Insights on Applicability of MBSE & Use of MBSE in Optimization of System Design

Michel R.V. Chaudron
Professor in Software Engineering
Joint Computer Science and Engineering dept.
Chalmers and Gothenburg university
chaudron@chalmers.se

Link to publications

- [http://dblp.uni-trier.de/pers/hd/c/Chaudron:Michel_R=V=](http://dblp.uni-trier.de/pers/hd/c/Chaudron:Michel_R=V=)
Outline of talk

Introduction
  • Division Software Engineering

Insights on Applicability of MBSE
  • How are models used in SE practice?
  • How much modeling is good enough? – Quality of Modeling

Use of MBSE in Optimization of System Design
  • Automated Quality-driven System Architecture Design
  • Case study in automotive

Software Engineering Division - Faculty

- Cyber Physical Systems
- Autonomous Vehicles/Drones
- Continuous Software Engineering
- Requirements
- Software Architecture
- Model-driven software Development
- Testing
- Software Metrics
- Security
Education Programs

• Software Engineering and Management
  • Combines technical skills with business skills

• Project-based learning
  • Self-driving miniature vehicles

• Model-based software engineering
  • Started using Papyrus in 2014

• Internship – often with industry
  • Volvo, Ericsson, Saab, ...

My Background

• Since 2012 Prof. in Software Engineering in Gothenburg, Sweden

• Background
  — Ph.D. in Formal Methods: Calculi for program derivation (formal methods, concurrency, process algebra)
  — Visiting student/researcher in the UK.
  — 2 years in industry as IT consultant
  — Assistant & Associate prof. at TU Eindhoven & Leiden, Netherlands

• Industrial research collaborations
  — Software Architecture, Design & Modeling, Component-based SE
  — Enterprise Architecture, IT Portfolio, Alignment, Agility
  — Research Collaborations: EU/ITEA – o.a. Philips, Nokia, CSEM, also: KLM, ING, T-mobile, Shell, Capgemini, KPMG
Introduction: Research Interest

- Is software modelling effective?
- What are the pay-offs of investing in early design/architecture/modeling?
  Fewer defects?
  Cheaper maintenance? ...

- Analysis and Reasoning about Extra-Functional Quality Properties of System Architectures

- Many modeling approaches around; focus on UML in custom software development

Increasing Complexity of Software

Modeling is inevitable, Automation is needed
Insights on Applicability of MBSE

- Work with ph.d. students:
  - Johan Muskens (Eindhoven)
  - Christian Lange (Eindhoven)
  - Werner Heijstek (Leiden)
  - Ariadi Nugroho (Leiden)
  - Hafeez Osman (Leiden)

The Model-based SE Spectrum

"What's a Model?"
"The code is the model"
"Code and model coexist"
"The model is the code"
"Let's do some design"

‘Model’ means different thing to different people

Model driven architecture: Principles and practice, Brown, A.W., Software and Systems Modeling, 2004
Phase 1: Explore via Case Studies

<table>
<thead>
<tr>
<th>Case study</th>
<th>No. of classes</th>
<th>No. of person-years spent on modeling</th>
<th>No. of team members</th>
<th>CMM level (estimated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>794</td>
<td>15</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>168</td>
<td>20</td>
<td>20</td>
<td>2–3</td>
</tr>
<tr>
<td>C</td>
<td>108</td>
<td>20</td>
<td>10</td>
<td>2–3</td>
</tr>
<tr>
<td>D</td>
<td>716</td>
<td>Unknown</td>
<td>Unknown</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>443</td>
<td>10</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>75</td>
<td>10</td>
<td>Unknown</td>
<td>1</td>
</tr>
<tr>
<td>H</td>
<td>478</td>
<td>Unknown</td>
<td>Unknown</td>
<td>1</td>
</tr>
<tr>
<td>I</td>
<td>705</td>
<td>12</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>J</td>
<td>51</td>
<td>12</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>K</td>
<td>14</td>
<td>1.5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>L</td>
<td>46</td>
<td>1.5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>M</td>
<td>73</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>369</td>
<td>5</td>
<td>5</td>
<td>1–2</td>
</tr>
</tbody>
</table>

Your project here? Mail me: chaudron@chalmers.se

Styles of Modeling

- Sketch
- Communication
- Recipe for construction
Models in Software Development

+ Understanding
+ Communicating
+ Analysing / Predicting
+ Guiding

On-boarding new members
Distributed development
Easing maintenance

Doing modeling is a quality improving activity

UML in practice

Communication

More effort ⇒ More expensive

Recipe for construction
Modelling style and model purpose

Styles of using UML
- as a sketch – thinking tool/understanding
- for communicating system design
- as a blueprint – guide the implementation work
- as an implementation (MDA) - code generation

Does the use of modeling improve software quality?

A large number of developers indicated the use of UML improves understandability and modularity
Level of Detail for Sequence Diagrams (i.e. behavior models) is significantly (negatively) correlated with defect density. More detailed model => fewer defects.

What about class (i.e. structure) diagrams?

Cost of SQ = Achievement Cost + Non-conformance Cost
Experience in Formal Modeling

• BOS project (1998)
• At CMG (later Logica, now CGI)
• Safety Critical project
• Fixed-time, fixed cost project

• Applied Formal Methods
  • Promela and model-checking
  • But only for critical properties
  • Also: reviewing and inspections
  • Modeling effort proportional to risk

• CMG started business offer: center for critical software

Economy of Modeling
Chaudron’s conjecture

Modeling finds defects early

Does modelling find the right defects?

Quality Assurance for Software Design / UML Model

- How to assess quality?
- How to check conformance of code to design?
How to assess quality UML models?

Design Heuristics
- Coupling, Cohesion, absence of anti-patterns

Completeness
- Interaction of classes must be described in SD
- Methods of CD must be called in SD

Consistency
- Messages in SD must correspond to method in class diagram

In many cases it cannot be automatically determined whether a flaw is an incompleteness or an inconsistency.

Automatically Checking Quality of Design

Quality = absence of ‘bad things’ such as omissions, spaghetti

UML model

UML Analysis Tool

Quality Metrics/Rules
- Completeness
- Consistency

Visualization of model + metrics
3D visualization of design measurements

Your cases welcome...

http://www.youtube.com/watch?v=G3HJ_QR9EG4


Findings from State of the Practice

UML is used in different ways for different purposes

- Incompleteness
  - Modelers focus on complex and critical parts

- Disproportion
  - Different parts of the system are modeled different in level of quality

- Inconsistency
  - Conflicting information in different views of one model

Quality Assurance for modeling should address these
Findings on Technology Transfer

• There is a gap in transiting prototypes into industrial use
  • No/little credits for academics in building tools (funding, prestige)

• Maintenance:
  • There is no funding for maintaining long-lived software tools

• Development ‘end user’ Company
  • Interesting idea, but not yet industrial strength
  • This fails to generate necessary feedback!

• Tooling Company:
  • We don’t have any company interested yet

How to organize feedback?

Use of MBSE in Optimization of System Design

This is the work of Ramin Etemaadi Ph.D.’s research at Leiden University, The Netherlands

https://scholar.google.com/citations?user=EUD3EkJ4AAAAJ
Quality Driven Component-based Design

With 9 types of processors and 6 types of busses, there are 7,346,640,384,000,000 architecture design alternatives.

Evaluating Non-Functional Properties of System Architectures

Time-consuming manual work
Limited insight in design space
Modeling & Analysis Tool

- Processor
- Memory
- Interconnect

Eclipse Plug-in, Java-based, compatible with AADL, EAST-ADL, Auto-SAR

Example Specification of Component Behaviour
Quality-Driven Architecture Design Method and Tool

Architecture Model

- HW & SW-components
- generates automated
- manual modeling / profiling tool
- compare alternatives automated
- generate new alternative architectures automated

Performance Analysis
Safety Analysis
Cost Analysis

Scenario-based Analysis

- Performance or Schedulability analysis are performed with scheduling algorithms (RMA, EDF, CBS)
- Analysis results in task latencies, number of missed deadlines, CPU, memory and bus utilization
Automotive case study

Algorithm: SMS-EMOA
Modeling & Analysis in PMS

Tool development started with interviewing system architects

- They maintain between 3 and 10 critical scenario’s
- They start with estimates, gradually replace with measurements
- Not all sketchy parts are replaced, some stay sketchy.

Modeling & Design

• Notation on whiteboard is very rich

• We are using development tools as enriched typewriters

• Modeling = encoding a design in a modelling language

• Designing = making design decisions
Challenge for Modeling tools

- Understand role of modeling in designing process
  - Need improved attention to HCI in development tools:
    - task-analysis studies in system design
    - Designers mix rigorous and sketchy modelling
    - Driven by ‘economy’ of design

Concluding Remarks

- There are payoffs for modeling in:
  - improved communication
  - improved quality of design and code

- Modeling is a means to support Design

- Economy of engineering demands a mix of sketchy/informal and rigorous modeling

- Usability/HCI of Development Environments deserves more attention!
  - Task analyses of developers needed
Tooling Ecosystem Platform

- **Input Language**
  - IL1: AADL
  - IL2: SysML
  - IL3: UML+Martes
  - Extensible ...

- **Parser**
- **Internal Representation Meta-model**
- **Model Transformations**
- **Analysis ‘Language’**
- **Performance: Markov Q**
- **Safety: Failure Trees**
- **Cost: Dedicated**
- **Optimization Modules**
  - Selection
  - Mutation

Lessons Learned

- **Medical Systems**:
  - Lowered level of detail & completeness in favour of economy of modeling

- **Consumer Electronics**
  - Overhead of modeling became too high due to fast changing platform/technology
  - Successful introduction of MBSE requires:
    - ‘champion’
    - Management decision
    - Investment by experts, pay-off by team
    - Lack of agreement/alignment by peers
What to model and how

- Context dependent
  - Understand your ‘receiving party’
  - Ask about receiver expectation and needs
  - Do you share domain knowledge?
  - Agree on conventions

- Pay attention to behaviour modeling
  - Sequence / Activity Diagrams
How much to model?

• Design the architecture in “Sprint 0”

• Time-boxed

  “For a three week iteration, spend […] at most one day at the start of the iteration drawing UML for the hard, creative parts”

  Craig Larman
  “Applying UML and Patterns”, 2009

Online Software Design Principles Test

Consider the designs of the same system below:

A
B
C
D

Which one is a better design, considering assignment of responsibility?

With Dave Stikkolorum (Leiden)

Ongoing development:

• Moving to UML editor in the browser
• Interactive feedback on design
Practical Advice for UML modeling

Naming and layout-conventions
*The Elements of UML2.0 Style*, Scott Ambler, Cambridge Univ Press, 2005

Traceability
Organize traceability from Use Cases to Sequence diagrams (tool/conventions)

Reviews & Inspections
Guidelines by e.g. Shull et.al., Biffl,

Tooling
Version Management - (e.g. CVS, SVN, ...)
Metrics : SDMetrics  http://www.sdmetrics.com/

Ongoing Ph.D. students

- Dave Stikkolorum  – Teaching software design by modeling
- Bilal Karasneh  – Towards a repository of UML design models
- Ana Fernandez  – Effect of modeling in Software Maintenance
- Truong Ho Quang – Interactive Views for Navigating Complex systems
- Rodi Jolak  – Multimodal User Interfaces for Modeling
Needs for Modeling tools

Tools
- Flexibility
- Integratability
  - Open API’s
- Scalability
- Performance
- Usability

Companies
- Customize
  - Configure/Tailor
  - In-company standardization
Empirical Research

... is a way of gaining knowledge by means of observation or experience.

Theory
SE body of knowledge

Practice
People/Skills, Processes, Methods, Technology, Artefacts

Topics: CMM, OO, Agile, MDA, ...
Methods: Experiment, Case study, Survey, Project Repositories ...

‘Hypothetical’ Benefits of Modeling

UML Modeling

Developer Benefits
Better Understanding Problem Domain

Process Benefits
Improved requirements
Improved Design Compliance
Improved Design Quality

Product Benefits
Fewer Defects
Reduced rework
More efficient Testing
More efficient Maintenance

Improved Quality
Improved Productivity

Improved Design Quality
Shared System Model

Improved Communication
Better Understanding Solution Space

More Accurate Estimating
Better Risk management

Project Management Benefits

Improved maintenance effort
Reduced testing effort