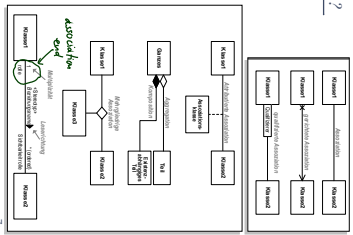


What Do We (Have to) Cover?

- An association has **just a link to**
 - a name,
 - at least two ends,
 - a reading direction, and **the order of the diagram**
- Each end has
 - a set of stereotypes
 - a role name,
 - a multiplicity,
 - a set of properties such as **unique, ordered**, etc.
- Qualifiers, (**not in lecture**)
- a visibility,
- a navigability,
- an ownership,
- and possibly a diamond (**case-idea**)



7/9

(Temporarily) Extend Signature: Associations

- Only for the course of Lectures 08/09 we assume that each attribute in \mathcal{V}
 - either is $(v : \tau, \xi, \text{exp}_0, P_1)$ with $\tau \in \mathcal{S}$ (as before)
 - or is an association of the form

$$(r : \langle \text{role}_1 : C_1, \mu_1, P_1, S_1, \nu_1, \alpha_1 \rangle, \dots, \langle \text{role}_n : C_n, \mu_n, P_n, S_n, \nu_n, \alpha_n \rangle)$$

- where
 - $n \geq 2$ (at least two ends),
 - r_i, role_i are just names, $C_i \in \mathcal{C}, 1 \leq i \leq n$,
 - μ_i is the multiplicity μ_i is an expression of the form $\mu_i ::= \mathbb{N} \mid \mathbb{N} \cdot * \mid \mathbb{N} \cdot * \mid \mu_i \mu$ ($\mathbb{N}, M \in \mathbb{N}$)
 - P_i is a set of properties (as before),
 - $S_i \in \{+, -, \# \}$ (as before),
 - $\nu_i \in \{x, \rightarrow, \sim\}$ is the navigability,
 - $\alpha_i \in \mathbb{B}$ is the ownership.

8/9

(Temporarily) Extend Signature: Associations

- Only for the course of Lectures 08/09 we assume that each attribute in \mathcal{V}
 - either is $(v : \tau, \xi, \text{exp}_0, P_1)$ with $\tau \in \mathcal{S}$ (as before)
 - or is an association of the form

$$(r : \langle \text{role}_1 : C_1, \mu_1, P_1, S_1, \nu_1, \alpha_1 \rangle, \dots, \langle \text{role}_n : C_n, \mu_n, P_n, S_n, \nu_n, \alpha_n \rangle)$$

- where
 - $n \geq 2$ (at least two ends),
 - r_i, role_i are just names, $C_i \in \mathcal{C}, 1 \leq i \leq n$,
 - μ_i is the multiplicity μ_i is an expression of the form $\mu_i ::= \mathbb{N} \mid \mathbb{N} \cdot * \mid \mathbb{N} \cdot * \mid \mu_i \mu$ ($\mathbb{N}, M \in \mathbb{N}$)
 - P_i is a set of properties (as before),
 - $S_i \in \{+, -, \# \}$ (as before),
 - $\nu_i \in \{x, \rightarrow, \sim\}$ is the navigability,
 - $\alpha_i \in \mathbb{B}$ is the ownership.

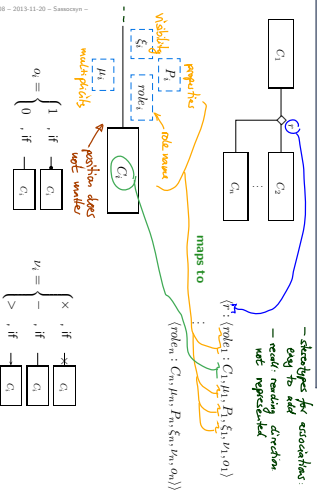
8/9

(Temporarily) Extend Signature: Basic Type Attributes

- Also only for the course of this lecture
 - we only consider **basic type attributes** to "belong" to a class (to appear in $\text{atr}(C)$).
 - associations are not "owned" by a particular class (do not appear in $\text{atr}(C)$), but live on their own.

Formally, we only call $(\mathcal{S}, \mathcal{C}, \mathcal{V}, \text{atr})$ a signature (extended for associations) if $\text{atr} : \mathcal{C} \rightarrow 2^{\mathcal{C} \times \mathcal{V} \times \mathcal{S}}$.

From Association Lines to Extended Signatures



Association Example



Signature: $\mathcal{S} = \{\text{true}, \&C, D\}, \{x, \rightarrow, \sim\}$
 $\langle r : \langle C, c, 0..*, \&C, \rightarrow, x, \text{true} \rangle, \langle D, d, 1..1, \&D, \rightarrow, x, \text{true} \rangle \rangle$
 {C} → {D} ← only basic type attributes here

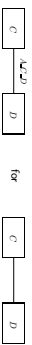
What If Things Are Missing?

Most components of associations or association end may be omitted. For instance [OMG, 2007b, 17], Section 6.4.2, proposes the following rules:

- **Name:** Use $A(C_1, \dots, C_n)$

if the name is missing.

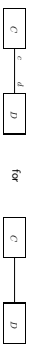
Example:



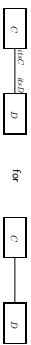
- **Reading Direction:** no default.

- **Role Name:** use the class name at that end in lower-case letters

Example:



Other convention: (used e.g. by modelling tool Rhapsody) $pc\>cv$ *class name* $cd\>dc$



12/9

What If Things Are Missing?

- **Multiplicity:** 1. In my opinion, it's safer to assume 0, 1 or * if there are no fixed written agreed conventions ("expect the worst").
- **Properties:** \emptyset
- **Visibility:** public
- **Navigability and Ownership:** not so easy. [OMG, 2007b, 43]

"Various options may be chosen for showing navigation arrows on a diagram. In practice, it is often convenient to suppress some of the arrows and crosses and just show exceptional situations:

- Show all arrows and \times 's. Navigation and its absence are made completely explicit.
- Suppress all arrows and \times 's. No inference can be drawn about navigation. This is similar to any situation in which information is suppressed from a view.
- Suppress arrows for associations with navigability in both directions, and show arrows only for associations with one-way navigability.
- In this case, the two-way navigability cannot be distinguished from situations where there is no navigation at all; however, the latter case occurs rarely in practice."

13/9

Wait, If Omitting Things...

- ...is causing so much trouble (e.g. leading to misunderstanding), why does the standard say "In practice, it is often convenient..."?
- Is it a good idea to trade convenience for precision/ambiguity?

It depends.

- Convenience as such is a legitimate goal.
- In UML-As-Sketch mode, precision "doesn't matter", so convenience (for writer) can even be a primary goal.

- In UML-As-Blueprint mode, precision is the primary goal. And misunderstandings are in most cases annoying.

But: (even in UML-As-Blueprint mode)

If all associations in your model have multiplicity \ast , then it's probably a good idea not to write all these \ast 's. So: tell the reader about it and leave out the \ast 's.

14/9

Rhapsody Demo

Association Semantics

- Overview**
- What's left? Named association with at least two typed ends, each having
- a role name,
 - a set of properties,
 - a multiplicity,
 - a visibility,
 - a navigability, and
 - an ownership.
- The Plan:**
- Extend system states, introduce so-called links as instances of associations — depends on name and on type and number of ends.
 - Integrate role name and multiplicity into OCL syntax/semantics.
 - Extend typing rules to care for visibility and navigability
 - Consider multiplicity also as part of the constraints set $Inv(CD)$.
 - Properties: for now assume $P_n = \{uniques\}$.
 - Properties (in general) and ownership: later.

References

49/96

References

- [Osterwech, 2006] Osterwech, 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.

50/96