeling languages should have such elements, diagrams, visualizations, etc. The methods of evaluation are useful for making efficient evaluation and quality criteria behind such decision allow for trade-offs when compromises are necessary. They could be applied by Object Management Group (OMG) members working on modeling standards, e.g., Unified Modeling Language (UML) and its profiles or Business Process Modeling Notation (BPMN). As these standards influence advances in practice, the increase of the maturity of modeling technology should result in better perception of this technology and more successful application in practice. This approach could be useful also for committees which work on application-type modeling languages, e.g., internet applications; special type of models, e.g., task models for user behavior modeling; and models for emerging technology, e.g., aspect-oriented technology. Finally, they could be applied in the area of domain specific modeling. The tasks of designing and evaluating modeling languages in this case are performed in software companies which means a universal application of the methods of visual modeling language evaluation.

The objective of this chapter is to present an approach to dealing with model quality in a way which integrates quality criteria and methods of evaluation. The quality criteria are useful for understanding which aspects are evaluated and for managing the scope of evaluation. The methods of evaluation are necessary for efficient conduct of evaluation. The integrated approach should provide the following benefits. It should allow for designing methods of evaluation based on quality criteria and elements of the model (or modeling language) in the context of specific