Empirical Software Engineering Group

ESEG

Centro de Informática U·F·P·E

Universidade Federal de Pernambuco

DEFINITIONS AND APPROACHES TO MODEL QUALITY IN MODEL-BASED SOFTWARE DEVELOPMENT

This briefing reports the practices found to improve the quality of models.

FINDINGS

- Most practices are concerned with error prevention, while some also facilitate error detection.
- Lack of process contributed to a large number of errors since modelers do not have process that guides where to start and conventions that provide uniformity.
- Lack of quality assurance (for example reviews) led to a staggering number of errors.
- Design models that originated from analysis had fewer errors than those originated as designs.
- Having control of the quality of the MDE tooling is necessary to have control of the quality of systems being developed using the tools.
- For evaluating the progress and the quality of development artifacts, a study identified tasks of model-centric software engineering and the information (e.g., questions about the UML model) that is required to fulfill these tasks.
- It seems clear that having high quality requirements process is important, and having ways of saying ‘‘how good” the requirements are is thus a clear advantage.
- Quality assurance techniques such as collecting metrics by tools and analyzing them should be part of a modeling process.
- Guidelines or conventions regarding the modeling process:

  - the early modeling effort should cover the entire breadth of the domain.

  - Identify ‘‘out of scope” use cases as early as possible.

  - Discover business objects and their services through use case definition with sequence diagrams.

  - Elicit requirements and processes by starting at interfaces and modeling inward.

  - Create simple content; the implication is that you should not add additional aspects to your models unless they are justifiable.

  - Single source information; you should also model a concept once and once only, storing the information in the best place possible.

- It is also an advantage if artifacts in one phase may be used to develop or generate artifacts in the next phase in order to achieve completeness and consistency.
- Many of the proposed conventions may be enforced by tools, but the problem today is that most modeling tools enforce syntactical and aesthetic constraints very weakly and semantic constraints are not enforced at all.
- For error detection, one may use tools such as Design Advisor or perform inspections.
- There is little empirical evidence in the covered literature regarding the benefits of conventions and the results of few student experiments are not conclusive.
- The quality impact of conventions seems to depend on the task, the complexity of conventions and tool support as well and empirical studies should describe these factors better in order to help evaluating the usefulness and cost.
- Some studies suggest that UML is unnecessarily complex in many ways, and this inherent deficiency hinders coherent modeling and comprehension of systems.
- Technical solutions that involve sophisticated CASE tools to impose consistency alleviate manual consistency maintenance, but they do not address the core problem of mental integration. Some indicate the use of single-diagrams.
- Some authors have proposed to use the observer pattern to achieve consistency between different views in conceptual models: as the user changes a specification in one view, all other views are informed about the modification.
- Practices can, of course, be combined, for example DSMLs often include constraints and formal semantics to prevent errors and facilitate generation, and the number of diagrams is often reduced. Having a model-based development process that includes guide.

Keywords:

Model quality
Model-driven development

Who is this briefing for?

Software engineers practitioners who want to make decisions about quality in model-based software development on scientific evidence.

Where the findings come from?

All findings of this briefing were extracted from the systematic review conducted by Mohagheghi et al. .

What is a systematic review?

cin.ufpe.br/eseg/slrs

What is included in this briefing?

The main findings of the original systematic review.

What is not included in this briefing?

Additional information not presented in the original systematic review.

Detailed descriptions about the studies analised in the original systematic review.

To access other evidence briefings on software engineering:

cin.ufpe.br/eseg/briefings

For additional information about ESEG:

cin.ufpe.br/eseg

ORIGINAL SYSTEMATIC REVIEW REFERENCE