Software Design, Modelling and Analysis in UML

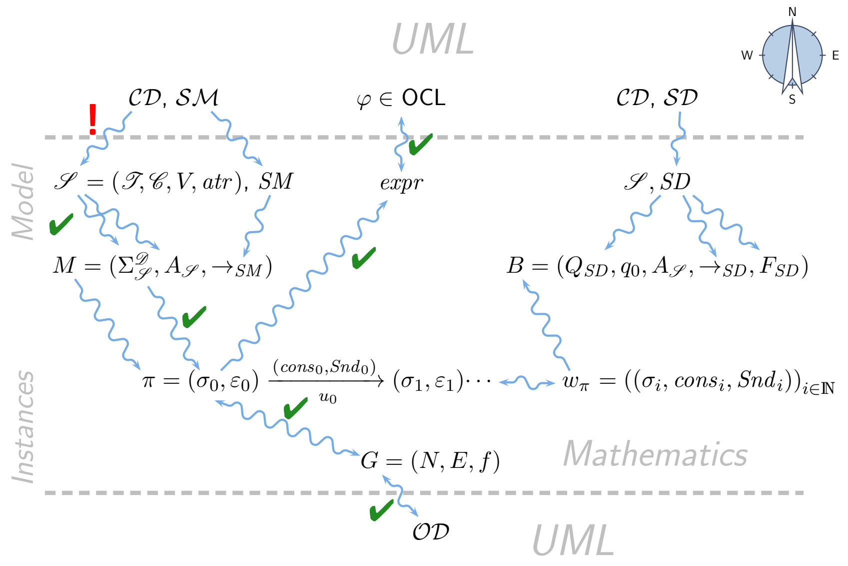
Lecture 6: Class Diagrams I

2015-11-12

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

Albert-Ludwigs-Universität Freiburg, Germany

Course Map



Contents & Goals

Last Lecture:

- Object Diagrams
  - partial vs. complete; for analysis; for documentation...

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:**
  - Study UML syntax.
  - Prepare (extend) definition of signature.
  - Map class diagram to (extended) signature.
  - Stereotypes.

UML Class Diagrams: Stocktaking

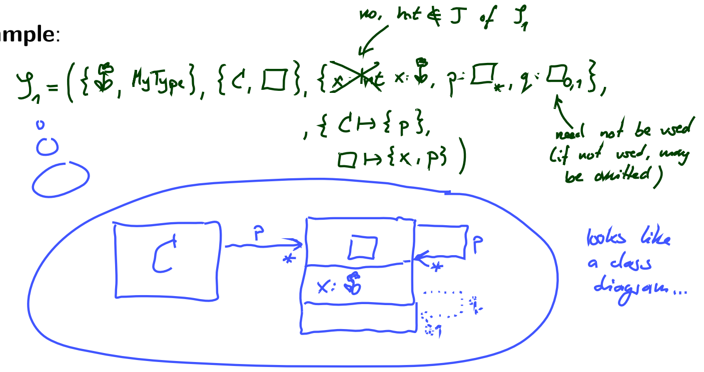
Recall: Signature vs. Class Diagram

Basic Object System Signature Another Example

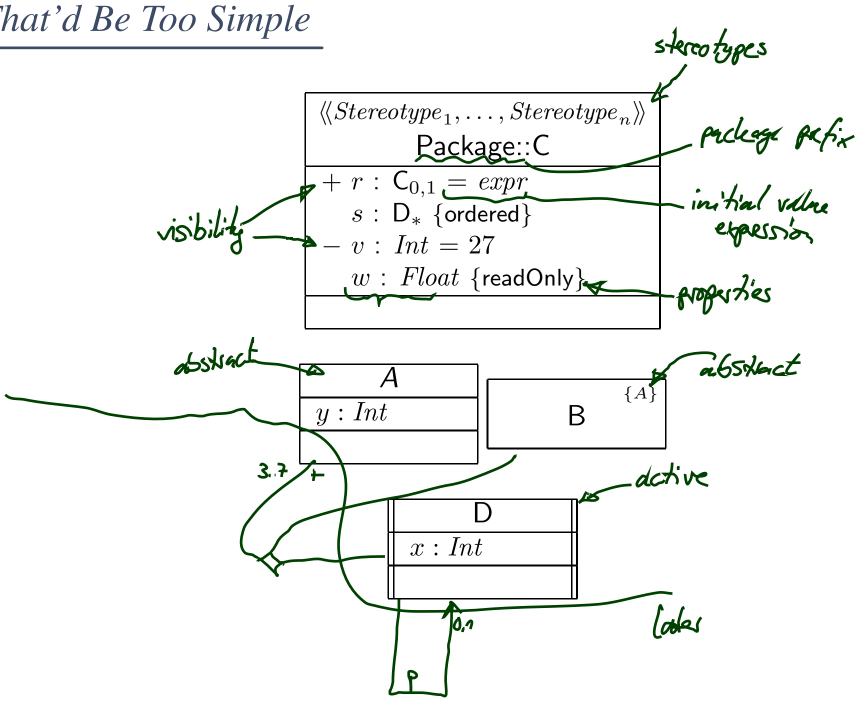
$\mathscr{S} = (\mathscr{T}, \mathscr{C}, V, atr)$ where

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

Example:



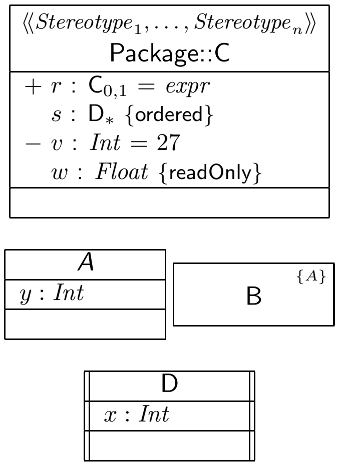
That'd Be Too Simple



What Do We Want / 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 **attributes**, ✓
- has a set of **operations**. (later)



Each **attribute** has

- a **visibility**,
- a **name**, ✓ a **type**, ✓
- a **multiplicity**, an **order**, (later)
- 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

Extended Signature

Definition. An (Extended) Object System Signature is a quadruple $\mathscr{S} = (\mathscr{T}, \mathscr{C}, V, atr)$ where

- $\mathscr{T}$ is a set of (basic) types,
- $\mathscr{C}$ is a finite set of classes $\langle C, S_C, a, t \rangle$ where
  - S_C is a finite (possibly empty) set of **stereotypes**,
  - $a \in \mathbb{B}$ is a boolean flag indicating whether C is **abstract**,
  - $t \in \mathbb{B}$ is a boolean flag indicating whether C is **active**,
- V is a finite set of attributes $\langle v : T, \xi, expr_0, P_v \rangle$ where
  - T is a type from $\mathscr{T}$, or $C_{0,1}, C_*$ for some $C \in \mathscr{C}$,
  - $\xi \in \{\underbrace{\text{public}}_{:=+}, \underbrace{\text{private}}_{:=-}, \underbrace{\text{protected}}_{:=\#}, \underbrace{\text{package}}_{:=\sim}\}$ is the **visibility**,
  - an **initial value expression** $expr_0$ given as a word from a **language for initial value expressions**, e.g. OCL, or C++ in the Rhapsody tool,
  - a finite (possibly empty) set of **properties** P_v.
- $atr : \mathscr{C} \to 2^V$ maps each class to its set of attributes.

We use $S_\mathscr{C}$ to denote the set $\bigcup_{C \in \mathscr{C}} S_C$ of stereotypes in $\mathscr{S}$.

or: $\in$

Conventions

- We write $\langle C, S_C, a, t \rangle$ if we want to refer to **all aspects** of C.
- If the new aspects are irrelevant (for a given context), we simply write C i.e. old definitions are still valid.
- We write $\langle v : T, \xi, expr_0, P_v \rangle$ if we want to refer **to all aspects** of v.
- Write only $v : T$ or v if details are irrelevant.
- **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.

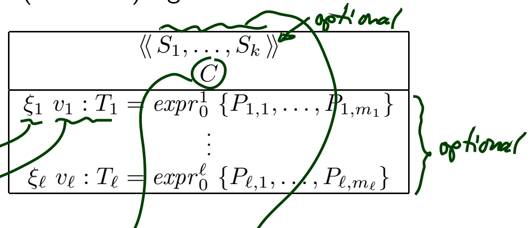
  For instance, system states and object diagrams will remain mostly unchanged.
- **The other way round**: **most** of the newly added aspects **do not contribute** to the constitution of system states or object diagrams.

Mapping UML Class Diagrams to Extended Signatures

From Class Boxes to Extended Signatures

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

n:



$$C(n) := \langle C, \{S_1, \ldots, S_k\}, a(n), t(n) \rangle$$

$$V(n) := \{\langle v_1 : T_1, \xi_1, expr_0^1, \{P_{1,1}, \ldots, P_{1,m_1}\}\rangle, \ldots, \langle v_\ell : T_\ell, \xi_\ell, expr_0^\ell, \{P_{\ell,1}, \ldots, P_{\ell,m_\ell}\}\rangle\}$$

$$atr(n) := \{C \mapsto \{v_1, \ldots, v_\ell\}\}$$

where

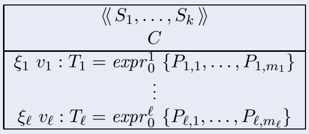
- "abstract" is determined by the font:

$$a(n) = \begin{cases} \textit{true} & \text{, if } n = \boxed{\textit{C}} \text{ or } n = \boxed{\textit{C}\ \{A\}} \\ \textit{false} & \text{, otherwise} \end{cases}$$

- "active" is determined by the frame:

$$t(n) = \begin{cases} \textit{true} & \text{, if } n = \boxed{\textbf{C}} \text{ or } n = \boxed{\| \text{C} \|} \\ \textit{false} & \text{, otherwise} \end{cases}$$

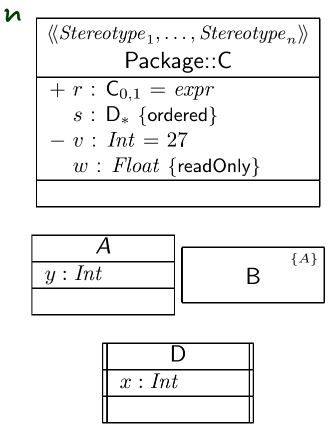
Example



$$C(n) := \langle C, \{S_1, \dots, S_k\}, a(n), t(n) \rangle$$

$$V(n) := \{\langle v_1 : T_1, \xi_1, expr_0^1, \{P_{1,1}, \dots, P_{1,m_1}\}\rangle, \dots, \langle v_\ell : T_\ell, \xi_\ell, expr_0^\ell, \{P_{\ell,1}, \dots, P_{\ell,m_\ell}\}\rangle\}$$

$$atr(n) := \{C \mapsto \{v_1, \dots, v_\ell\}\}$$



$C(n) = \langle C, \{Stereotype_1, \dots, Stereotype_n\}, false, false \rangle$

$V(n) = \{ \langle r : C_{0,1}, +, expr, \emptyset \rangle,$

$\langle v : Int, -, 27, \emptyset \rangle,$

$\langle s : D_*,$?

What If Things Are Missing?

It depends.

- What does the standard say? (OMG, 2011a, 121)

> "**Presentation Options.**
> *The type, visibility, default, multiplicity, property string may be suppressed from being displayed, even if there are values in the model.*"

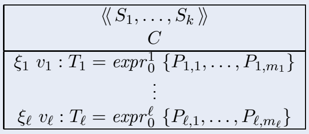
- **Visibility**: There is no "no visibility" — an attribute **has** a visibility in the (extended) signature.

  Some (and we) assume **public** as default, but conventions may vary.

- **Initial value**: some assume it **given by domain** (such as "leftmost value", but what is "leftmost" of $\mathbb{Z}$?).
  Some (and we) understand **non-deterministic initialisation** if not given.

- **Properties**: probably safe to assume $\emptyset$ if not given at all.

Example Cont'd

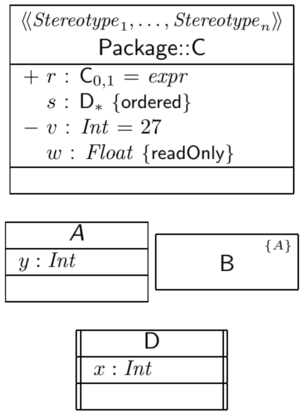


$$\rightsquigarrow$$

$$C(n) := \langle C, \{S_1, \ldots, S_k\}, a(n), t(n) \rangle$$

$$V(n) := \{\langle v_1 : T_1, \xi_1, expr_0^1, \{P_{1,1}, \ldots, P_{1,m_1}\}\rangle, \ldots, \langle v_\ell : T_\ell, \xi_\ell, expr_0^\ell, \{P_{\ell,1}, \ldots, P_{\ell,m_\ell}\}\rangle\}$$

$$atr(n) := \{C \mapsto \{v_1, \ldots, v_\ell\}\}$$



⋮

$\langle s : D_*, +, ?, \{ordered\} \rangle$

$\rightsquigarrow$

$\langle s : D_*, +, \ \ , \{ordered\} \rangle$

From Class Diagrams to Extended Signatures

- We view a **class diagram** $\mathcal{CD}$ as a graph with nodes $\{n_1, \ldots, n_N\}$ (each "class rectangle" is a node).
  - $\mathscr{C}(\mathcal{CD}) := \{C(n_i) \mid 1 \leq i \leq N\}$
  - $V(\mathcal{CD}) := \dot{\bigcup}_{i=1}^{N} V(n_i)$
  - $atr(\mathcal{CD}) := \dot{\bigcup}_{i=1}^{N} atr(n_i)$

- In a **UML model**, we can have **finitely many** class diagrams,

$$\mathscr{C}\mathscr{D} = \{\mathcal{CD}_1, \ldots, \mathcal{CD}_k\},$$

which **induce** the following signature:

$$\mathscr{S}(\mathscr{C}\mathscr{D}) = \left(\mathscr{T}, \dot{\bigcup_{i=1}^{k}} \mathscr{C}(\mathcal{CD}_i), \dot{\bigcup_{i=1}^{k}} V(\mathcal{CD}_i), \dot{\bigcup_{i=1}^{k}} atr(\mathcal{CD}_i)\right).$$

(Assuming $\mathscr{T}$ given. In "reality" (i.e. in full UML), we can introduce types in class diagrams, the class diagram then contributes to $\mathscr{T}$. Example: enumeration types.)

Is the Mapping a Function?

Question: Is $\mathscr{S}(\mathscr{C}\mathscr{D})$ **well-defined**?

References

References

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

OMG (2011a). Unified modeling language: Infrastructure, version 2.4.1. Technical Report formal/2011-08-05.

OMG (2011b). Unified modeling language: Superstructure, version 2.4.1. Technical Report formal/2011-08-06.

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.