Overview

Theoretical Foundations of the UML

Lecture 1: Introduction

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

http://moves.rwth-aachen.de/teaching/ws-1415/uml/

13. Oktober 2014

Target audience

You are studying:

- Master Computer Science, or
- Master Systems Software Engineering, or
- Bachelor Computer Science, or
- ......

Usage as:

- elective course Theoretical Computer Science
- not a Wahlpflicht course for bachelor students
- specialization MOVES (Modeling and Verification of Software)
- complementary to Model-based Software Development (Rumpe)

Target audience (contd.)

In general:

- interest in system software engineering
- interest in formal methods for software
- interest in semantics and verification
- application of mathematical reasoning

Prerequisites:

- mathematical logic
- formal language and automata theory
- algorithms and data structures
- computability and complexity theory

Organization

Schedule:

	Day	Time	Lecture hall
Lecture	Mon	13:15 - 14:45	AH4
	Tue	10:00 - 11:30	5056
Exercises	Wed	16:15- 17:45	AH6

about 20 lectures in total; Keep track of website for precise dates!

People involved:

	Lecturer	EMail
Lectures	Joost-Pieter Katoen	`katoen@cs.rwth-aachen.de`
Exercises	Hao Wu	`hao.wu@cs.rwth-aachen.de`
	Souymodip Chakraborty	`chakraborty@cs.rwth-aachen.de`

Organization (contd.)

Assignments:

- (almost) weekly assignments
- available from course web-site
- first assignment: Wednesday October 22
- hand in solution at start next exercise class
- groups of maximally two students

Organization (contd.)

Examination: (6 ECTS credit points)

- written exam: February XY, 2015 (to be fixed soon)
- written re-exam: March 16, 2015 (afternoon)

Admission:

- at least 40% of exercise points

Motivation

Scope:

- Goal: formal description + analysis of (concurr.) software systems
- Focus: the Unified Modeling Language

More specifically:

- Sequence Diagrams (used for requirements analysis)
- Propositional Dynamic Logic
- Communicating Finite State Automata
- Hierarchical State Machines (behavioral description of systems)

Aims:

- clarify and make precise the semantics of treated UML fragments
- formal reasoning about basic properties of UML models
- algorithms to verify such properties

What this course is **NOT** about:

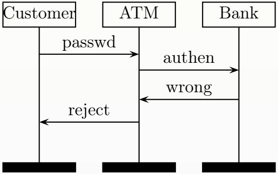
What is it **not** about?

- the use of the UML in the software development cycle
  - see the complementary course by Prof. Rumpe
- other notations of the UML (e.g., class diagrams, activity diagrams)
- what is precisely in the UML, and what is not
  - liberal interpretation of which constructs belong to the UML
- applying the UML to concrete SW development case studies
- empirical results on the usage of UML
- drawing pictures
- . . .

RWTH AACHEN UNIVERSITY

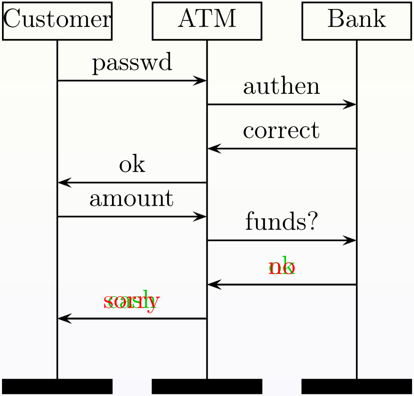
Sequence Diagrams

- origin: telecommunications: “Message Sequence Charts” (MSCs)
- describe interactions between processes (or objects)
- attractive visual formalism



- describes a possible scenario
- standardized by the ITU (Z. 120)
- adopted by the OMG for UML

Another example MSC

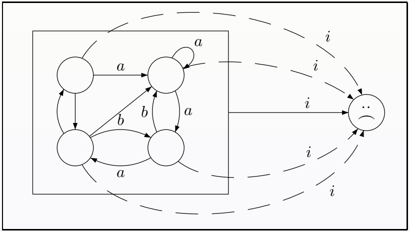


Message Sequence Charts

- MSCs
  (syntax, semantics, linearizations, races)
- Message sequence graphs
  (composition, expressiveness, compositional MSCs)
- Realizability
  (communicating finite-state machines, reachability in CFSMs, MSCs vs. CFSMs, boundedness)
- Regularity
  (regular MSCs and MSGs, realizability)
- Verification
  (positive + negative model checking, complexity results)
- PDL
  (Propositional Dynamic Logic for checking MSC properties)

Hierarchical State Machines

- finite state machines
  - no strategy for top-down or bottom-up development ("states have no structure")
  - no natural notion of hierarchy
  - uneconomical concerning transitions (e.g., high-level interrupt)



  - uneconomical wrt. parallel composition (exponential growth in # states)

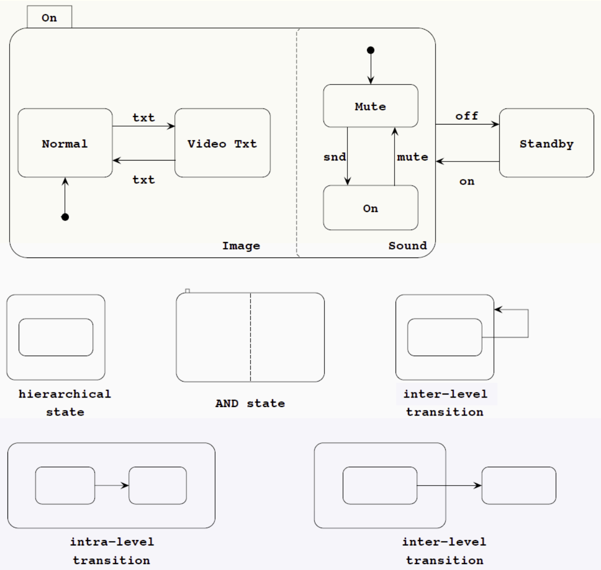
RWTH AACHEN UNIVERSITY

Statecharts

$$
\begin{array}{rcl}
\text{Statecharts} & = & \text{Mealy machines} \\
 & + & \text{depth} \\
 & + & \text{orthogonality} \qquad \text{[Harel'86]} \\
 & + & \text{broadcast} \\
 & + & \text{data}
\end{array}
$$

RWTH AACHEN UNIVERSITY

Statecharts (contd.)



Statecharts (contd.)

- Harel's Statecharts
  (basic features, syntax, state hierarchy, orthogonality, intra- and inter-level transitions)
- Semantics
  (main issues, formal semantics, flattening, succinctness)
- Verification
  (expressiveness, reachability, LTL model checking)