Seminar „Model-Based Quality Engineering“

Jun.-Prof. Dr.-Ing. Steffen Becker

Model-Driven Software Engineering

Software Engineering Group
Outline

- Basic Requirements
- Preliminary Dates
- Seminar Guidelines
- Presentation of the Topics
Basic Requirements

- Completion of a seminar thesis in English (approx. 20 pages written in LaTeX)
- Design and run a presentation
- Presentation is 20 min, to be held in a block seminar (presumably between 01.02.-15.02.2014)
- Internal peer-review among the students, but also review by your supervisor
Preliminary Dates

- **Tu, 15.10., 16:00:** Topic presentation
- **We, 23.10., 16:30:** Final topic assignment, introduction to scientific working

The following dates have their deadline 23:59 MEZ:

- **Su, 24.11.:** Outline and literature references (student)
- **Su, 08.12.:** Seminar thesis for review (student)
- **Tu, 10.12.:** Assignment of peer reviews (supervisors)
- **Su, 15.12.:** Completed peer-review (student)
- **Su, 12.01.:** Presentation for supervisor feedback (student)
- **Su, 19.01.:** Supervisor feedback: presentation (supervisors)
- **Su, 23.02.:** Camera-ready version of thesis (student)
- **Su, 02.03.:** Supervisor feedback: thesis (supervisors)
- **Su, 16.03.:** Final hand-in of thesis (student)

Presentations (block seminar): 01.02.-15.02.2014
Seminar Guidelines

- Wednesday, 23.10., 16:30 in room ZM1.02-48: Final topic assignment and seminar rules – Participation is mandatory

- Rules to create thesis and presentations
  - Next week
  - Use templates introduced next week for LaTeX and Powerpoint
Topic Selection

- We will use a Doodle poll with three options: yes, maybe, no
- We try to minimize conflicts
- Final conflict resolution is First-Come, First-Served
- Poll will be opened tomorrow morning

- We have 12 topics
- Topics will be assigned with priorities according to a random shuffled order of all registered participants
Motivation for the Seminar
Model-Based Quality Engineering

Model-Based

- Models are not central, but second-class artefacts
- Models typically serve for analyses and documentation purposes

Quality Engineering

- Internal, external or process quality
- Is achieved by analyzing models upfront
- Constructive Quality Assurance
Topics
1. Model Driven Testing and Gamification

- Model Driven Testing: tedious, time-consuming
- Game: interesting, engaging
- Gamification: the use of game design elements in non-game context

Your task:
- Investigate characteristics of model-driven testing and gamification
- Propose potential scenarios in which gamification could help with model driven testing

Literature:

Supervisor: Jinying Yu
2. QoS and Software Adaptability

- Quality-of-Service (QoS), e.g. response time, depends on workload & software design
- Adapting software design at runtime helps to ensure QoS level

Problem:
- Which adaptations help to maintain QoS level?
- How to build systems that maintain QoS levels autonomously?

Your task:
- Research influence of software adaptations on QoS

Literature:

Supervisor: Matthias Becker
3. Specification and Matching of Service Prices

**Service matching** compares service requests to provided service specifications in order to enable service discovery and service composition.

Functional and non-functional properties have to match, including service prices (e.g., single use price vs. price-per-hour, ...) → complex price models.

Literature (starting point for your literature search):
- Kiemes, Oberle: "Service Pricing" in Handbook of Service Description. Springer 2012

Further Literature (address the topic marginally):

Goal of the Seminar:
- Compare different approaches for specification / matching of service prices

Supervisor: Marie Christin Platenius
4. Survey: Performance Requirements

- Performance requirements are often imprecise
- Software projects fail because of performance issues

- Problem:
  - What are (good) performance requirements?
  - How to specify them?
  - How to ensure them?

- Your task:
  - Conduct Literature Review for Performance Requirements Elicitation Methods

- Literature:

- Supervisor: Matthias Becker
5. Synthesis of Components from Distributed Live Sequence Charts (dLSCs)

- **Synthesis** derives an implementation from a specification
  - Here: dLSCs → Petri-Net Components

- **Live Sequence Charts**: formal variant of sequence diagrams

- **Goal of the Seminar:**
  - Understand the approach by Fahland and Kantor
  - Identify assumptions and limitations

- **Literature:**

- **Supervisor**: Christian Brenner
6. Domain-specific Model Checking for Mechatronic Systems with UPPAAL PORT

- Mechatronic systems require exhaustive verification
- Problem: huge gap between domain-specific design and model checkers
- UPPAAL PORT is an approach towards domain-specific model checking

Your tasks:
- Compare UPPAAL PORT to MECHATRONICUML
- Assess PORT’s applicability to networked mechatronic systems
- Feasibility analysis: model checking MECHATRONICUML with PORT?

Supervisor: Christopher Gerking

7. Performance Engineering for Multi-Core Systems

- Multi- and Many-Core: > 10 cores even in consumer devices
- However, old software uses often just 1-2 cores
- Major reengineering tasks need to be done

**Your task:**
- Investigate multi-core software engineering challenges
- Create a recent survey of reengineering approaches for multi-core systems

**Literature:**

**Supervisor:** Steffen Becker
8. Agile Model Driven Web Engineering

Model Driven Web Engineering facilitates portability, abstraction and productivity

- Agile Approach provides constant interaction with customers and enhances former prototypes continuously

- Goals of the Seminar:
  - Understand both Agile approach and model driven web engineering approach
  - Identify existing hybrid approaches

- Literature:

- Supervisor: Jinying Yu
9. Modelling Architectural Decisions

Mostly documented as plain text, sometimes little illustrations

- difficult to comprehend, analyse, verify and ensure completeness and consistency

- Solution: graphical model

- Your task:
  - Present architecture evolution approach and compare with existing architecture decision making approaches

- Literature:

- Supervisor: Claudia Priesterjahn
10. Software Architecture Erosion

- Usually code gets large
  - -> growing deviations from intended design, because of changing requirements
- Erosion: software gets hard to test, comprehend, and extend
- Need to control erosion

Your task:
- Present foundations on software erosion
- Compare methods presented in paper due to most important capabilities
- Present methods selected by your own criteria

Literature:
- von Detten, Markus; Platenius, Marie Christin; Becker, Steffen: Reengineering Component-Based Software Systems with Archimetrix. Journal of Software and Systems Modeling 2013

Supervisor: Claudia Priesterjahn
11. Assessing the Quality of Model-to-Model Transformations

- Model transformation is core enabling technology for MDSE

- Problem: No established engineering methods for developing M2M transformations
  - Which transformation technology to use?
  - Quality of the implemented transformation?

- Your tasks:
  - Systematic literature overview
  - Comparison of existing quality assessment approaches
  - Towards an engineering approach for M2M based on quality assessment

Supervisor: Sebastian Lehrig, Jan Rieke

Literature:
- S. Lehrig: Assessing the Quality of Model-to-Model Transformations Based on Scenarios, Master’s Thesis, 2012
12. Modeling the Cloud

• Cloud Computing:
  – On-demand access to computing resources
  – Pay-per-use price model
• Modeling approaches are emerging
• Goals of the Seminar:
  → Analyze given approaches, find other approaches, and compare
• Starting Points:
  – http://www.planforcloud.com/
  – http://www.madeiracloud.com/
  – http://aws.amazon.com/de/cloudformation/
• Supervisor: Sebastian Lehrig
13. Using the OPC UA Information Model for MechatronicUML

- MechatronicUML supports a complex software component model
- OPC UA provides a specification for data exchange in industrial automation

Your task:
- Study the MechatronicUML Component Model and the OPC UA Information Model
- Create a relation of MechatronicUML Component Model with an instance of the OPC UA Information Model

Literature:
- Becker, S; Brenner, C; Brink, C; Dziwok, S; Löffler, R; Heinzemann, C; Pohlmann, U; Schäfer, W; Suck, J; et al.: The MechatronicUML Design Method - Process, Syntax, and Semantics, 2012

Supervisor: Uwe Pohlmann
14. Timing Requirements in Automotive Systems

- Two architecture „languages“ for automotive systems
  - EAST-ADL
  - AUTOSAR

- Allow to specify model-based timing requirements for automotive systems on different abstraction levels

- Goal of the seminar
  - understand overall methodology
  - investigate,
    - how timing requirements are refined across hierarchy levels
    - how they relate to functional requirements

- Literature
  - TIMMO Project: TADL: Timing Augmented Description Language Version 2
  - EAST-ADL Consortium: Specification EAST-ADL V2.1.11

- Supervisor: Jörg Holtmann