A structured operational semantics for UML-statecharts

- Shadi Al-Dehni
  LMU München
- Jan Jürjens
  TU München

A structured operational semantics for UML-statecharts

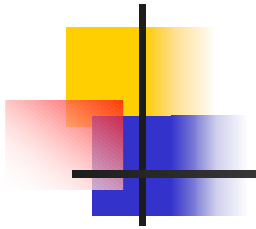
We will discuss in this Seminar the following concepts:

- UML Diagrams (Unified Modeling Language).
- UML- Statecharts
- Syntax of UML-statechart
- Configurations on UML-statechart diagram

A structured operational semantics for UML-statecharts

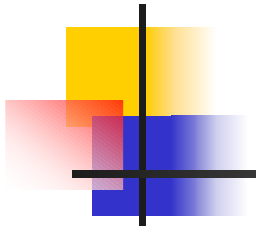
We will discuss in this Seminar the following concepts:

- 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. $\rightarrow$ (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.

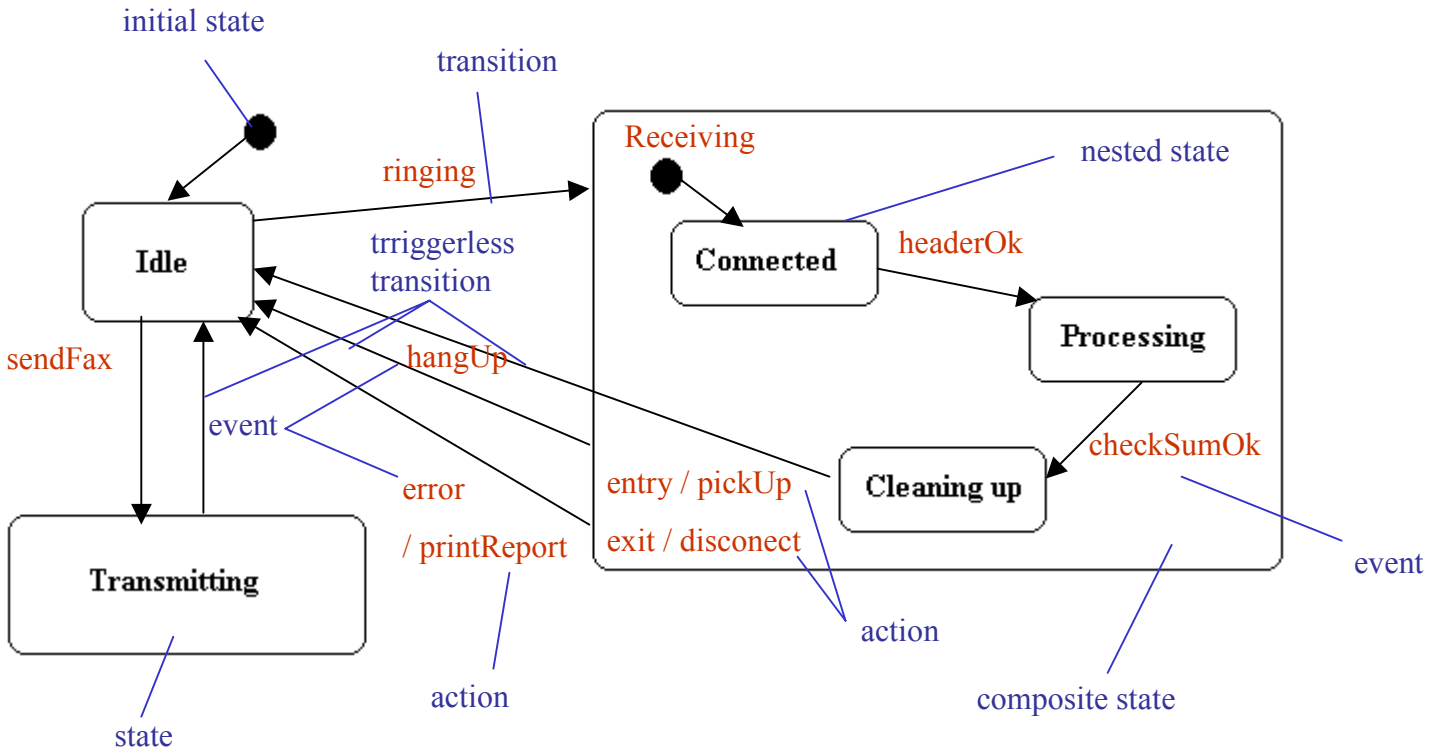
UML Statechart Diagram

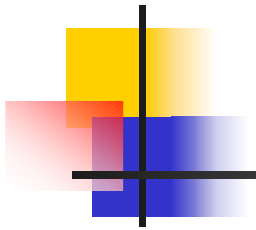
- A statechart diaram A state machine, emphasizing the flow of control from state to state. Which is a behavior that specifies the sequences of states an object goes through during ist lifetime in responce to events, together with ist responses to those events.
- A state: Condition or situation in the life of an object, satisfies some condition, performs some activity, or waits for some event.

UML Statechart Diagram

- An event: Occurance 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 occures 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.

UML Statechart Diagrams





Syntax of UML statechart

- Basic term.
- Or term.
- And term.
- Case study.
- Configurations.

Syntax of UML statechart

Basic term.

N set of state names,

T set of transition names,

Π set of events,

A set of actions.

a,b,c,... events or actions.

α,β,γ sequence of events or sequence of actions

$en, ex \in A, n \in N$

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

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

Syntax of UML statechart

Or term.

$s_1,..., s_k$ UML-statechart terms $\quad \rho = \{1,..,k\}, l \in \rho,$

$HT = \{none,deep,shallow\}$

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

$$s = [n,(s_1,..., s_k),l,T,(en,ex)]$$

UML-statechart term with type(s) = or

$$t = (\hat{t}, i, sr, e, \alpha, td, j, ht)$$

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

$sou(t) =_{df} s_i$, $tar(t) =_{df} s_j$, $historyType(t) =_{df} ht$

Syntax of UML statechart

And term.

$s_1,..., s_k$ UML-statechart terms for $k > 0$

$$s = [n,(s_1,..., s_k),(en,ex)]$$

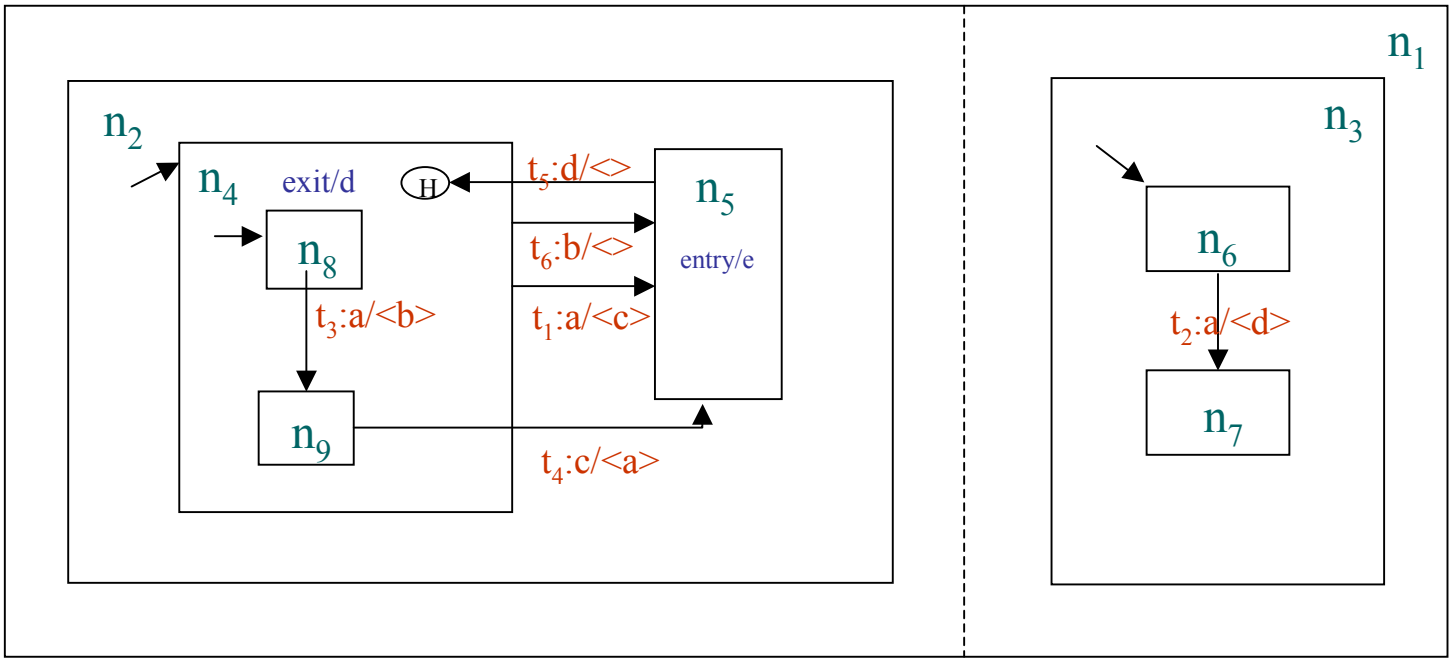
UML-statechart term with type(s) = and

$s_1,..., s_k$ called subterms of s

sequence of action $a_1, ..., a_k \longrightarrow <a_1,...,a_k>$

Syntax of UML statechart

Case study, Basic term $s = [n,(en,ex)]$

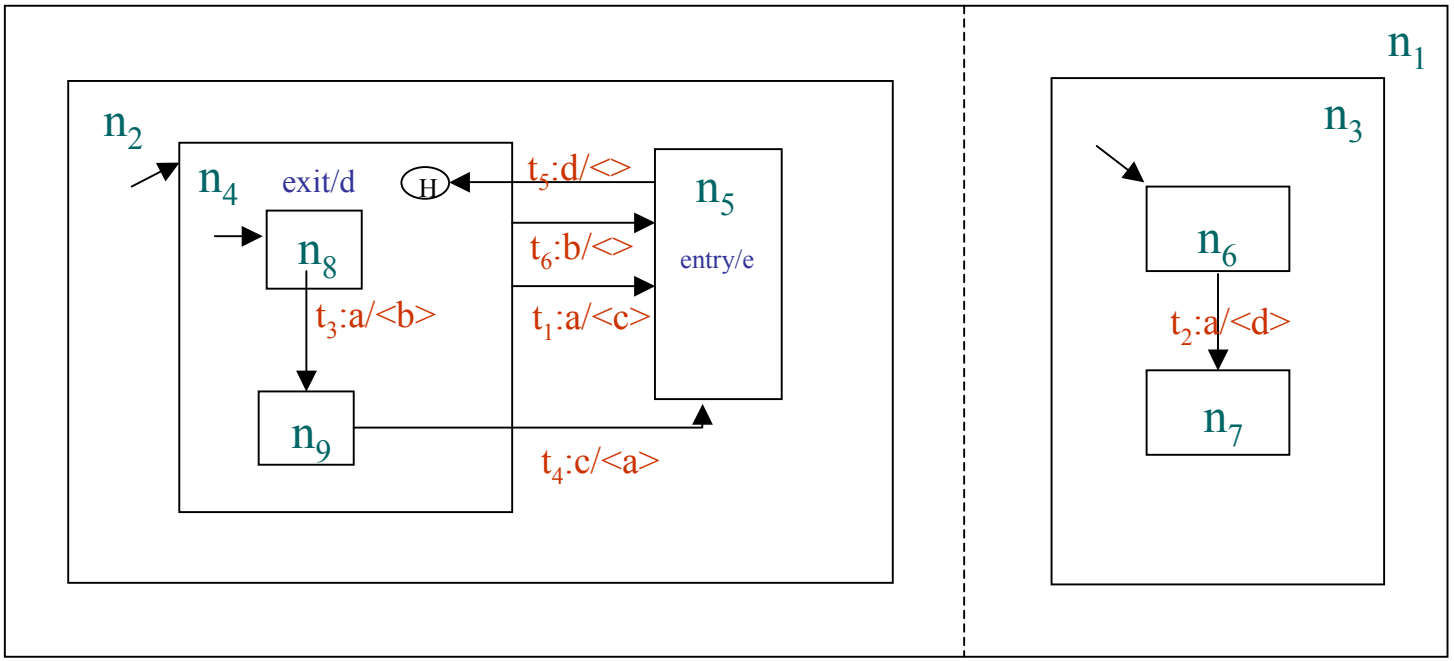


$s_5 = [n_5,(<e>,<>)]$

$s_i = [n_i,(<>,<>)]\ (6 \le i \le 9)$

Syntax of UML statechart

Case study, Or term $s = [n,(s_1,..., s_k),I,T,(en,ex)]$

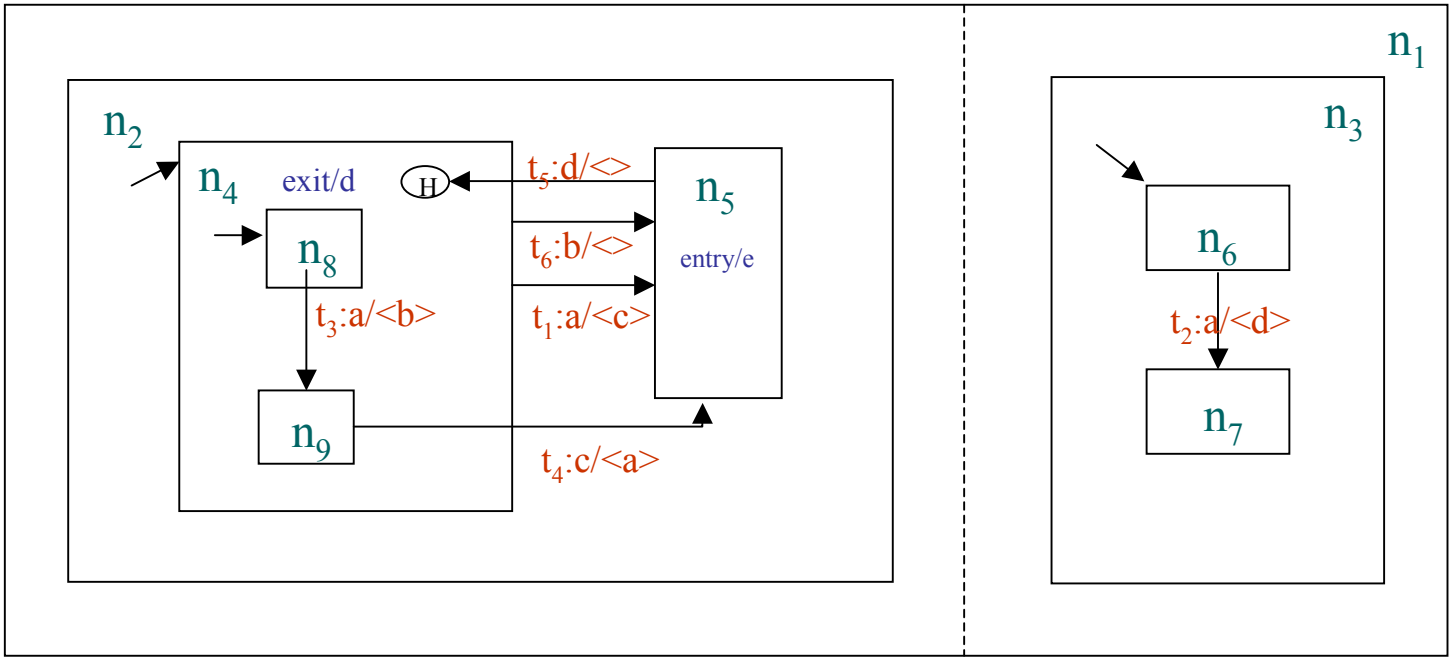


$s_2 = [n_2,(s_4,s_5),I,\{t_1,t_4,t_5,t_6\},(<>,<>)]$

$s_3 = [n_3,(s_6,s_7),I,\{t_2\},(<>,<>)]$ $s_4 = [n_4,(s_8,s_9),I,\{t_3\},(<>,<d>)]$

Syntax of UML statechart

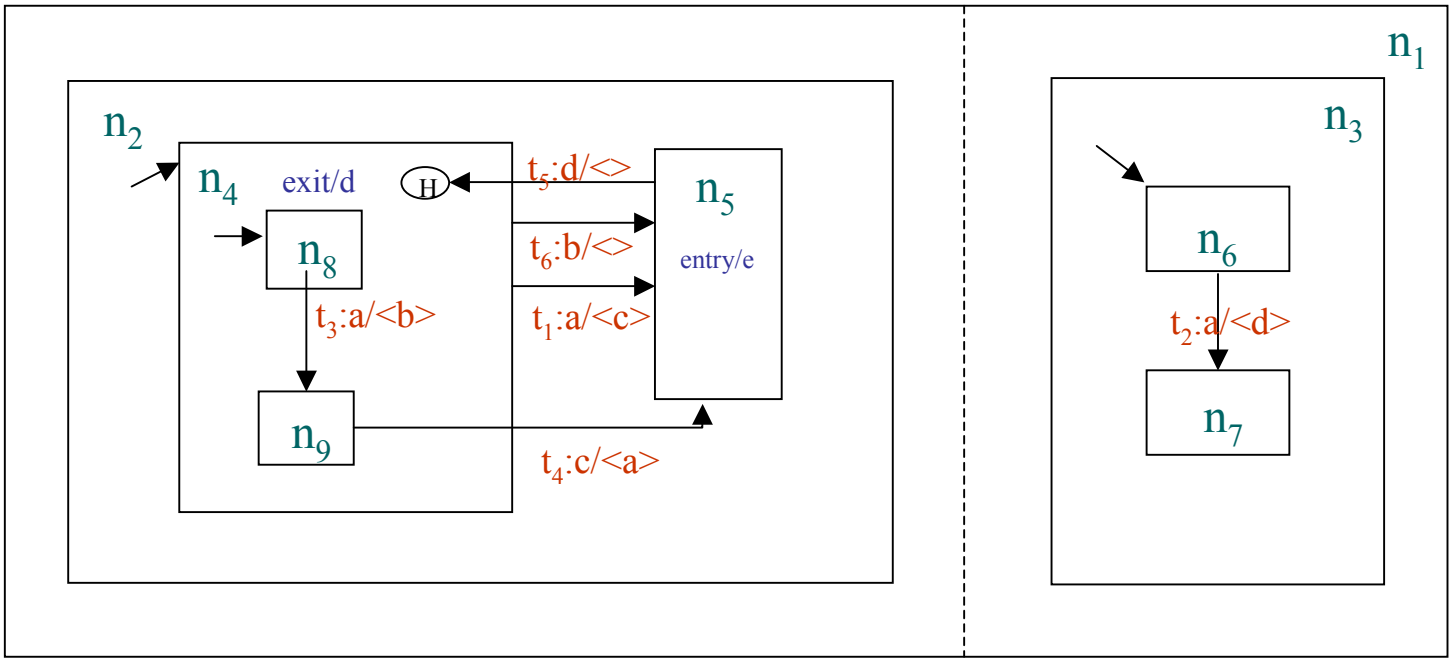
Case study, And term $s = [n,(s_1,\ldots, s_k),(en,ex)]$



$s_1 = [n_1,(s_2,s_3),(<>,<>)]$

Syntax of UML statechart

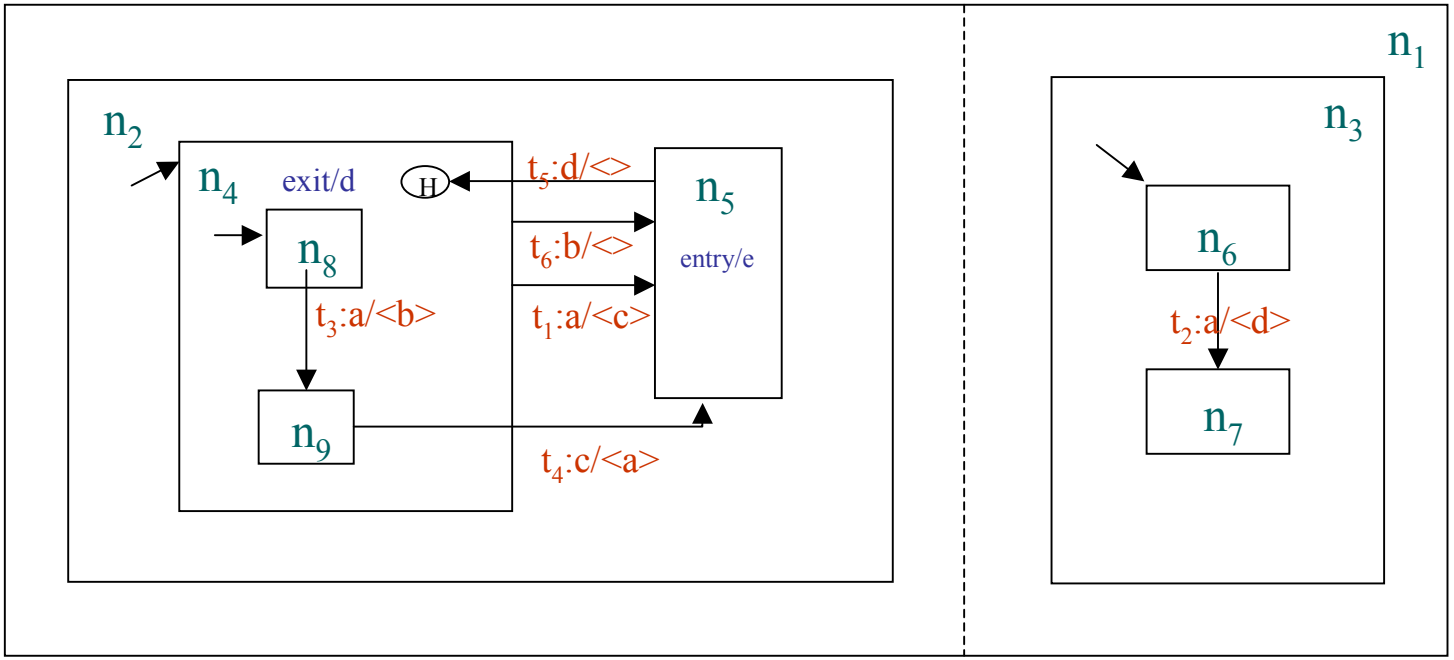
Case study, transition $t = (\hat{t}, i, sr, e, \alpha, td, j, ht)$



$t_1 = (\hat{t}_1, 1, \square, a, \langle c \rangle, \square, 2, \text{none})$ $t_2 = (\hat{t}_2, 1, \square, a, \langle d \rangle, \square, 2, \text{none})$

Syntax of UML statechart

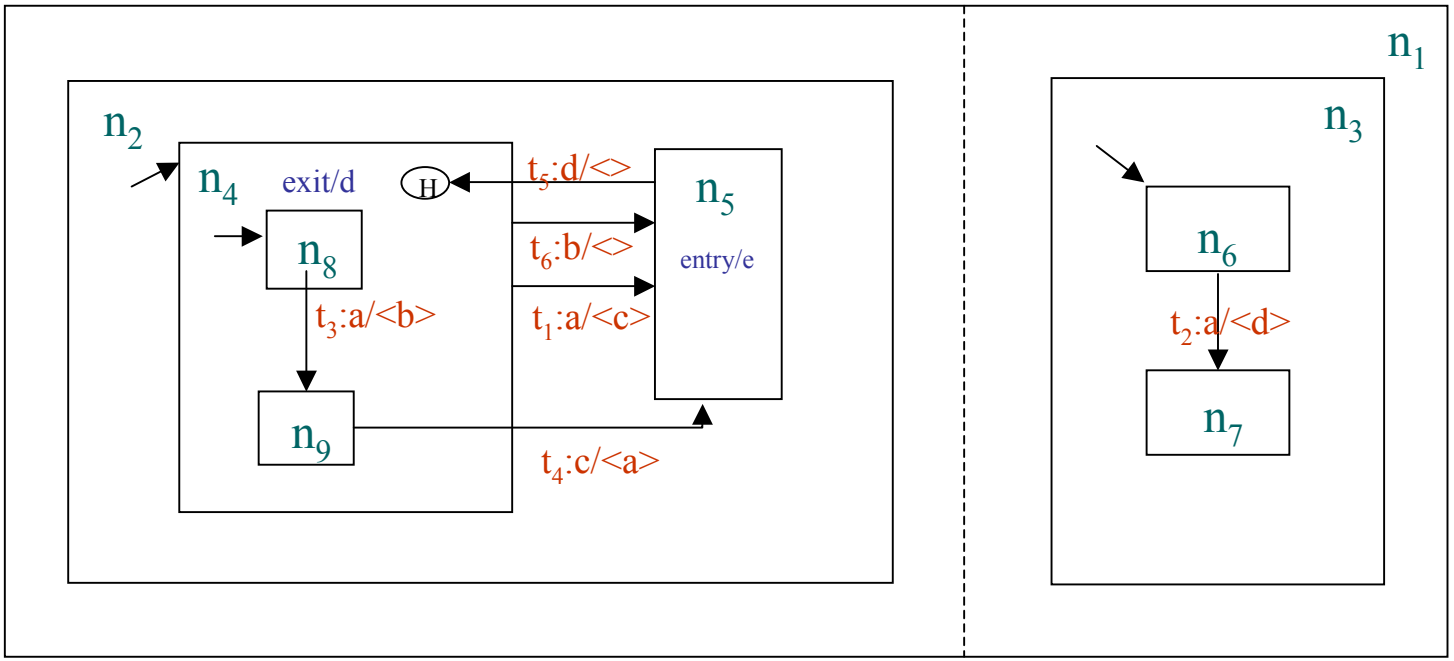
Case study, transition $t = (\hat{t}, i, sr, e, \alpha, td, j, ht)$



$t_3 = (\hat{t}_3, 1, \square, a, <b>, \square, 2, none)$ $t_4 = (\hat{t}_4, 1, \{n_9\}, c, <a>, \square, 2, none)$

Syntax of UML statechart

Case study, transition $t = (\hat{t}, i, sr, e, \alpha, td, j, ht)$



$t_5 = (\hat{t}_5, 2, \square, d, <>, \square, 1, \text{shallow})$ $t_6 = (\hat{t}_6, 1, \square, b, <>, \square, 2, \text{none})$

Syntax of UML statechart

Configuration $\quad conf: \text{UML-SC} \rightarrow 2^{N}$

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

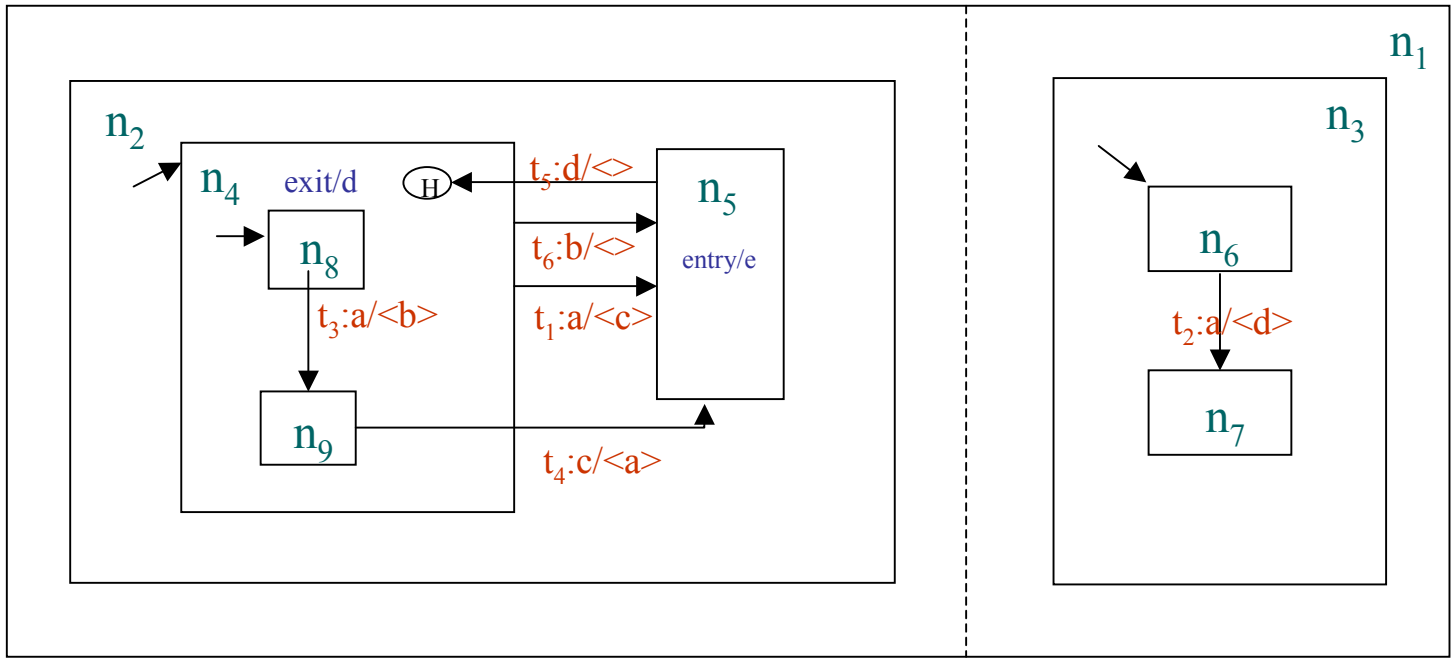
$$conf([n,\_]) =_{df} \{n\}$$

$$conf([n,(s_{1..k}),l,T,\_]) =_{df} \{n\} \cup conf(s_l)$$

$$conf([n, (s_{1..K}),\_]) =_{df} \{n\} \cup \bigcup_{i=1}^{k} conf(s_i)$$

Syntax of UML statechart

Case study, configuration



$conf(s_1) = \{n_1, n_2, n_4, n_8, n_3, n_6\}$ $conf(s_2) = \{n_2, n_4, n_8\}$

Syntax of UML statechart

Subconfiguration $\quad subconf: UML\text{-}SC \rightarrow 2^{N}$

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

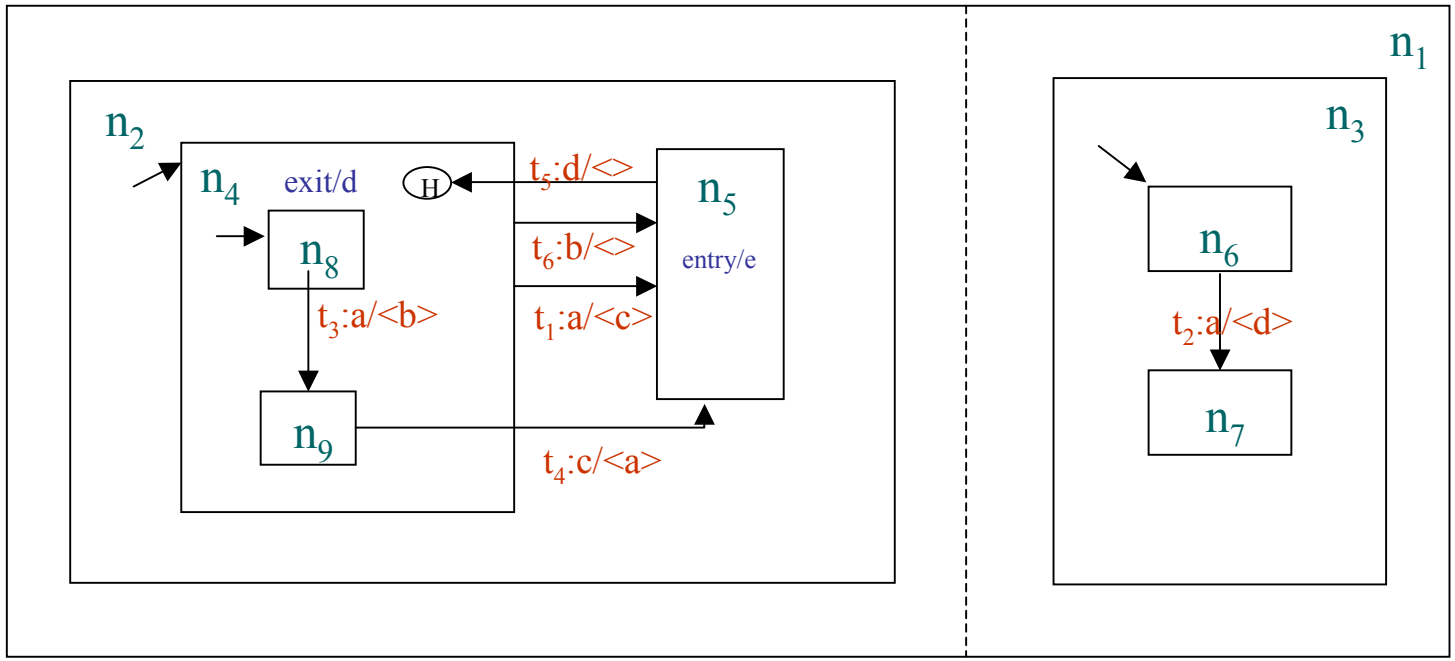
$$subconf([n,\_]) =_{df} \{n\}$$

$$subconf([n,(s_{1..k}),I,T,\_]) =_{df} subconf(s_I)$$

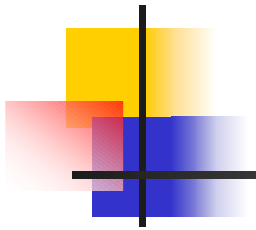
$$subconf([n, (s_{1..K}),\_]) =_{df} \bigcup_{i=1}^{k} subconf(s_i)$$

Syntax of UML statechart

Case study, Subconfiguration



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



Semantic of UML statechart

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

Semantic of UML statechart

Exit actions

a transition (_,l,_, α, _,i,_) from s_l to s_i is taken

⟹ A sequence ex::α::en of actions is executed

$exit(s_l)$ is the set of all possible sequences of exit action of s_l

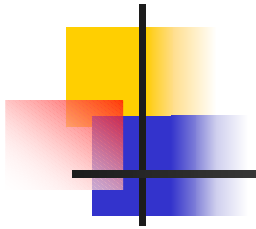
$$exit: \text{UML-SC} \rightarrow 2^{A^*}$$

$$exit([n, (en, ex)]) =_{df} \{ex\}$$

$$exit([n, (s_{1..k}),l,T,(en,ex)]) =_{df} \{ex¥: ex \mid ex¥\in exit(s_l)\}$$

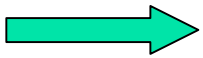
$$exit([n, (s_{1..k}), (en,ex)]) =_{df} \{m_1:: \ldots :: m_k :: ex \mid \exists \text{ bijection } b : \{1..k\} \rightarrow \{1..k\}.\ m_i \in exit(s_{b(i)})\ \forall i Ä \{1..k\}\}$$

Semantic of UML statechart



Entry actions

a transition (_,l,_, α, _,i,_) from s_l to s_i is taken



A sequence ex::α::en of actions is executed

$entry(s_i)$ is the set of all possible sequences of entry action of s_i

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

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

$$entry([n, (s_{1..k}),l,T,(en,ex)]) =_{df} \{en:: en¥| en¥\in entry(s_l)\}$$

$$entry([n, (s_{1..k}), (en,ex)]) =_{df} \{en:: m_1:: \ldots ::m_k \mid \exists \text{ bijection } b : \{1..k\} \rightarrow \{1..k\}.\ m_i \in entry(s_{b(i)})\ \forall i \in \{1..k\}\}$$

Semantic of UML statechart

Semantics of the next state

t UML-statechart transition ht historyType(t)

$\mathbb{N}$ = tarDet(t) of t next: HT $\times$ $\mathbb{N}$ $\times$ UML-SC $\rightarrow$ UML-SC

s¥= next(ht, tarDet(t), s)

next (ht, N, [n]) $=_{df}$ [n]

next (ht,N,[n, ($s_{1..k}$),I,T])

$=_{df}$ { [n, ($s_{1..k}$)$_{[sj/\ next(ht,N,sj)]}$, j, T] if ∃n´ ∈ N, j Ä {1,..., K}. n¥=name(s_j)

next_stop(ht,[n,($s_{1..k}$),I, T]) otherwise

next(ht, N, [n, ($s_{1..k}$)])

$=_{df}$ [n, (next(ht, N, s_1), ..., 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 destinction occurs:

$$\text{next\_stop}(ht, [n, (s_{1..k}), l, T]$$

$$=_{df} \begin{cases} [n, (s_{1..k}), l, T] & \text{if } ht = deep \\ [n, (s_{1..k})_{[s1 \,/\, default(s1)]}, 1, T] & \text{if } ht = none \\ [n, (s_{1..k})_{[s1 \,/\, default(sl)]}, l, T] & \text{if } ht = shallow \end{cases}$$

Semantic of UML statechart

Semantics of the next state

In the definition of the function next_stop, the following case destinction occurs:

$default([n]) =_{df} [n]$

$default([n, (s_{1..k}), l, T]) =_{df} [n, (s_{1..k})_{[s1 / default(s1)]}, 1, T]$

$default([n, (s_{1..k})]) =_{df} [n, (default(s_1), ..., default(s_k))]$

Semantic of UML statechart

Auxiliary UML-statecharts semantics

Deals with processing single input events

$[[.]]_{aux}$: UML-SC ⟶ LTS

s ∈ UML-SC given by Labeled transition system (UML-SC,L,→, s) ∈ LTS : UML-SC set of states.

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

$\rightarrow \subseteq$ UML-SC × L × UML-SC transition relation.

(s,(e,α,f),s¥) ∈ → ⟹ $s \xrightarrow[\alpha]{e}_f s¥$

f =1 (positive flag) f =0 (negative flag) ⟹ stuttering step

Semantic of UML statechart

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

$$\text{name}\ \frac{\text{premise}}{\text{conclusion}}\ \text{(condition)}$$

basic state BAS (stuttering)

$$\text{BAS}\ \frac{\text{true}}{[n] \xrightarrow[\alpha]{e}_0 [n]}$$

Or-state s OR-1 (progress)

$$\text{OR-1}\ \frac{(\_,l,sr,e,\alpha,td,i,ht) \in T,\ sr \subseteq conf(s_l),\ s_l \overset{e}{\not\mapsto}_1 \quad \left[\begin{array}{c} ex \in exit(s_l), \\ en \in entry(next(ht,td,s_i)) \end{array}\right]}{[n, (s_{1..k}),l,T] \xrightarrow[ex::\alpha::en]{e}_1 [n,(s_{1..k})_{[si\ /\ next(ht,td,si)]}, i,T]}$$

Semantic of UML statechart

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

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

$$\text{OR-2}\ \frac{s_l \xrightarrow[\alpha]{e}_1 s¥}{[n, (s_{1..k}), l, T] \xrightarrow[\alpha]{e}_1 [n, (s_{1..k})_{[sl\ /\ sl']}, l, T]}$$

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

$$\text{OR-3}\ \frac{s_l \xrightarrow[<>]{e}_0 s_l,\quad [n, (s_{1..k}), l, T] \not\xrightarrow{e}_1}{[n, (s_{1..k}), l, T] \xrightarrow[<>]{e}_0 [n, (s_{1..k}), l, T]}$$

Semantic of UML statechart

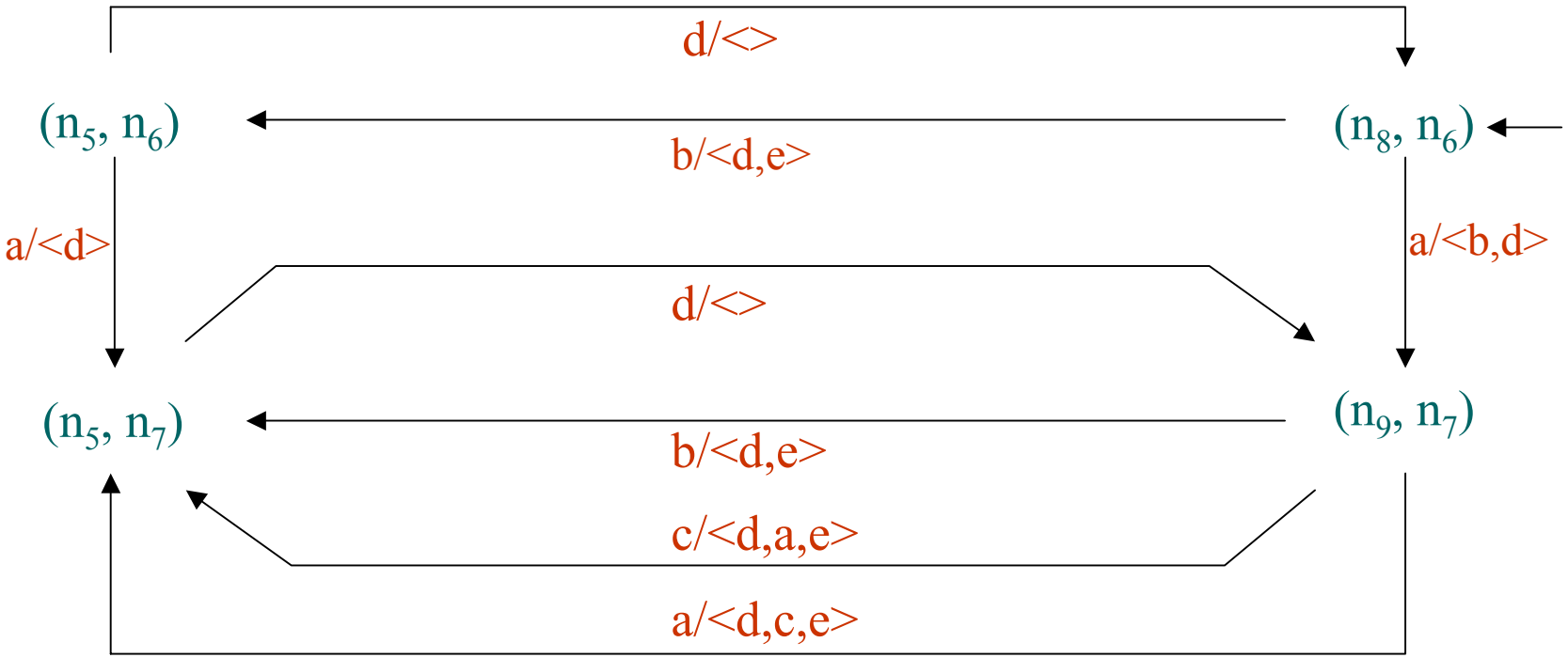
Auxiliary UML-statecharts semantics, transition relation $\rightarrow$

And-state s AND (composition)

$$\text{AND}\ \frac{\forall j \in \{1,\ldots,k\}.s_j \xrightarrow{e}_{\alpha_j f_j} s¥_j}{[n,(s_{1..k})] \xrightarrow{e}_{\alpha V_{J=1}^{k} f_j} [n,(s´_{1..k})]} \left(\alpha \in \{\alpha_{b(1)} :\ldots:: \alpha_{b(k)} \mid \exists \text{ bijection } b: \{1..k\} \rightarrow \{1..k\}\} \right)$$

Semantic of UML statechart

Example for Auxiliary UML-statecharts semantics,



Semantic of UML statechart

Complete UML-statechart semantics,

complete semantics [[s]] of $s \in$ 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-inp $\dfrac{s \xrightarrow[\alpha\, f]{e} s¥}{(s, \varepsilon) \rightarrow (s¥, \varepsilon¥¥)}$ $(\exists(\varepsilon, e, \varepsilon¥) \in \text{sel}, \exists(\alpha, \varepsilon¥, \varepsilon¥¥) \in \text{join})$

Semantic of UML statechart

Conclusions and further work,

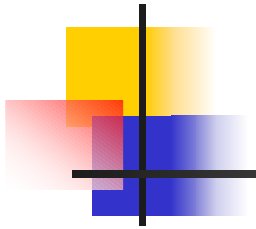
- UML-statechart features.
- Syntax of UML-statechart.
- Entry and exit semantics.
- Next-state semantics.
- SOS rules of auxiliary semantics.
- Complete semantics of UML-statecharts.

Model checking

Refinement checking

Transformation between tools

Additional features of UML-statechart



Semantic of UML statechart

Vielen Dank