A structured operational semantics for UML-statecharts

Shadi Al-Dehni
LMU München

Jan Jürjens
TU München
We will discuss in this Seminar the following concepts:

- UML Diagrams (Unified Modeling Language).
- UML- Statecharts
- Syntax of UML-statechart
- Configurations on UML-statechart diagram
A structured operational semantics for UML-statecharts

We will discuss in this Seminar the following concepts:

- Semantics definition.
- Entry & exit action Semantics.
- Next state semantics.
- SOS rules of the auxiliary semantics.
- Case study will carry on during the whole Seminar.
**UML Diagrams**

- **A use case Diagram**: Use cases with actors and their relationships.
- **Interaction Diagram**: Objects and their relationships including the message, that may be dispatched among them. → (sequence Diagram, collaboration Diagram)
- **Activity Diagram**: Flow from activity to activity within a system.
- **Statechart Diagram**: State machine, (states, transitions, events, and activities).
UML Diagrams

- **Class Diagram**: Classes, interfaces, collaborations and their relationships.
- **Object Diagram**: Objects and their relationships.
- **Component Diagram**: Organizations and dependencies among a set of components.
- **Deployment Diagram**: Configuration of run-time processing nodes and the components that live on them.
A statechart diagram A state machine, emphasizing the flow of control from state to state. Which is a behavior that specifies the sequences of states an object goes through during its lifetime in response to events, together with its responses to those events.

A state: Condition or situation in the life of an object, satisfies some condition, performs some activity, or waits for some event.
An event: Occurrence of a stimulus that can trigger a state transition.

A transition: Relationship between two states where an object in the first state will perform certain actions and enter the second state when a specified event occurs and specified conditions are satisfied.

An activity: Ongoing nonatomic execution within a state machine.

An action: Executable atomic computation that results in a change in state of the model or the return of a value.
Syntax of UML statechart

- Basic term.
- Or term.
- And term.
- Case study.
- Configurations.
Syntax of UML statechart

Basic term.

\[ \mathbb{N} \text{ set of state names, } \mathbb{T} \text{ set of transition names, } \mathcal{I} \text{ set of events, } \mathcal{A} \text{ set of actions.} \]

\[ a, b, c, ... \text{ events or actions. } \alpha, \beta, \gamma \text{ sequence of events or sequence of actions} \]

\[ e_n, e_x \in \mathcal{A}, n \in \mathbb{N} \]

\[ s = [n, (e_n, e_x)] \]

UML-statechart term with type(s) = basic.
Syntax of UML statechart

Or term.

\[ s_1, \ldots, s_k \text{ UML-statechart terms} \quad \rho = \{1, \ldots, k\}, \quad l \in \rho, \]

\[ HT = \{\text{none, deep, shallow}\} \]

\[ TR =_{df} T \times \rho \times 2 \times \prod \times A^* \times 2 \times \rho \times HT \]

\[ s = [n, (s_1, \ldots, s_k), l, T, (en, ex)] \]

UML-statechart term with type(s) = or

\[ t = (t, i, sr, e, \alpha, td, j, ht) \]

sr \( \in \) conf(s_i), td \( \in \) conf(s_j),

\[ \text{sou}(t) =_{df} s_i, \quad \text{tar}(t) =_{df} s_j, \quad \text{historyType}(t) =_{df} ht \]
Syntax of UML statechart

And term.

\( s_1, \ldots, s_k \) UML-statechart terms for \( k > 0 \)

\[ s = [n,(s_1,\ldots, s_k),(en,ex)] \]

UML-statechart term with \( \text{type}(s) = \text{and} \)

\( s_1, \ldots, s_k \) called subterms of \( s \)

sequence of action \( a_1, \ldots, a_k \) \( \rightarrow \) \( <a_1,\ldots,a_k> \)
Syntax of UML statechart

Case study, Basic term

\[ s = [n,(en,ex)] \]

\[ s_5 = [n_5,(<e>,<>)] \]

\[ s_i = [n_i,(<>,,<>)] (6 \leq i \leq 9) \]
Syntax of UML statechart

Case study, Or term  $s = [n,(s_1,\ldots, s_k),l,T,(en,ex)]$

$s_2 = [n_2,(s_4,s_5),l,{t_1,t_4,t_5,t_6},(<>,,<>)]$

$s_3 = [n_3,(s_6,s_7),l,{t_2},(<>,,<>)]$  $s_4 = [n_4,(s_8,s_9),l,{t_3},(<>,,<d>)]$
Syntax of UML statechart

Case study, And term  \( s = [n,(s_1,\ldots, s_k),(en,ex)] \)

\[ s_1 = [n_1,(s_2,s_3),(<>,,<>)] \]
Syntax of UML statechart

Case study, transition

\[ t = (t, i, sr, e, \alpha, td, j, ht) \]

\[ t_1 = (t_1, 1, a, <c>, 2, \text{none}) \]

\[ t_2 = (t_2, 1, a, <d>, 2, \text{none}) \]
Syntax of UML statechart

Case study, transition

\[ t = (t, i, sr, e, \alpha, td, j, ht) \]

\[ t_3 = (t_3, 1, a, b, 2, \text{none}) \] \[ t_4 = (t_4, 1, \{n_9\}, c, a, 2, \text{none}) \]
Syntax of UML statechart

Case study, transition

\[ t = (t, i, sr, e, \alpha, td, j, ht) \]

\[ n_1 \]

\[ n_2 \]

\[ n_4 \]^\text{exit/d}\]

\[ n_5 \]^\text{entry/e}\]

\[ n_6 \]

\[ n_3 \]

\[ n_7 \]

\[ n_8 \]^t_3:a/<b>\]

\[ n_9 \]^t_1:a/<c>\]

\[ t_3:d/<>\]

\[ t_1:a/<c>\]

\[ t_5:d/<>\]

\[ t_4:c/<a>\]

\[ t_5 = (t_5, 2, d, <>, 1, \text{shallow}) \]

\[ t_6 = (t_6, 1, b, <>, 2, \text{none}) \]
Syntax of UML statechart

Configuration

conf: UML-SC $\rightarrow 2^N$

The set of the root names of all currently active substates within $s$, also including the root name of $s$

\[
\begin{align*}
\text{conf}([n,\_]) &= df \{n\} \\
\text{conf}([n,(s_{1..k}),l,T,\_]) &= df \{n\} \cup \text{conf}(s_l) \\
\text{conf}([n,(s_{1..K}),\_]) &= df \{n\} \cup \bigcup_{i=1}^{k} \text{conf}(s_i)
\end{align*}
\]
Syntax of UML statechart

Case study, configuration

conf(s₁) = \{ n₁, n₂, n₄, n₈, n₃, n₆ \}    conf(s₂) = \{ n₂, n₄, n₈ \}

Hauptseminar WS 05  Semantic der UML V2.0
Syntax of UML statechart

Subconfiguration subconf: UML-SC → 2^N

The set of all root names in the configuration of s which denote basic states.

\[
\begin{align*}
\text{subconf}([n,\_]) &= \{n\} \\
\text{subconf}([n,(s_1..k),l,T,\_]) &= \text{subconf}(s_i) \\
\text{subconf}([n, (s_1..k),\_]) &= \bigcup_{i=1}^{k} \text{subconf}(s_i)
\end{align*}
\]
Syntax of UML statechart

Case study, Subconfiguration

subconf(s_1) = \{n_8, n_6\}  
subconf(s_2) = \{n_8\}  

Hauptseminar WS 05  
Semantic der UML V2.0
Semantic of UML statechart

- Semantic of entry and exit actions.
- Semantics of the next state.
- Auxiliary UML-statechart semantics.
- UML-statechart semantics.
Exit actions

A transition \((\_, l, \_, \alpha, \_, i, \_)\) from \(s_i\) to \(s_i\) is taken

A sequence \(ex::\alpha::en\) of actions is executed

\(\text{exit}(s_i)\) is the set of all possible sequences of exit action of \(s_i\)

\(\text{exit}(\left[n, (en, ex)\right]) = df \{ex\}\)

\(\text{exit}(\left[n, (s_1..k), l, T, (en, ex)\right]) = df \{ex\,: ex | ex\in exit(s_i)\}\)

\(\text{exit}(\left[n, (s_1..k), (en, ex)\right]) = df \{m_1:: ... :: m_k :: ex | \exists \text{ bijection } b : \{1..k\} \rightarrow \{1..k\}. m_i \in exit(s_{b(i)}) \forall i \in \{1..k\}\)
Entry actions

A transition \((\_, \_\_\_, \alpha, \_, \_\_\_, \_)\) from \(s_i\) to \(s_i\) is taken

A sequence \(ex::\alpha::en\) of actions is executed

\[\text{entry}(s_i)\] is the set of all possible sequences of entry action of \(s_i\)

\[\text{entry}: \text{UML-SC} \rightarrow 2^{A^*}\]

\[\text{entry}([n, (en, ex)]) =_{df} \{ex\}\]

\[\text{entry}([n, (s_1..k), l, T, (en, ex)]) =_{df} \{en:: en\_\¥ | en\_\¥ \in \text{entry}(s_i)\}\]

\[\text{entry}([n, (s_1..k), (en, ex)]) =_{df} \{en:: m_1:: ... ::m_k | \exists \text{ bijection } b: \{1..k\} \rightarrow \{1..k\}. m_i \in \text{entry}(s_{b(i)}) \forall i \in \{1..k\}\}\]
Semantic of UML statechart

Semantics of the next state

\[ t \text{ UML-statechart transition} \quad h t \text{ historyType}(t) \]
\[ N = \text{tarDet}(t) \text{ of } t \quad \text{next: } HT \times N \times \text{UML-SC} \rightarrow \text{UML-SC} \]
\[ s \neq \text{next}(ht, \text{tarDet}(t), s) \]

\[ \text{next } (ht, N, [n]) =_{df} [n] \]
\[ \text{next } (ht, N, [n, (s_1..k), l, T]) =_{df} \begin{cases} 
[n, (s_1..k)[s_j/ \text{next}(ht, N, s_j)], j, T] & \text{if } \exists n' \in N, \\
\text{next_stop}(ht, [n, (s_1..k), l, T]) & \text{otherwise}
\end{cases} \]
\[ \text{next}(ht, N, [n, (s_1..k)]) =_{df} [n, (\text{next}(ht, N, s_1), ..., \text{next}(ht, N, s_k))] \]
Semantic of UML statechart

Semantics of the next state

In the definition of the function `next_stop`, the following case distinction occurs:

\[
\text{next\_stop}(ht, [n, (s_{1..k}), l, T]) =_{df}\begin{cases} 
[n, (s_{1..k}), l, T] & \text{if } ht = \text{deep} \\
[n, (s_{1..k})[s_1 / \text{default}(s_1)], l, T] & \text{if } ht = \text{none} \\
[n, (s_{1..k})[s_1 / \text{default}(s_1)], l, T] & \text{if } ht = \text{shallow}
\end{cases}
\]
Semantic of UML statechart

Semantics of the next state

In the definition of the function `next_stop`, the following case distinction occurs:

\[
\text{default}([n]) =_{df} [n]
\]

\[
\text{default}([n, (s_1..k), l, T]) =_{df} [n, (s_1..k)[s_1 / \text{default}(s_1)], 1, T]
\]

\[
\text{default}([n, (s_1..k)]) =_{df} [n, (\text{default}(s_1), ..., \text{default}(s_k))]
\]
Semantic of UML statechart

Auxiliary UML-statecharts semantics

Deals with processing single input events

\[[[\cdot]]_{\text{aux}} : \text{UML-SC} \rightarrow \text{LTS}\]

\(s \in \text{UML-SC}\) given by Labeled transition system \((\text{UML-SC}, L, \rightarrow, s) \in \text{LTS}\) : UML-SC set of states.

\(L = \Pi \times A^* \times \{0,1\}\) set of labels. \(s\) start state.

\(\rightarrow \subseteq \text{UML-SC} \times L \times \text{UML-SC}\) transition relation.

\((s,(e,\alpha,f),s\not\Rightarrow) \in \rightarrow\) \hspace{1cm} \(s \xrightarrow{e}_{\alpha,f} s\not\Rightarrow\)

\(f = 1\) (positive flag) \hspace{1cm} \(f = 0\) (negative flag) \hspace{1cm} stuttering step
Semantic of UML statechart

Auxiliary UML-statecharts semantics, transition relation $\rightarrow$

**name** | **premise** | **conclusion** (condition)
---|---|---

basic state BAS (stuttering)

$\text{true} \quad \frac{}{\text{BAS} \; e} \frac{}{[n] \xrightarrow{\alpha} [n]}$

Or-state $s$ OR-1 (progress)

$\frac{}{\text{OR-1} \; e} \frac{}{[n, (s_{1..k}), l, T]} \frac{}{[n, (s_{1..k})]_{\text{ex} :: \alpha :: \text{en}}} \frac{}{[n, (s_{1..k})]_{\text{ex} :: \alpha :: \text{en}}}$

$\left\{ \begin{array}{l}
ex \in \text{exit}(s_i), \\
en \in \text{entry}(\text{next}(ht,td,s_i))
\end{array} \right\}$
Semantic of UML statechart

Auxiliary UML-statecharts semantics, transition relation

Or-state s OR-2 (propagation of progress)

\[
\begin{align*}
\text{OR-2} & : \quad s_i \xrightarrow{e, \alpha} s_\neq \bar{s} \\
& \quad [n, (s_1..k), l, T] \xrightarrow{e, \alpha} \neq 1 [n,(s_1..k)[s_l/s_l'], l, T]
\end{align*}
\]

Or-state s OR-3 (propagation of stuttering)

\[
\begin{align*}
\text{OR-3} & : \quad s_i \xrightarrow{e, \neq} s_i, \quad [n, (s_1..k), l, T] \xrightarrow{e, \neq} 1 \\
& \quad [n, (s_1..k), l, T] \xrightarrow{e, <>} 0 [n,(s_1..k), l, T]
\end{align*}
\]
Semantic of UML statechart

Auxiliary UML-statecharts semantics, transition relation →

And-state s AND (composition)

\[ \forall j \in \{1,\ldots,k\}. s_j^e \xrightarrow{\alpha_j^{f_j}} s_j^\gamma \]

\[ [n, (s_1\ldots k)] \overset{\alpha}{\xrightarrow{\bigwedge_{j=1}^k f_j}} [n, (s_1\ldots k)] \]

\[ \alpha \in \{\alpha_1^{b(1)} : \ldots : \alpha_k^{b(k)} | \exists \text{ bijection } b: \{1..k\} \rightarrow \{1..k}\} \]
Example for Auxiliary UML-statecharts semantics,
Semantic of UML statechart

Complete UML-statechart semantics,

complete semantics $[[s]]$ of $s \in \text{UML-SC} \rightarrow$ Kripke structure $K = (S, st, \rightarrow) \in K$,

$S = \text{UML-SC} \times \prod^*$ set of Kripke states of $K$.

$st = (s, \varepsilon_0) \in S$ start state of $K$; $\varepsilon_0 \in \prod^*$

$\rightarrow \subseteq S \times S$ transition relation of $K$

get-input $s \xrightarrow{e, \alpha_f} s'$

$(s, \varepsilon) \rightarrow (s', \varepsilon')$ ($\exists (\varepsilon, e, \varepsilon') \in \text{sel}, \exists (\alpha, \varepsilon', \varepsilon'') \in \text{join}$)
Semantic of UML statechart

Conclusions and further work,
- UML-statechart features.
- Syntax of UML-statechart.
- Entry and exit semantics.
- Next-state semantics.
- SOS rules of auxiliary semantics.
- Complete semantics of UML-statecharts.

Additional features of UML-statechart

Model checking
Refinement checking
Transformation between tools
Semantic of UML statechart

Vielen Dank