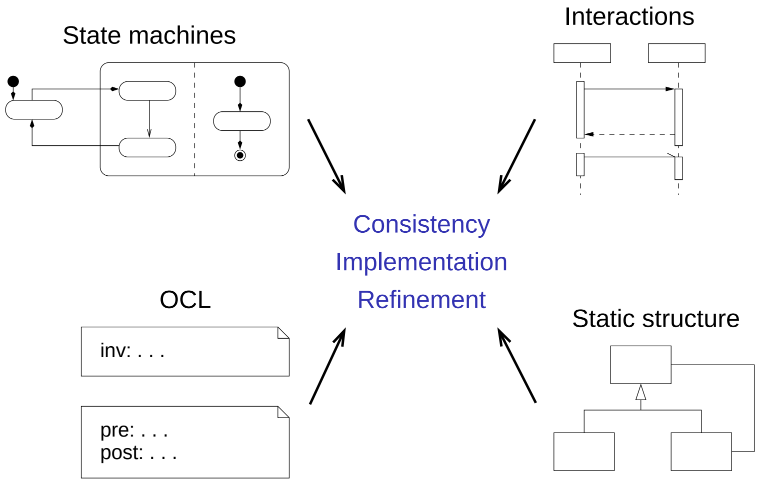
UML 2.0 Interactions
Semantics and Refinement

María Victoria Cengarle
Technische Universität München

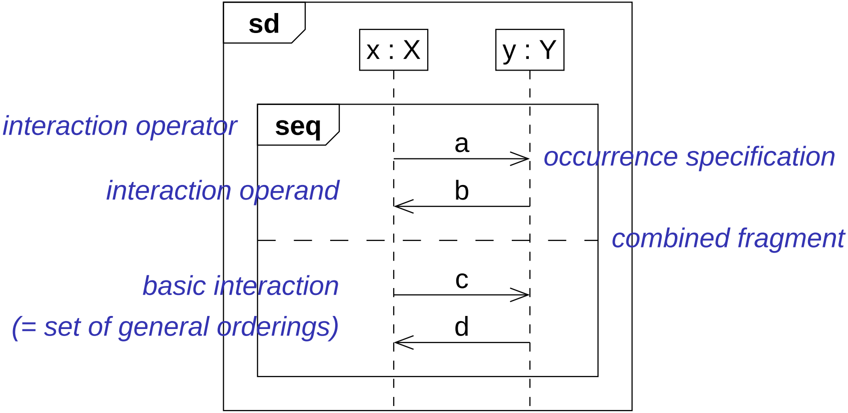
Alexander Knapp
Ludwig-Maximilians-Universität München

Formal Semantics for UML 2.0

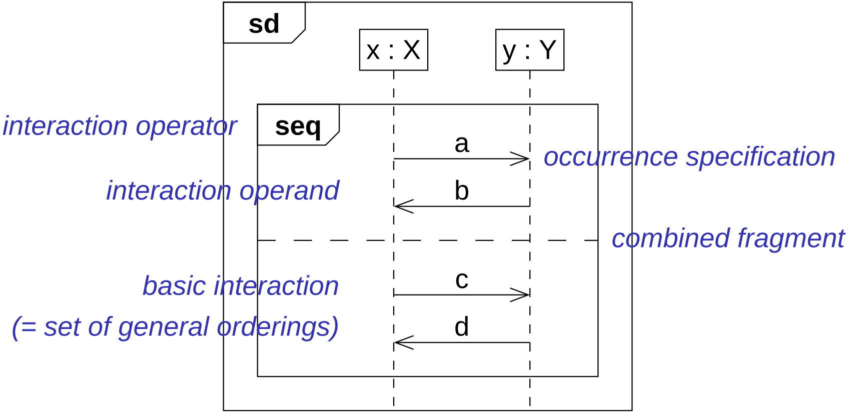


- **Heterogenous approach** formalise different diagram types in appropriate domains

UML 2.0 Interactions



UML 2.0 Interactions



Semantics positive, negative, inconclusive traces

Semantics of Positive Fragment

- based on pomsets

$$
\begin{array}{ll}
t \models_{\mathrm{p}} B & \text{if } t \in \mathit{lin}(B) \\[1em]
t \models_{\mathrm{p}} \mathsf{strict}(S_1, S_2) & \text{if } \exists t_1, t_2 \,.\, t \in \mathit{lin}(t_1 \,;\, t_2) \;\wedge t_1 \models_{\mathrm{p}} S_1 \wedge t_2 \models_{\mathrm{p}} S_2 \\
t \models_{\mathrm{p}} \mathsf{seq}(S_1, S_2) & \text{if } \exists t_1, t_2 \,.\, t \in \mathit{lin}(t_1 \,;_{\asymp}\, t_2) \wedge t_1 \models_{\mathrm{p}} S_1 \wedge t_2 \models_{\mathrm{p}} S_2 \\
t \models_{\mathrm{p}} \mathsf{par}(S_1, S_2) & \text{if } \exists t_1, t_2 \,.\, t \in \mathit{lin}(t_1 \parallel t_2) \;\wedge t_1 \models_{\mathrm{p}} S_1 \wedge t_2 \models_{\mathrm{p}} S_2 \\[1em]
t \models_{\mathrm{p}} \mathsf{alt}(S_1, S_2) & \text{if } t \models_{\mathrm{p}} S_1 \vee t \models_{\mathrm{p}} S_2
\end{array}
$$

- same as in Störrle's and the STAIRS approach
- similarly for other interaction operators
  (opt, loop, ignore, consider, break, critical)

Semantics of Negation: Positive Satisfaction

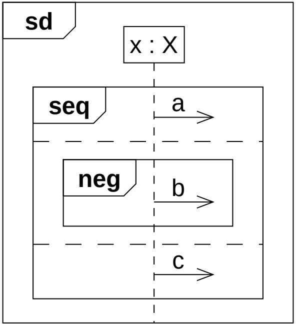
$$t \models_{\mathrm{p}} \mathsf{neg}(S) \quad \text{if } t = \varepsilon$$

$$t \models_{\mathrm{p}} \mathsf{assert}(S) \quad \text{if } t \models_{\mathrm{p}} S$$

- same as in the STAIRS approach
- for every interaction there is a positive trace

Semantics of Negation: Negative Satisfaction

$$t \models_{\mathrm{n}} \mathsf{seq}(S_1, S_2)$$
$$\text{if } \exists t_1, t_2 \,.\, t \in \mathit{lin}(t_1 \,;_{\lessgtr}\, t_2) \wedge ((t_1 \models_{\mathrm{n}} S_1) \vee (t_1 \models_{\mathrm{p}} S_1 \wedge t_2 \models_{\mathrm{n}} S_2))$$

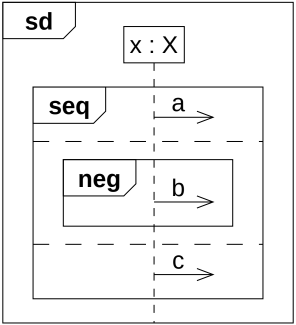


Negative traces

- $\mathsf{a} \cdot \mathsf{b} \cdot t$ for all $t \in \mathbb{T}$

Semantics of Negation: Negative Satisfaction

$$t \models_{\mathrm{n}} \mathsf{seq}(S_1, S_2)$$
$$\text{if } \exists t_1, t_2 \,.\, t \in \mathit{lin}(t_1 \,;_{\lessgtr}\, t_2) \wedge ((t_1 \models_{\mathrm{n}} S_1) \wedge (t_2 \models_{\mathrm{p}} S_2 \vee t_2 \models_{\mathrm{n}} S_2)) \vee (t_1 \models_{\mathrm{p}} S_1 \wedge t_2 \models_{\mathrm{n}} S_2))$$

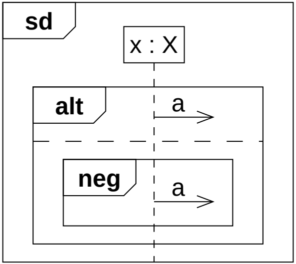


Negative traces (STAIRS)

- a · b · t for all $t \in \mathbb{T}$
- a · b · c

Semantics of Negation: Negative Satisfaction

$$t \models_{\mathrm{n}} \mathsf{alt}(S_1, S_2) \quad \text{if } t \models_{\mathrm{n}} S_1 \wedge t \models_{\mathrm{n}} S_2$$

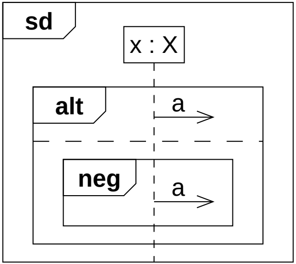


Negative traces

- none

Semantics of Negation: Negative Satisfaction

$$t \models_{\mathrm{n}} \mathsf{alt}(S_1, S_2) \quad \text{if } t \models_{\mathrm{n}} S_1 \vee t \models_{\mathrm{n}} S_2$$



Negative traces (STAIRS)

- none
- a

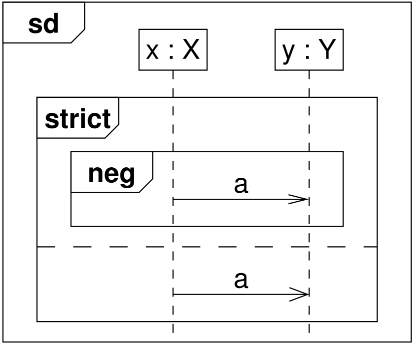
Semantic Properties

- There are inconclusive traces
- strict(–, –) and seq(–, –) associative
- par(–, –) and alt(–, –) associative and commutative

Semantic Properties

- There are inconclusive traces
- strict(–, –) and seq(–, –) associative
- par(–, –) and alt(–, –) associative and commutative

Overspecified interactions $\exists t \in \mathbb{T} \,.\, t \models_{\mathrm{p}} S \wedge t \models_{\mathrm{n}} S$



Reduction to Positive Satisfaction

Replace $\mathsf{neg}(-)$ by Empty $\qquad \sigma(S)$

- e.g. $\sigma(\mathsf{neg}(S)) = \mathsf{Empty}$
- $t \models_{\mathrm{p}} S$ iff $t \models_{\mathrm{p}} \sigma(S)$

Reduction to Positive Satisfaction

Replace $\mathsf{neg}(-)$ by Empty $\qquad \sigma(S)$

- e.g. $\sigma(\mathsf{neg}(S)) = \mathsf{Empty}$
- $t \models_{\mathrm{p}} S$ iff $t \models_{\mathrm{p}} \sigma(S)$

Introduce classical negation $\quad \mathsf{not}(-)$

- $t \models_{\mathrm{p}} \mathsf{not}(S)$ if $t \not\models_{\mathrm{p}} S$
- derived operators $\quad \mathsf{and}(-,-)$, Any, None

Reduction to Positive Satisfaction

Replace neg($-$) by Empty $\quad \sigma(S)$

- e.g. $\sigma(\text{neg}(S)) = \text{Empty}$
- $t \models_{\mathrm{p}} S \quad$ iff $\quad t \models_{\mathrm{p}} \sigma(S)$

Introduce classical negation $\quad$ not($-$)

- $t \models_{\mathrm{p}} \text{not}(S) \quad$ if $\quad t \not\models_{\mathrm{p}} S$
- derived operators $\quad$ and($-$, $-$), Any, None

Express negative satisfaction by positive satisfaction $\quad \nu(S)$

- e.g. $\nu(\text{strict}(S_1, S_2)) = \text{alt}(\text{strict}(\nu(S_1), \text{Any}), \text{strict}(\sigma(S_1), \nu(S_2)))$
- $t \models_{\mathrm{n}} S \quad$ iff $\quad t \models_{\mathrm{p}} \nu(S)$

Implementation of UML 2.0 Interactions

Process I implements interaction S $I \models S$

- $\exists t \in lin(I) \,.\, t \models_{\mathrm{p}} S$
- $\forall t \in lin(I) \,.\, t \not\models_{\mathrm{n}} S$

In UML 2.0 e.g., cooperating state machines

Implementation of UML 2.0 Interactions

Process I implements interaction S $I \models S$

- $\exists t \in lin(I) \,.\, t \models_{\mathrm{p}} S$
- $\forall t \in lin(I) \,.\, t \not\models_{\mathrm{n}} S$

In UML 2.0 e.g., cooperating state machines

- Unimplementable strict(neg(B), B)
- Overspecified interactions may be implementable

  alt(strict(neg(B), B), strict(neg(B), B'))

Refinement of UML 2.0 Interactions

Model-based notion of refinement

Interaction S' refines interaction S $\qquad S \rightsquigarrow S'$

- $\forall I \in \mathbb{I} \,.\, I \models S' \Rightarrow I \models S$

Refinement of UML 2.0 Interactions

Model-based notion of refinement

Interaction S' refines interaction S $\quad S \rightsquigarrow S'$

- $\forall I \in \mathbb{I} \,.\, I \models S' \Rightarrow I \models S$

Example $\quad \mathrm{alt}(S_1, S_2) \rightsquigarrow S_1$ and $\mathrm{alt}(S_1, S_2) \rightsquigarrow S_2$

Refinement of UML 2.0 Interactions

Model-based notion of refinement

Interaction S' refines interaction S $\qquad S \leadsto S'$

- $\forall I \in \mathbb{I} \,.\, I \models S' \Rightarrow I \models S$

Example $\quad \mathrm{alt}(S_1, S_2) \leadsto S_1$ and $\mathrm{alt}(S_1, S_2) \leadsto S_2$

Properties of $S \leadsto S'$

- Less positive traces

  $$\forall t \in \mathbb{T} \,.\, t \not\models_{\mathrm{p}} S \vee t \models_{\mathrm{n}} S \Rightarrow t \not\models_{\mathrm{p}} S' \vee t \models_{\mathrm{n}} S'$$

- More negative traces if S' implementable

  $$\forall t \in \mathbb{T} \,.\, t \models_{\mathrm{n}} S \Rightarrow t \models_{\mathrm{n}} S'$$

- Keeping a positive trace if S' implementable

  $$\exists t \in \mathbb{T} \,.\, t \models_{\mathrm{p}} S \wedge t \models_{\mathrm{p}} S'$$

Comparison to STAIRS Approach

Interaction S' STAIRS-refines interaction S $S \succeq S'$

- $t \models_{\mathrm{p}} S \Rightarrow t \models_{\mathrm{p}} S' \lor t \models_{\mathrm{n}} S'$
- $t \models_{\mathrm{n}} S \Rightarrow t \models_{\mathrm{n}} S'$

- encompasses “narrowing” and “supplementing”

Comparison to STAIRS Approach

Interaction S' STAIRS-refines interaction S $S \succeq S'$

- $t \models_p S \Rightarrow t \models_p S' \vee t \models_n S'$
- $t \models_n S \Rightarrow t \models_n S'$

- encompasses “narrowing” and “supplementing”

In general, $\rightsquigarrow$ and $\succeq$ incomparable

- $S \succeq \mathrm{neg}(S)$ but $S \not\rightsquigarrow \mathrm{neg}(S)$

Comparison to STAIRS Approach

Interaction S' STAIRS-refines interaction S $S \succeq S'$

- $t \models_{\mathrm{p}} S \Rightarrow t \models_{\mathrm{p}} S' \lor t \models_{\mathrm{n}} S'$
- $t \models_{\mathrm{n}} S \Rightarrow t \models_{\mathrm{n}} S'$

- encompasses “narrowing” and “supplementing”

In general, $\leadsto$ and $\succeq$ incomparable

- $S \succeq \mathrm{neg}(S)$ but $S \not\leadsto \mathrm{neg}(S)$
- $\mathrm{par}(S, S') \not\succeq \mathrm{strict}(S, S')$ but $\mathrm{par}(S, S') \leadsto \mathrm{strict}(S, S')$

Ongoing Work

- Compositionality of refinement

$$\frac{S_1 \rightsquigarrow_{\mathrm{p}} S_1'}{\mathsf{seq}(S_1, S_2) \rightsquigarrow_{\mathrm{p}} \mathsf{seq}(S_1', S_2)} \qquad \frac{S_2 \rightsquigarrow S_2'}{\mathsf{seq}(S_1, S_2) \rightsquigarrow \mathsf{seq}(S_1, S_2')}$$

- Operational semantics
  - Sound and complete global operational semantics
  - Process-oriented, local operational semantics

$$(\mathsf{alt}^1_{\mathrm{p}}) \ l_f : \mathsf{alt}_p(S_1, S_2) \xrightarrow{\tau}_{\mathrm{p}} S_1$$
$$\text{if } 2 \notin h(p,f) \wedge h' = h[(p,f) \mapsto \{1\}]$$
$$(\mathsf{alt}^2_{\mathrm{p}}) \ l_f : \mathsf{alt}_p(S_1, S_2) \xrightarrow{\tau}_{\mathrm{p}} S_2$$
$$\text{if } 1 \notin h(p,f) \wedge h' = h[(p,f) \mapsto \{2\}]$$

- Equational theory

Conclusions and Outlook

- Semantics of UML 2.0 interactions
- Implementation and refinement relations

Conclusions and Outlook

- Semantics of UML 2.0 interactions
- Implementation and refinement relations
- Implementation relation in UML 2.0
  - based on, e.g., one of the formal semantics for state machines
  - points of interaction: event occurrences
- Relation to OCL(/RT)
  - points of interaction: message specifications