Software Design, Modelling and Analysis in UML

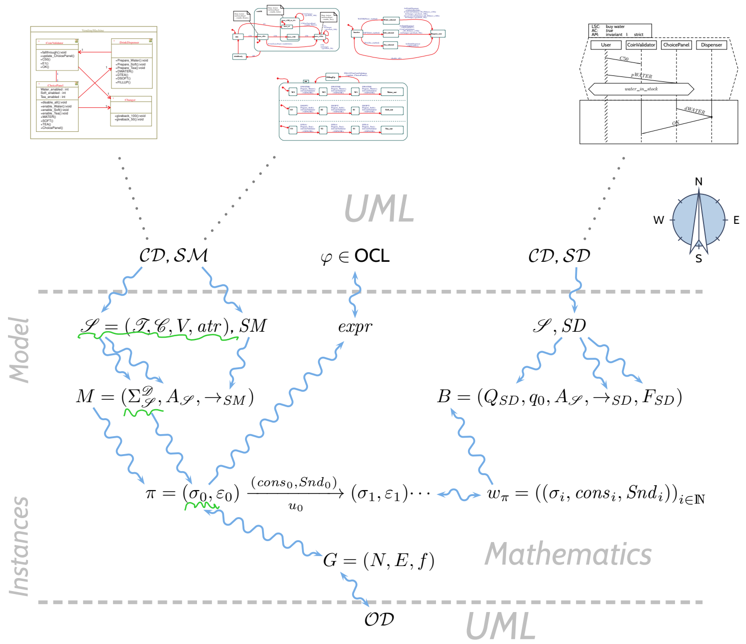
Lecture 2: Semantical Model

2016-10-20

Prof. Dr. Andreas Podelski, **Dr.** ~~Bernd Westphal~~ Jochen Hoenicke

Albert-Ludwigs-Universität Freiburg, Germany

Course Map



Content

- **Basic Object System Signature**
  - (basic) types, classes,
  - typed attributes,
  - attribute mapping.
- **Basic Object System Structure**
  - objects / object identities,
  - domains of basic and derived types.
- **System State**
  - concrete and symbolic,
  - dangling references,
- **A Complete Example**

Semantical Foundation

Basic Object System Signature

Definition. A (Basic) 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,
- V is a finite set of typed attributes, i.e., each $v \in V$ has a type
  - $\tau \in \mathscr{T}$, or
  - $C_{0,1}$ or C_*, where $C \in \mathscr{C}$

  (written $v : \tau$ or $v : C_{0,1}$ or $v : C_*$),
- $atr : \mathscr{C} \to 2^V$ maps each class to its set of attributes.

Note: Inspired by OCL 2.0 standard OMG (2006), Annex A.

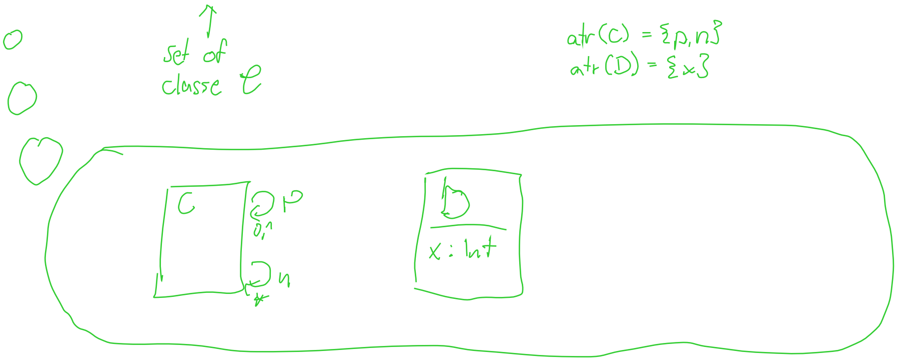
Basic Object System Signature Example

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

$$\mathscr{S}_0 = (\{Int\}, \{C, D\}, \{x : Int, p : C_{0,1}, n : C_*\}, \{C \mapsto \{p, n\}, D \mapsto \{x\}\})$$



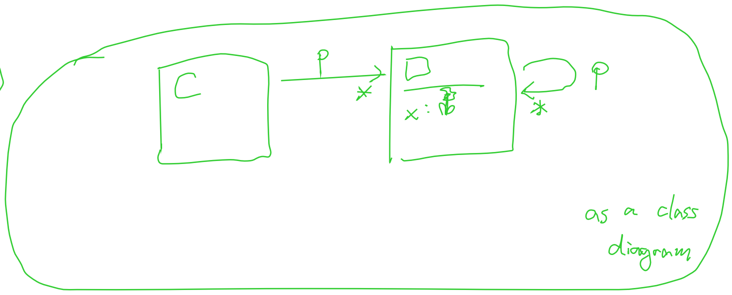
Basic Object System Signature Another Example

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

$\mathscr{S}_1 = (\{✿, MyType\}, \{C, D\}, \{x : ✿, p : D_*, q : D_{0,1}\},$
$\{C \mapsto \{p\}, D \mapsto \{x, p\}\})$



Basic Object System Structure

Definition. A Basic Object System Structure of $\mathscr{S} = (\mathscr{T}, \mathscr{C}, V, atr)$ is a domain function $\mathscr{D}$ which assigns to each type a domain, i.e.

- $\tau \in \mathscr{T}$ is mapped to $\mathscr{D}(\tau)$, $\mathscr{D}(\tau)$ is a set
- $C \in \mathscr{C}$ is mapped to an infinite set $\mathscr{D}(C)$ of (object) identities.

  Note: Object identities only have the "=" operation.
- Sets of object identities for different classes are disjoint, i.e.

$$\forall C, D \in \mathscr{C} : C \neq D \to \mathscr{D}(C) \cap \mathscr{D}(D) = \emptyset.$$

- C_* and $C_{0,1}$ for $C \in \mathscr{C}$ are mapped to $2^{\mathscr{D}(C)}$.

  $\mathscr{D}(C_*) := 2^{\mathscr{D}(C)}$, $\mathscr{D}(C_{0,1}) := 2^{\mathscr{D}(C)}$

We use $\mathscr{D}(\mathscr{C})$ to denote $\bigcup_{C \in \mathscr{C}} \mathscr{D}(C)$; analogously $\mathscr{D}(\mathscr{C}_*)$.

$\mathscr{D}(\mathscr{C}_*) = \bigcup_{C \in \mathscr{C}} \mathscr{D}(C_*)$

Note: We identify objects and object identities, because both uniquely determine each other (cf. OCL 2.0 standard).

Basic Object System Structure Example

Wanted: a structure for signature

$$\mathscr{S}_0 = (\{Int\}, \{C, D\}, \{x : Int, p : C_{0,1}, n : C_*\}, \{C \mapsto \{p, n\}, D \mapsto \{x\}\})$$

> $\mathscr{D}$ needs to map:
>
> - $\tau \in \mathscr{T}$ to **some** $\mathscr{D}(\tau)$,
> - $C \in \mathscr{C}$ to **some** set of identities $\mathscr{D}(C)$ (infinite, disjoint for different classes),
> - C_* and $C_{0,1}$ for $C \in \mathscr{C}$: always mapped to $\mathscr{D}(C_*) = \mathscr{D}(C_{0,1}) = 2^{\mathscr{D}(C)}$.

$$\begin{aligned}
\mathscr{D}(Int) &= \mathbb{Z} \\
\mathscr{D}(C) &= \mathbb{N}^+ \times \{C\} \cong \{1_C, 2_C, 3_C, 4_C, \dots\} \\
\mathscr{D}(D) &= \mathbb{N}^+ \times \{D\} \cong \{1_D, 2_D, 3_D, \dots\} \\
\mathscr{D}(C_{0,1}) = \mathscr{D}(C_*) &= 2^{\mathscr{D}(C)} \\
\mathscr{D}(D_{0,1}) = \mathscr{D}(D_*) &= 2^{\mathscr{D}(D)}
\end{aligned}$$

$\mathscr{D}(Int) = \{-2, -1, 0, 1, 2\}$
$\mathscr{D}(C) = \{a, aa, aaa, \dots\}$
$\mathscr{D}(D) = \{b, bb, bbb, \dots\}$

System State

> **Definition.** Let $\mathscr{D}$ be a structure of $\mathscr{S} = (\mathscr{T}, \mathscr{C}, V, atr)$.
>
> A system state of $\mathscr{S}$ wrt. $\mathscr{D}$ is a type-consistent mapping
>
> (set of all class identies)
>
> $$\sigma : \mathscr{D}(\mathscr{C}) \nrightarrow (V \nrightarrow (\mathscr{D}(\mathscr{T}) \cup \mathscr{D}(\mathscr{C}_*))).$$
>
> (partial fnc.) (union of all domains $\mathscr{D}(\tau)$ and all subset of object identifies)
>
> That is, for each $u \in \mathscr{D}(C)$, $C \in \mathscr{C}$, if $u \in \text{dom}(\sigma)$
>
> - $\text{dom}(\sigma(u)) = atr(C)$
> - $\sigma(u)(v) \in \mathscr{D}(\tau)$ if $v : \tau, \tau \in \mathscr{T}$
> - $\sigma(u)(v) \in \mathscr{D}(D_*)$ if $v : D_{0,1}$ or $v : D_*$ with $D \in \mathscr{C}$
>
> (type-consistent)
>
> We call $u \in \mathscr{D}(\mathscr{C})$ alive in σ if and only if $u \in \text{dom}(\sigma)$.
>
> We use $\Sigma_{\mathscr{S}}^{\mathscr{D}}$ to denote the set of all system states of $\mathscr{S}$ wrt. $\mathscr{D}$.

System State Example

$$\mathscr{S}_0 = (\{Int\}, \{C, D\}, \{x : Int, p : C_{0,1}, n : C_*\}, \{C \mapsto \{p, n\}, D \mapsto \{x\}\})$$

$$\mathscr{D}(Int) = \mathbb{Z}, \quad \mathscr{D}(C) = \{1_C, 2_C, 3_C, ...\}, \quad \mathscr{D}(D) = \{1_D, 2_D, 3_D, ...\}$$

Wanted: $\sigma : \mathscr{D}(\mathscr{C}) \nrightarrow (V \nrightarrow (\mathscr{D}(\mathscr{T}) \cup \mathscr{D}(\mathscr{C}_*)))$ such that (i) $\mathrm{dom}(\sigma(u)) = atr(C)$, and (ii) $\sigma(u)(v) \in \mathscr{D}(\tau)$ if $v : \tau, \tau \in \mathscr{T}$, (iii) $\sigma(u)(v) \in \mathscr{D}(C_*)$ if $v : D_*$ with $D \in \mathscr{C}$.

$\sigma_0 = \emptyset$ ("empty function")

alive in σ_0: none

$\sigma_1 = \{ 1_C \mapsto \{ p \mapsto \emptyset, n \mapsto \{1_C, 5_C\} \}, 5_C \mapsto \{ p \mapsto \{1_C\}, n \mapsto \emptyset \}, 3_D \mapsto \{x \mapsto 3\} \}$

$\sigma_2 = \{ 1_D \mapsto \{x \mapsto 27\}, 2_D \mapsto \{x \mapsto 27\}, 17_D \mapsto \{x \mapsto 0\} \}$

System State Example

$$\mathscr{S}_0 = (\{Int\}, \{C, D\}, \{x : Int, p : C_{0,1}, n : C_*\}, \{C \mapsto \{p, n\}, D \mapsto \{x\}\})$$

$$\mathscr{D}(Int) = \mathbb{Z}, \quad \mathscr{D}(C) = \{1_C, 2_C, 3_C, ...\}, \quad \mathscr{D}(D) = \{1_D, 2_D, 3_D, ...\}$$

Wanted: $\sigma : \mathscr{D}(\mathscr{C}) \nrightarrow (V \nrightarrow (\mathscr{D}(\mathscr{T}) \cup \mathscr{D}(\mathscr{C}_*)))$ such that (i) $\mathrm{dom}(\sigma(u)) = atr(C)$, and (ii) $\sigma(u)(v) \in \mathscr{D}(\tau)$ if $v : \tau, \tau \in \mathscr{T}$, (iii) $\sigma(u)(v) \in \mathscr{D}(C_*)$ if $v : D_*$ with $D \in \mathscr{C}$.

Two options:

- **Concrete, explicit** identities:

$$\sigma = \{1_C \mapsto \{p \mapsto \emptyset, n \mapsto \{5_C\}\}, 5_C \mapsto \{p \mapsto \emptyset, n \mapsto \emptyset\}, 1_D \mapsto \{x \mapsto 23\}\}.$$

$1_C \mapsto \emptyset$

- **Alternative**: **symbolic** system state.

$$\sigma = \{c_1 \mapsto \{p \mapsto \emptyset, n \mapsto \{c_2\}\}, c_2 \mapsto \{p \mapsto \emptyset, n \mapsto \emptyset\}, d \mapsto \{x \mapsto 23\}\}$$

assuming $c_1, c_2 \in \mathscr{D}(C), d \in \mathscr{D}(D), c_1 \neq c_2$.

System State: Spot the 10 (?) Mistakes

$$\mathscr{S}_0 = (\{Int\}, \{C, D\}, \{x : Int, p : C_{0,1}, n : C_*\}, \{C \mapsto \{p, n\}, D \mapsto \{x\}\})$$
$$\mathscr{D}(Int) = \mathbb{Z}, \quad \mathscr{D}(C) = \{1_C, 2_C, 3_C, ...\}, \quad \mathscr{D}(D) = \{1_D, 2_D, 3_D, ...\}$$

Wanted: $\sigma : \mathscr{D}(\mathscr{C}) \nrightarrow (V \nrightarrow (\mathscr{D}(\mathscr{T}) \cup \mathscr{D}(\mathscr{C}_*)))$ such that (i) $\mathrm{dom}(\sigma(u)) = atr(C)$, and
(ii) $\sigma(u)(v) \in \mathscr{D}(\tau)$ if $v : \tau, \tau \in \mathscr{T}$, (iii) $\sigma(u)(v) \in \mathscr{D}(C_*)$ if $v : D_*$ with $D \in \mathscr{C}$.

- $\sigma = \{1_C \mapsto \{p \mapsto \emptyset, n \mapsto \{5_C\}\}, \quad 5_C \mapsto \{p \mapsto \emptyset, n \mapsto 1_C\}, \quad 1_D \mapsto \{x \mapsto 2.3\}\}$.
  (iii) $\notin \mathscr{D}(C_*)$ (ii) $\notin \mathscr{D}(Int)$
- $\sigma = \{1_C \mapsto \{p \mapsto \emptyset, n \mapsto \{5_C\}\}, \quad 5_C \mapsto \{p \mapsto 1_C, n \mapsto \emptyset\}, \quad 1_D \mapsto \{x \mapsto 23\}\}$.
- $\sigma = \{1_C \mapsto \{p \mapsto \emptyset, n \mapsto \{1_D\}\}, \quad 5_C \mapsto \{p \mapsto \emptyset, n \mapsto \emptyset\}, \quad 1_D \mapsto \{x \mapsto 22\}\}$.
  (iii) $\notin \mathscr{D}(C_*)$
- $\sigma = \{1_C \mapsto \{p \mapsto \emptyset, n \mapsto \{5_C\}\}, \quad 5_C \mapsto \{n \mapsto \emptyset\}, \quad 1_D \mapsto \{x \mapsto 1, p \mapsto \{1_C\}\}\}$.
  (i) $p \notin \mathrm{dom}\,\sigma(5_C)$
- $\sigma = \{1_C \mapsto \{p \mapsto \emptyset, n \mapsto \{5_C\}\}, \quad 5_C \mapsto \{p \mapsto \emptyset, n \mapsto \{9_C\}\}\}$
  (iii): $\{9_C\} \in \mathscr{D}(C_*)$

Dangling References

Definition. Let $\sigma \in \Sigma_{\mathscr{S}}^{\mathscr{D}}$ be a system state.
We say attribute $v \in V_{0,1,*}$, i.e. $v : C_{0,1}$ or $v : C_*$, in object $u \in \mathrm{dom}(\sigma)$ has a dangling reference if and only if the attribute's value comprises an object which is not alive in σ, i.e. if

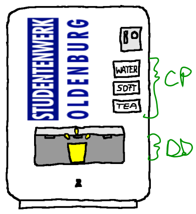
$$\sigma(u)(v) \not\subseteq \mathrm{dom}(\sigma).$$

We call σ closed if and only if no attribute has a dangling reference in any object alive in σ.

Example:

- $\sigma = \{1_C \mapsto \{p \mapsto \emptyset, n \mapsto \{5_C\}\}\}$

A Complete Example: Vending Machine Structure



$$\mathcal{S} = (\{Int, Bool\}, \{VM, CP, DD\}, \{cp : CP_*, dd : DD_{0,1}, win : Int, wen : Bool\}, \{VM \mapsto \{cp, dd\}, CP \mapsto \{wen\}, DD \mapsto \{win\}\})$$

$\mathcal{D}(Bool) = \{true, false\}$ $\mathcal{D}(Int) = \mathbb{Z}$

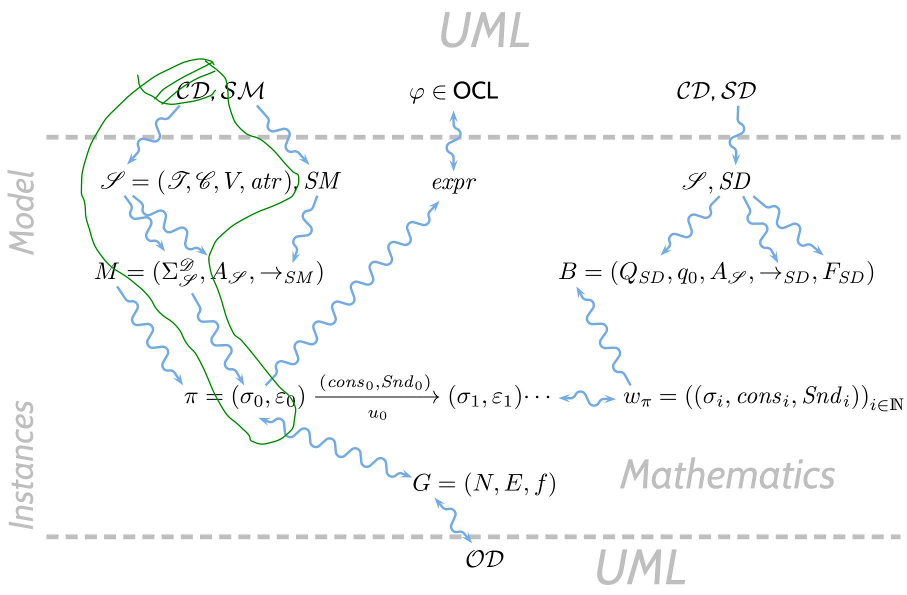
$\mathcal{D}(VM) = \{1_{VM}, 2_{VM}, \dots\}$

$\mathcal{D}(DD) = , \dots$

$$\sigma = \{1_{VM} \mapsto \{dd \mapsto \{1_{DD}\}, cp \mapsto \{42_{CP}, 3_{CP}\}\}, 1_{DD} = \{win \mapsto 10\}, 42_{CP} = \{wen \mapsto true\}, 3_{CP} = \{wen \mapsto false\}\}$$

You Are Here.

Course Map



Tell Them What You've Told Them...

- We can directly use **object system signatures** to model the structure of systems.

  → We don't **need** diagrams, they will be more pleesent to read.
- We introduce
  - **basic types** and **classes**,
  - basic type and derived type **attributes**, and
  - assign to each class a set of attributes.
- **Object system structures** provide domains for basic and derived types.
- An **object system signature** $\mathscr{S}$ and an **object system structure** $\mathscr{D}$ uniquely define the set $\Sigma_{\mathscr{S}}^{\mathscr{D}}$ of **system states**.
- **Outlook**:
  - **Object system signatures** will be used to capture the **abstract syntax** of class diagrams.
  - **OCL expressions** will be evaluated on **system states**.
  - **State machines** will define sequences of **system configurations** (consisting of a **system state** and an **ether**).

References

References

OMG (2006). Object Constraint Language, version 2.0. Technical Report formal/06-05-01.

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.