Software Design, Modelling and Analysis in UML

Lecture 02: Semantical Model

2013-10-23

Prof. Dr. Andreas Podelski, Dr. Bernd Westphal
Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lecture:
• Motivation: model-based development of things (houses, software) to cope with complexity, detect errors early
• Model-based (or-driven) Software Engineering

This Lecture:
• Educational Objectives:
  - Why is UML of the form it is?
  - Shall one feel bad if not using all diagrams during software development?
  - What is a signature, an object, a system state, etc.? What's the purpose of signature, object, etc. in the course?
  - How do Basic Object System Signatures relate to UML class diagrams?

Content:
• Brief history of UML
• Course map revisited
• Basic Object System Signature, Structure, and System State

Why (of all things) UML?

Note: being a modelling language doesn't mean being graphical (or: being a visual formalism [Harel]).

For instance, [Kastens and Büning, 2008] also name:
• Sets, Relations, Functions
• Terms and Algebras
• Propositional and Predicate Logic
• Graphs
• XML Schema, Entity-Relation Diagrams, UML Class Diagrams
• Finite Automata, Petri Nets, UML State Machines

Pro: visual formalisms are found appealing and easy to grasp.
Yet they are not necessarily easy to write!

Beware: you may meet people who dislike visual formalisms just for being graphical—maybe because it is easier to "trick" people with a meaningless picture than with a meaningless formula.
More serious: it's maybe easier to misunderstand a picture than a formula.

A Brief History of UML

• Boxes/lines and finite automata are used to visualise software for ages.

• 1970's, Software Crisis
  - Idea: learn from engineering disciplines to handle growing complexity.
  - Languages: Flowcharts, Nassi-Shneiderman, Entity-Relation Diagrams

• Mid 1980's:
  - Statecharts [Harel, 1987], StateMate TM [Harel et al., 1990]

• Early 1990's, advent of Object-Oriented - Analysis/Design/Programming
  - Inflation of notations and methods, most prominent:
    - Object-Modelling Technique (OMT) [Rumbaugh et al., 1990]
Which ones are really only hopes and expectations?. . .?
In how far/in what sense does it hold? Why? Why not? How can it be achieved?

N, E, f
G

Mathematics

Seriously: After the course, you should have an own opinion on each of these claims.

\[\mathbb{N} \in i \mathbb{I} \pi \mathbb{I} 1 \mathbb{I} 0 \mathbb{I} \mathbb{I} \mathbb{I} 0 \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \mathbb{I} \math
A BasicObjectSystemSignature is a quadruple \( \langle \mathcal{C}, \mathcal{B}, V, \text{atr} \rangle \) where:

- \( \mathcal{C} \) is a set of (basic) types,
- \( \mathcal{B} \) is a finite set of classes,
- \( V \) is a finite set of typed attributes, i.e., each \( v \in V \) has a type \( \tau \in \mathcal{C} \) or \( C_0,1 \) or \( C^* \), where \( C \in \mathcal{B} \),
- \( \text{atr} : \mathcal{B} \rightarrow 2^V \) maps each class to its set of attributes.

Example: \( \langle \mathcal{C}, \mathcal{B}, V, \text{atr} \rangle = (\{\text{Int}\}, \{\text{C,D}\}, \{\text{x: Int, p: C}_0,1, \text{n: C}^*\}, \{\text{C} \mapsto \{p, n\}, \text{D} \mapsto \{x\}\}) \)

A BasicObjectSystemStructure is a domain function \( \langle \mathcal{B} \rangle \) which assigns to each type a domain, i.e.,

- \( \tau \in \mathcal{C} \) is mapped to \( \langle \tau \rangle \),
- \( C \in \mathcal{B} \) is mapped to an infinite set \( \langle C \rangle \) of (object) identities.

Object identities only have the "\( = \)" operation; object identities of different classes are disjoint, i.e., \( \forall C,D \in \mathcal{B} : C \neq D \rightarrow \langle C \rangle \cap \langle D \rangle = \emptyset \).

Note:\( \langle \mathcal{B} \rangle \) denotes \( \bigcup C \in \mathcal{B} \langle C \rangle \); analogously \( \langle \mathcal{B}^* \rangle \).


