Operating Systems Seminar

Concepts of Composition

Christine Jakobs

Operating Systems Group, TU Chemnitz
Preliminary Warning

- You can give your presentation and submit your written paper in German
- Nevertheless it is necessary to have good knowledge in English
- It is necessary to read ...
  - ... a lot (it is a seminar :) )
  - ... original literature!
Composition

- Concept from Mathematics transferred to Computer Science
- Roughly speaking: How to combine parts into a whole
- For example:
  - Combine different parts of a model into an integrated view
  - Combine different software components into a program
- Different definitions and use cases in different areas of computer science
- For example:
  - Navigating through models and manipulating structures of model elements (France 2007)
  - Integrate web services to create new value-added web services (Hamadi 2003)
  - ...

Goal of this seminar is to find an overall concept of composition
Solution strategy

▶ Understand the concept of composition in each area
▶ Abstract the information:
  • How is composition defined?
  • How is composition used?
  • What are the means?
  • How is the usage in comparison to other areas?
  • ...maybe more
▶ Map findings to an abstract definition
  • Syntactic composition
  • Semantic composition
  • Property preserving

Map findings to a formal definition
(Maybe) a Formal Definition

- Following the applied form of model theory from Stachowiak, each model is
  - Is a representation of the reality
  - An abstraction of unnecessary information
  - To answer a certain question

- Therefore, every area of interest is basically using models and languages to build them
- For examining their definition of composition we try to adapt a certain algebraic view:
  - A modeling language $M$ is a set of well-formed models $m$ (each element is basically a model by itself)
  - Each model gets a semantics by mapping the artifacts/concepts to a semantic domain $D$
  - e.g. in programming languages each concept has a semantic meaning out of the semantic domain

- This mapping is a function $sm: M \rightarrow D$
- This mapping can be abstracted onto all systems $S$ having this semantics = loose semantics
- e.g. Following constraints: 1. There is a class „Person“ 2. All of its instances have an attribute „name“
  Then all systems following this two constraints have the same semantics


(Maybe) a Formal Definition

- If we compose model elements/models we want to have consistent syntactics and semantics
  - Syntactic composition is much easier and usually a solved problem
  - Semantic composition is hard and may not in all areas be a solved problem

- Model composition takes two models as an input and has a composed model as output
- This may be property preserving on semantic level -> Semantic domain is preserved
  - On one model
  - On both models
- Composition may emerge semantic properties
  - E.g. security properties may arise out of composition

What to do in the seminar?

▶ Decide for one area of interest:
   • Model composition
   • Service composition (SOA, Web services)
   • Aspect oriented modeling
   • Architectural level
   • Model simulation
   • OOP
   • Databases

▶ Or one of the overall topics
   • Base Object Model Specification
   • Meta Object Facility (MOF)

▶ Read through the paper(s) provided

▶ Find other papers
   • Related work section
   • Google Scholar (try to find newer papers)
   • Digg deeply and do not give up
   • …

▶ Proceed with the solution strategy
Administrative Stuff

- Decide for a topic (2 weeks time, Scheduling Policy is FCFS)
- Write me an e-mail with your decision
  - Topic
  - Your details (Name, Matriculation number, course of study)
  - Type of seminar (Pro-, Haupt-, Forschungsseminar)
- Start with your work
  - See hints on the other slides
- Supervision must be claimed...
  - Arrange meetings
  - At least 1 time before your presentation
- Presentation (before exam period)
- Written paper (till end of semester)

- If you do not want to proceed with your seminar, please write me an e-mail!
Good luck!
Literature

- **Composability in general:**

- **Aspect oriented modeling:**

- **Architectural level:**

- **Composition of software components:**

- **Model composition:**
Literature II

▶ **Modelling and Simulation:**

▶ **OOP:**

▶ **Services (mixed SOA and Web Services):**