Software Design, Modelling and Analysis in UML

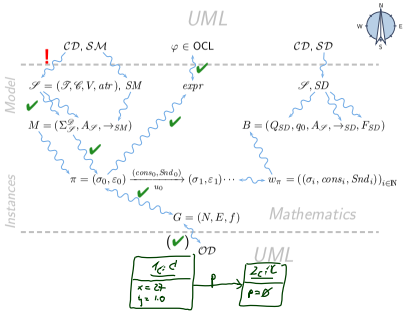
Lecture 06: Class Diagrams I

2014-11-11

Prof. Dr. Andreas Podelski, **Dr. Bernd Westphal**

Albert-Ludwigs-Universität Freiburg, Germany

Course Map



Contents & Goals

Last Lecture:

- OCL Semantics
- Object Diagrams

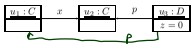
This Lecture:

- **Educational Objectives:** Capabilities for following tasks/questions.
  - What is a class diagram?
  - For what purposes are class diagrams useful?
  - Could you please map this class diagram to a signature?
  - Could you please map this signature to a class diagram?
- **Content:**
  - Final notes on object diagrams.
  - Study UML syntax.
  - Prepare (extend) definition of signature.
  - Map class diagram to (extended) signature.
  - Stereotypes – for documentation.

The Other Way Round

The Other Way Round

- If we **only** have a picture as below, we typically assume that it's **meant to be** an object diagram wrt. **some** signature and structure.

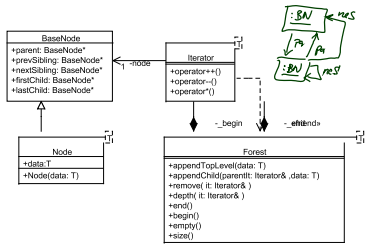


- In the example, we can conclude that the author is referring to **some** signature $\mathscr{S} = (\mathcal{T}, \mathcal{C}, V, atr)$ with at least
  - $\{C, D\} \subseteq \mathcal{C}$ or also func. dep. known
  - $T \in \mathcal{T}$
  - $\{x : C_1, p : C_2, z : T\} \subseteq V$
  - $\{x\} \subseteq atr(C)$
  - $\{p, z\} \in atr(D)$

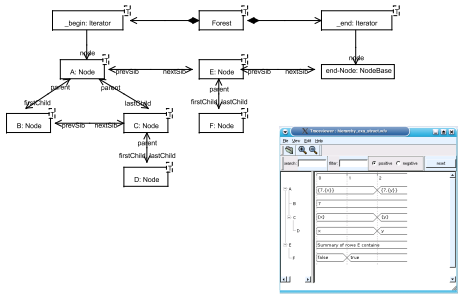
  and a structure with
  - $0 \in \mathcal{D}(T)$
  - $\{w_1, w_2\} \subseteq \mathcal{D}(C)$
  - $\{w_3\} \in \mathcal{D}(D)$

Example: Object Diagrams for Documentation

Example: Data Structure [Schumann et al., 2008]

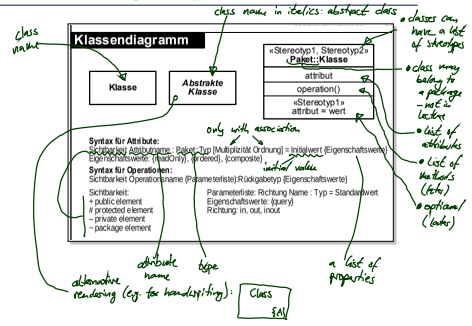


Example: Illustrative Object Diagram [Schumann et al., 2008]



UML Class Diagrams: Stocktaking

UML Class Diagram Syntax [Oestereich, 2006]



What Do We (Have to) Cover?

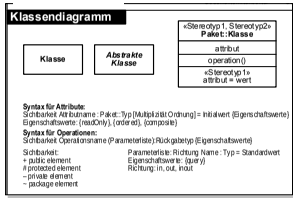
A **class**

- has a set of **stereotypes**,
- has a **name**,
- belongs to a **package**,
- can be **abstract**,
- can be **active**,
- has a set of **operations**,
- has a set of **attributes**.

Each **attribute** has

- a **visibility**,
- a **name**, a **type**,
- a **multiplicity**, an **order**,
- an **initial value**, and
- a set of **properties**, such as **readOnly**, **ordered**, etc.

Wanted: places in the signature to represent the information from the picture.



Extended Signature

Recall: Signature

$\mathscr{S} = (\mathscr{T}, \mathscr{C}, V, atr)$ where

- (basic) types $\mathscr{T}$ and classes $\mathscr{C}$, (both finite),
- typed attributes V, τ from $\mathscr{T}$ or $C_{0,1}$ or C_*, $C \in \mathscr{C}$,
- $atr : \mathscr{C} \to 2^V$ mapping classes to attributes.

Too abstract to represent class diagram, e.g. no "place" to put class **stereotypes** or attribute **visibility**.

So: **Extend** definition for classes and attributes: Just as attributes already have types, we will assume that

- classes have (among other things) **stereotypes** and
- attributes have (in addition to a type and other things) a **visibility**.

Extended Classes

From now on, we assume that each class $C \in \mathscr{C}$ has:

- a finite (possibly empty) set S_C of **stereotypes**,
- a boolean flag $a \in \mathbb{B}$ indicating whether C is **abstract**, (a=true iff abstract)
- a boolean flag $t \in \mathbb{B}$ indicating whether C is **active**.

We use $S_{\mathscr{C}}$ to denote the set $\bigcup_{C \in \mathscr{C}} S_C$ of stereotypes in $\mathscr{S}$.
(Alternatively, we could add a set St as 5-th component to $\mathscr{S}$ to provides the stereotypes (names of stereotypes) to choose from. But: too unimportant to care.)

Convention:

- We write
$$\langle C, S_C, a, t \rangle \in \mathscr{C}$$
when we want to refer to all aspects of C.
- If the new aspects are irrelevant (for a given context), we simply write $C \in \mathscr{C}$ i.e. old definitions are still valid.

Extended Attributes

- From now on, we assume that each attribute $v \in V$ has (in addition to the type):
  - a **visibility**
  $$\xi \in \{\underbrace{\text{public}}_{:=+}, \underbrace{\text{private}}_{:=-}, \underbrace{\text{protected}}_{:=\#}, \underbrace{\text{package}}_{:=\sim}\}$$
  - an **initial value** $expr_0$ given as a word from **language for initial values**, e.g. OCL expresions. ($\in$)
    (If using Java as **action language** (later) Java expressions would be fine.)
  - a finite (possibly empty) set of **properties** P_v.
    We define $P_{\mathscr{C}}$ analogously to stereotypes.

Convention:

- We write $\langle v : \tau, \xi, expr_0, P_v \rangle \in V$ when we want to refer to all aspects of v.
- Write only $v : \tau$ or v if details are irrelevant.

And?

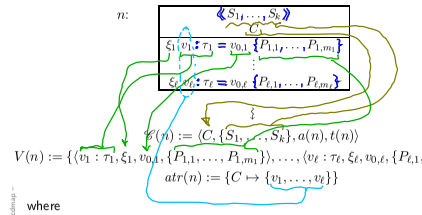
- **Note**:
  All definitions we have up to now **principally still apply** as they are stated in terms of, e.g., $C \in \mathscr{C}$ — which still has a meaning with the extended view.
  For instance, system states and object diagrams remain mostly unchanged.
- **The other way round**: **most** of the newly added aspects **don't contribute** to the constitution of system states or object diagrams.

- Then what **are** they useful for...?
- First of all, to represent class diagrams.
- And then we'll see.

Mapping UML CDs to Extended Signatures

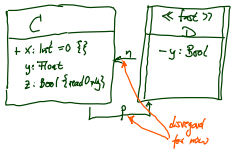
From Class Boxes to Extended Signatures

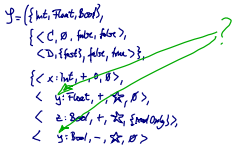
A class box n **induces** an (extended) signature class as follows:



where

- "abstract" is determined by the font:
$$a(n) = \begin{cases} \mathit{true} & \text{, if } n = \boxed{C} \text{ or } n = \boxed{C_{(A)}} \\ \mathit{false} & \text{, otherwise} \end{cases}$$
- "active" is determined by the frame:
$$t(n) = \begin{cases} \mathit{true} & \text{, if } n = \boxed{C} \text{ or } n = \boxed{\boxed{C}} \\ \mathit{false} & \text{, otherwise} \end{cases}$$





[Oestereich, 2006] Oestereich, B. (2006). *Analyse und Design mit UML 2.1, 8. Auflage*. Oldenbourg, 8. edition.

[OMG, 2007a] OMG (2007a). Unified modeling language: Infrastructure, version 2.1.2. Technical Report formal/07-11-04.

[OMG, 2007b] OMG (2007b). Unified modeling language: Superstructure, version 2.1.2. Technical Report formal/07-11-02.

[Schumann et al., 2008] Schumann, M., Steinke, J., Deck, A., and Westphal, B. (2008). Traceviewer technical documentation, version 1.0. Technical report, Carl von Ossietzky Universität Oldenburg und OFFIS.