Assoziation

1. \( i = 0 \) \( \rightarrow \) \( \emptyset \)

2. \( i \subseteq \) signature(extended for associations)

3. \( \{\} \rightarrow \{\} \) \( \}

From Association Linesto Extended Signatures

What Do We (Have to) Cover?

(Temporarily) Extend Signature: Associations
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 $A \langle C_1 \rangle \cdots \langle C_n \rangle$ if the name is missing. Example: $CD\langle C \rangle D\langle D \rangle$ for $CD$

• **Reading Direction**: nodefault.

• **RoleName**: use the classname at that end in lower-case letters. Example: $CD\langle cD \rangle$ for $CD$

Other conventions: (used e.g. by modelling tool Rhapsody) $CD\langle cD \rangle$ itsC itsD for $CD$

• **Multiplicity**: 1

In my opinion, it’s safe 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] "Variousoptions 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 x’s. Navigation and its absence are made completely explicit.
- Suppress all arrows and x’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 arrow 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."

Wait, If Omitting Things...

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

Association Semantics

Overview

What’s left?

Named association with at least two typed ends, each having

• a rolename,

• a multiplicity,

• a set of properties,

• a visibility,

• a navigability, and

• an ownership.

The Plan:

• Extend system states, introduce so-called links as instances of associations—dependson name and on type and number of ends.

• Integrate rolename and multiplicity into OCL syntax/semantics.

• Extend typing rules to care for visibility and navigability.

• Consider multiplicity also as part of the constraints set $\text{Inv}(\text{CD})$.

• Properties: for now assume $P_v = \{\text{unique} \}$.

• Properties (in general) and ownership: later.
Recall: We consider associations of the following form:

\[
\langle r: \langle \text{role} \ 1: C_1, \mu_1, P_1, \xi_1, \nu_1, o_1 \rangle, \ldots, \langle \text{role} \ n: C_n, \mu_n, P_n, \xi_n, \nu_n, o_n \rangle \rangle.
\]

Only these parts are relevant for extended system states:

\[
\langle r: \langle \text{role} \ 1: C_1, \mu_1, P_1, \xi_1, \nu_1, o_1 \rangle, \ldots, \langle \text{role} \ n: C_n, \mu_n, P_n, \xi_n, \nu_n, o_n \rangle \rangle.
\]

We assume \( P_1 = P_n = \{ \text{unique} \} \).

The UML standard thinks of associations as \( n \)-ary relations "living on their own" in a system state. That is, links (association instances) • do not belong (in general) to certain objects (in contrast to pointers, e.g.) • are "first-class citizens" next to objects, • are (in general) not directed (in contrast to pointers).

Reynolds's approach aims to represent extended state transitions as

\[
\text{Definition.}
\]

Let \( \mathcal{C} \) be a structure of the (extended) signature \( \mathcal{C} \mathcal{B} = (\mathcal{C} \mathcal{C}, \mathcal{B} \mathcal{V}, \mathcal{A} \mathcal{t}) \).

A system state of \( \mathcal{C} \mathcal{B} \) wrt. \( \mathcal{C} \mathcal{B} \) is a pair \((\sigma, \lambda)\) consisting of

• a type-consistent mapping \( \sigma: \mathcal{C} \mathcal{B}(\mathcal{B} \mathcal{V}) \rightarrow \mathcal{C} \mathcal{B}(\mathcal{C} \mathcal{C}) \),
• a mapping \( \lambda \) which assigns each association \( \langle r: \langle \text{role} \ 1: C_1, \mu_1, P_1, \xi_1, \nu_1, o_1 \rangle, \ldots, \langle \text{role} \ n: C_n, \mu_n, P_n, \xi_n, \nu_n, o_n \rangle \rangle \in \mathcal{B} \mathcal{V} \) a relation \( \lambda(r) \subseteq \mathcal{C} \mathcal{B}(C_1) \times \cdots \times \mathcal{C} \mathcal{B}(C_n) \) (i.e. a set of type-consistent \( n \)-tuples of identities).

Association/Link Example

\[
\begin{align*}
\mathcal{C} & \rightarrow \text{Int} \\
\mathcal{D} & \rightarrow \text{Int} \\
\mathcal{x} & : \text{Int} \\
A & : \text{C} \\
C & : \text{D} \\
\end{align*}
\]

Signature:

\[
\mathcal{C} \mathcal{B} = (\{\text{Int}\}, \{\text{C}, \text{D}\}, \{\mathcal{x}: \text{Int}, \langle A \mathcal{C} : \langle \mathcal{c}: \text{C}, 0 \ldots *, +, \{\text{unique}\}, \times, 1 \rangle, \langle n : \text{D}, 0 \ldots *, +, \{\text{unique}\}, >, 0 \rangle \rangle \})
\]

A system state of \( \mathcal{C} \mathcal{B} \) (somewhere reasonable \( \mathcal{C} \mathcal{B} \)) is \((\sigma, \lambda)\) with:

\[
\begin{align*}
\sigma & = \{1 \mathcal{C} \rightarrow \emptyset, 3 \mathcal{D} \rightarrow \{x \rightarrow 1\}, 7 \mathcal{D} \rightarrow \{x \rightarrow 2\}\} \\
\lambda & = \{A \mathcal{C} : \langle (1 \mathcal{C}, 3 \mathcal{D}), (1 \mathcal{C}, 7 \mathcal{D}) \rangle\} \\
\end{align*}
\]

Association Example

Extended system states and object diagrams. A legitimate question: how do we represent system states such as

\[
\begin{align*}
\sigma & = \{1 \mathcal{C} \rightarrow \emptyset, 3 \mathcal{D} \rightarrow \{x \rightarrow 1\}, 7 \mathcal{D} \rightarrow \{x \rightarrow 2\}\} \\
\lambda & = \{A \mathcal{C} : \langle (1 \mathcal{C}, 3 \mathcal{D}), (1 \mathcal{C}, 7 \mathcal{D}) \rangle\} \\
\end{align*}
\]

as object diagrams?
