Theoretical Foundations of the UML

Lecture 2: Sequence Diagrams

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

http://moves.rwth-aachen.de/i2/uml09100/

16. Oktober 2012

RWTH AACHEN UNIVERSITY

History

- 70s - 80s: often used informally
- 1992: first version of MSCs standardized by CCITT (currently ITU) Z.120
- 1992 - 1996: many extensions, e.g., high-level + formal semantics (using process algebras)
- 1996: MSC'96 standard
- 2000: MSC 2000, time, data, o-o features
- 2005: MSC 2004 ...

RWTH AACHEN UNIVERSITY

Variants of MSCs

- UML sequence diagrams
- (instantiations of) use cases
- triggered MSCs
- netcharts (= Petri net + MSC)
- STAIRS
- Live sequence charts
- ...

Characteristics

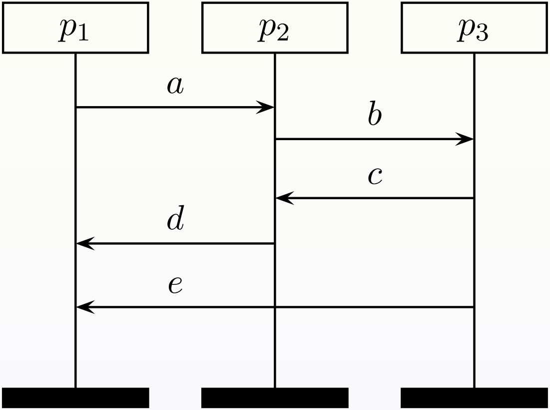
- scenario-based language
- visual representation
- "easy" to comprehend
- generalization possible towards automata (states are MSCs)
- widely used in industrial practice

RWTH AACHEN UNIVERSITY

Applications

- requirements specification
  (positive, negative scenarios, e.g., CREWS)
- system design and software engineering
- visualization of test cases
  (graphical extension to TTCN)
- feature interaction detection
- workflow management systems
- ...

Example



Preliminaries (1)

Definition

Let $\mathcal{P}$: finite set of ≥ 2 sequential processes
$\mathcal{C}$: finite set of message contents $(a, b, c, \ldots \in \mathcal{C})$

Definition

Communication action: $p, q \in \mathcal{P}$, $p \neq q$, $a \in \mathcal{C}$

$!(p, q, a)$ "p sends message a to q"

$?(p, q, a)$ "p receives message a sent by q"

Let Act denote the set of actions

RWTH AACHEN UNIVERSITY

Preliminaries (2)

Definition

Let E be a set of events

A partial order over E is a relation $\preceq \subseteq E \times E$ such that:

1. $\preceq$ is reflexive, i.e., $\forall e \in E.\, e \preceq e$,
2. $\preceq$ is transitive, i.e., $e \preceq e' \wedge e' \preceq e''$ implies $e \preceq e''$, and
3. $\preceq$ is anti-symmetric, i.e., $\forall e, e'.\, (e \preceq e' \;\wedge\; e' \preceq e) \Rightarrow e = e'$.

Definition

Let $(E, \preceq)$ be a poset.

The Hasse diagram $(E, \lessdot)$ is defined by:

$$e \lessdot e' \text{ iff } e \preceq e' \text{ and } \neg(\exists e'' \neq e, e'.\, e \preceq e'' \preceq e')$$

Preliminaries (3)

Definition

Let $(E, \preceq)$ be a poset.
A linearization of $(E, \preceq)$ is a total order $\sqsubseteq$ such that

$$e \preceq e' \quad \text{implies} \quad e \sqsubseteq e'$$

A linearization is a topological sort of the Hasse diagram of $(E, \preceq)$.

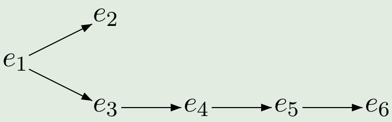
Preliminaries (4)

Example

Let $E = \{e_1, \ldots, e_6\}$,

$$\begin{array}{rcll} \preceq & = & \{ & (e_1,e_2),(e_1,e_3),(e_3,e_4),(e_4,e_5),(e_5,e_6),(e_1,e_4), \\ & & & (e_3,e_5),(e_1,e_5),(e_1,e_6),(e_3,e_6),(e_4,e_6) \\ & & \}^r & \text{where } R^r \text{ denotes the reflexive closure of } R \end{array}$$

Hasse diagram:



Linearizations:

- $e_1 e_2 e_3 e_4 e_5 e_6$,
- $e_1 e_3 e_2 e_4 e_5 e_6$,
- $e_1 e_3 e_4 e_2 e_5 e_6$,
- $e_1 e_3 e_4 e_5 e_2 e_6$,
- $e_1 e_3 e_4 e_5 e_6 e_2$

Not a linearization:

- $e_2 e_1 e_3 \ldots$, and $e_1 e_4 e_3 \ldots$

Message Sequence Chart (MSC) (1)

Definition

An MSC $M = (\mathcal{P}, E, \mathcal{C}, l, m, <)$ with:

- $\mathcal{P}$, a finite set of processes $\{p_1, p_2, \ldots, p_n\}$
- E, a finite set of events

$$E = \biguplus_{p \in \mathcal{P}} E_p = \underbrace{E_? \uplus E_! \uplus E_{\text{loc}}}_{\text{partitioning of } E}$$

- $\mathcal{C}$, a finite set of message content
- $l : E \to Act$, a labelling function defined by:

$$l(e) = \begin{cases} !(p,q,a) & \text{if} \quad e \in E_p \cap E_! \\ ?(p,q,a) & \text{if} \quad e \in E_p \cap E_? \\ p(a) & \text{if} \quad e \in E_p \cap E_{\text{loc}} \end{cases} \quad , p \neq q \in \mathcal{P},\ a \in \mathcal{C}$$

UNIVERSITY

Message Sequence Chart (MSC) (2)

Definition

- $m : E_! \to E_?$ a bijection ("matching function"), satisfying:

$$m(e) = e' \wedge l(e) = !(p, q, a) \text{ implies } l(e') = ?(q, p, a) \ \ (p \neq q, \ a \in \mathcal{C})$$

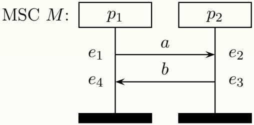
- $< \subseteq E \times E$ is a partial order ("visual order") defined by:

$$< = \Big(\underbrace{\bigcup_{p \in \mathcal{P}} <_p}_{} \ \cup \ \underbrace{\{(e, m(e)) \mid e \in E_!\}}_{} \Big)^*$$

$<_p$ is a total order = "top-to-bottom" order on process p

communication order $<_c$

where for relation R, R^* denotes its reflexive and transitive closure.



$M = (\mathcal{P}, E, \mathcal{C}, l, m, <)$ with:

$\mathcal{P} = \{p_1, p_2\}$ $\quad E_{p_1} = \{e_1, e_4\}$

$E = \{e_1, e_2, e_3, e_4\}$ $\quad E_{p_2} = \{e_2, e_3\}$

$\mathcal{C} = \{a, b\}$ $\quad E_! = \{e_1, e_3\}, E_? = \{e_2, e_4$

$l(e_1) = !(p_1, p_2, a)$ $\quad m(e_1) = e_2$

$l(e_2) = ?(p_2, p_1, a)$

$l(e_3) = !(p_2, p_1, b)$ $\quad m(e_3) = e_4$

$l(e_4) = ?(p_1, p_2, b)$

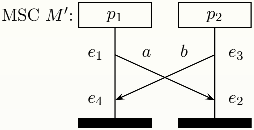
Ordering at processes: $e_1 <_{p_1} e_4$ and $e_2 <_{p_2} e_3$

Hasse diagram of $(E, <)$:

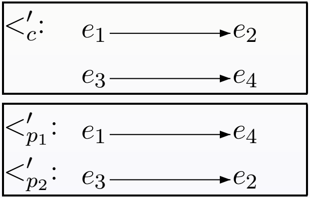
$$e_1 \longrightarrow e_2 \longrightarrow e_3 \longrightarrow e_4$$

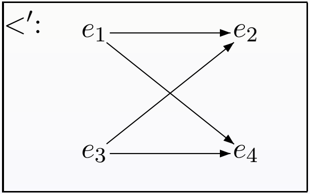
Linearizations?

Example (2)



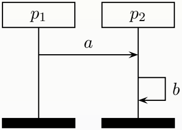
$$M' = (\underbrace{\mathcal{P}, E, \mathcal{C}, l, m}_{\text{as above}}, <') \text{ with:}$$





Example (3)

Not an MSC:

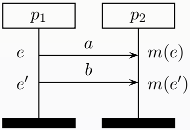


FIFO property

MSC $M = (\mathcal{P}, E, \mathcal{C}, l, m, <)$ has the *First-In-First-Out* (FIFO) property whenever:
for all $e, e' \in E_!$ we have

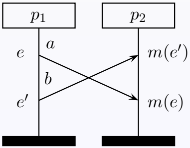
$$e < e' \wedge l(e) = !(p, q, a) \wedge l(e') = !(p, q, b) \text{ implies } m(e) < m(e')$$

i.e., "no message overtaking allowed"



FIFO

$$\begin{aligned} & l(e) = !(p_1, p_2, a) \\ & l(e') = !(p_1, p_2, b) \\ & e < e' \\ \Rightarrow\ & m(e) < m(e') \end{aligned}$$



non-FIFO

Note:

We assume an MSC to possess the FIFO property, unless stated otherwise!

Definition

Let $Lin(M)$ = denote the set of linearizations of MSC M.

Lemma: MSCs and their linearizations are interchangeable

There is a one-to-one correspondence between an MSC and its set of linearizations.

Thus:

$Lin(M)$ uniquely characterizes M.

Well-formedness

Let $Ch := \{(p,q) \mid p \neq q\,,\, p,q \in \mathcal{P}\}$ be a set of channels over $\mathcal{P}$.

We call $w = a_1 \ldots a_n \in Act^*$ proper if

1. every receive in w is preceded by a corresponding send, i.e.:
   $\forall (p,q) \in Ch$ and prefix u of w, we have:

$$\underbrace{\sum_{m \in \mathcal{C}} |u|_{!(p,q,m)}}_{\#\text{ sends from } p \text{ to } q} \quad \geqslant \quad \underbrace{\sum_{m \in \mathcal{C}} |u|_{?(q,p,m)}}_{\#\text{ receipts by } q \text{ from } p}$$

   where $|u|_a$ denotes the number of occurrences of action a in u

2. the FIFO policy is respected, i.e.:
   $\forall 1 \leqslant i < j \leqslant n$, $(p,q) \in Ch$, and $a_i = !(p,q,m_1)$, $a_j = ?(q,p,m_2)$:

$$\sum_{m \in \mathcal{C}} |a_1 \ldots a_{i-1}|_{!(p,q,m)} \;=\; \sum_{m \in \mathcal{C}} |a_1 \ldots a_{j-1}|_{?(q,p,m)} \quad \text{implies} \quad m_1 = m_2$$

A proper word w is well-formed if $\sum_{m \in \mathcal{C}} |w|_{!(p,q,m)} = \sum_{m \in \mathcal{C}} |w|_{?(q,p,m)}$

Properties of well-formedness

Lemma

For any MSC M, $w \in Lin(M)$ is well-formed.

we use $Lin(M)$ here as a set of words (and not linearizations)
the word of linearization $e_1 \ldots e_n$ equals $\ell(e_1) \ldots \ell(e_n)$

RWTH AACHEN UNIVERSITY

From linearizations to posets

Associate to $w = a_1 \ldots a_n \in Act^*$ an Act-labelled poset

$$M(w) = (E, \prec, \ell)$$

such that:

- $E = \{1, \ldots, n\}$ are the positions in w labelled with $\ell(i) = a_i$
- $\prec = \left(\prec_{\text{msg}} \cup \bigcup_{p \in \mathcal{P}} \prec_p \right)^*$ where
  - $i \prec_p j$ if and only if $i < j$ for any $i, j \in E_p$
  - $i \prec_{\text{msg}} j$ if for some $(p, q) \in Ch$ and $m \in \mathcal{C}$ we have:

$$\ell(i) = !(p,q,m) \text{ and } \ell(j) = ?(q,p,m) \text{ and}$$
$$\sum_{m \in \mathcal{C}} |a_1 \ldots a_{i-1}|_{!(p,q,m)} = \sum_{m \in \mathcal{C}} |a_1 \ldots a_{j-1}|_{?(q,p,m)}$$

Example

construct $M(w)$ for $w = !(r,q,m)!(p,q,m_1)!(p,q,m_2)?(q,p,m_1)?(q,p,m_2)?(q,r,m)$

Properties

Relating well-formed words to MSCs

For any well-formed $w \in Act^*$, $M(w)$ is an MSC.

Definition

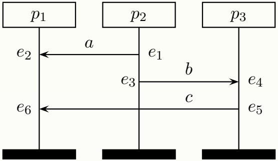
$(E, \preceq, \ell)$ and $(E', \preceq', \ell')$ are isomorphic if there exists a bijection $f : E \to E'$ such that $e \preceq e'$ iff $f(e) \preceq' f(e')$ and $\ell(e) = \ell'(f(e))$.

Linearizations yield isomorphic MSCs

For any well-formed $w \in Act^*$ and $w' \in Lin(M(w))$:

$$M(w) \text{ and } M(w') \text{ are isomorphic.}$$

Visual order vs. possible order



$e_2 < e_6$?

If message b takes much shorter than message a,
then c might arrive at p_1 before a!

Formally: $<_{p_1} = \{e_6, e_2\}$ is possible but $\neq$ visual order.

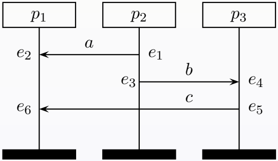
When are such situations possible and how to detect them?

Races (1)

- Let $M = (\mathcal{P}, E, \mathcal{C}, l, m, <)$ be an MSC.
- Let $\ll \subseteq E \times E$ be defined by:

$$
\begin{array}{lll}
e \ll e' & \text{iff} & e' = m(e) \\
 & \text{or} & e <_p e' \text{ and } E_! \cap \{e, e'\} \neq \varnothing \\
 & \text{or} & e, e' \in E_p \cap E_? \text{ and } m^{-1}(e) <_q m^{-1}(e')
\end{array}
$$

$\ll$ is the "interpreted / possible order" (also called causal order)



Example

$e_1 \ll e_2, \quad e_3 \ll e_4, \quad e_5 \ll e_6, \quad e_1 \ll e_3, \quad e_4 \ll e_5, \quad \neg(e_2 \ll e_6)$

Races (2)

Definition

MSC M contains a race if for some $e, e' \in E_?$:

$$e <_p e' \text{ but } \neg(e \ll^* e')$$

where $\ll^* \subseteq E \times E$ is the reflexive and transitive closure of $\ll$.

- How to check whether MSC M has a race?

  compute $\ll^*$ and compare to $<_p$

- $\ll^*$ can be computed using Floyd-Warshall's algorithm

  worst-case time complexity $\mathcal{O}(|E|^3)$, improved here to $\mathcal{O}(|E|^2)$

Races (3)

MSC M has a race if $< \not\subseteq \ll^*$ or equivalently:

$$\exists e, e' \in E_? \;.\; (e <_p e' \text{ and } e \not\ll^* e')$$

$\Rightarrow$ system implementation based on $<_p$ may cause problems, e.g.,

1. unspecified message reception
2. deadlock situations
3. use content of wrong message

Computing $\ll^*$: Warshall's algorithm

Algorithm

$\underbrace{\text{compute } \ll^*}_{\text{Warshall's Algorithm}}$ and compare with $<$

Warshall's Algorithm: <u>input</u>: $R \subseteq X \times X$ where X is a set
<u>output</u>: R^*

Idea:

Consider R and R^* as directed graphs

There is an edge $x \Rightarrow y$ in R^* iff there is a (possibly empty) path

$$x = x_0 \to x_1 \to x_2 \to \ldots \to x_n = y \text{ in } R$$

(our setting: $X = E, R = \ll, \ R^* = \ll^*$)

Warshall's algorithm

- assume: vertices are numbered $\{1, 2, \ldots, n\}$ where $n = |E|$
- for $j \in \{1, \ldots, n{+}1\}$ define relation $\stackrel{j}{\Longrightarrow}$ as follows:
  $x \stackrel{j}{\Longrightarrow} y$ iff $\exists$ path in R from x to y such that all vertices on the path ($\neq x, y$) have a smaller number than j
- Then:
  (1) $x \Longrightarrow y$ iff $x \stackrel{n+1}{\Longrightarrow} y$
  (2) $x \stackrel{1}{\Longrightarrow} y$ iff $x = y$ or $x \ll y$
  (3) $x \stackrel{y+1}{\Longrightarrow} z$ iff $x \stackrel{y}{\Longrightarrow} z$ or $x \stackrel{y}{\Longrightarrow} y \stackrel{y}{\Longrightarrow} z$
- Algorithm: determine the relations $\stackrel{1}{\Longrightarrow}, \ldots, \stackrel{n}{\Longrightarrow}, \stackrel{n+1}{\Longrightarrow}$ iteratively using properties (2) + (3); Result is then given by (1).
- Store $\stackrel{i}{\Longrightarrow}$ in a boolean matrix C
- Postcondition: $C[x, y] = \texttt{true}$ iff $(x, y) \in R^*$
- Precondition: $\forall x, y \in X \;.\; C[x, y] = \texttt{false}$

Warshall's algorithm (1)

```
for x := 1 to n do
    for y := 1 to n do
        C[x, y] := (x = y or (x, y) ∈ R)
```

$$C[x,y] := (x = y \text{ or } \underbrace{(x,y) \in R}_{x \ll y})$$

/* loop invariant */

/* after the j-th iteration of outermost loop it holds: $C[x,y]$ iff $x \overset{j+1}{\Longrightarrow} y$ */

```
for y := 1 to n do
    for x := 1 to n do
        if C[x, y] then
            for z := 1 to n do
                if C[y, z] then
                    C[x, z] := true
```

Correctness and complexity

Lemma: correctness

After j iterations: $x \xrightarrow{j+1} y$ iff $C[x, y] = 1$.

Proof:

if: trivial; *only if*: by induction on j.

Complexity

Time complexity of Warshall's algorithm : $\mathcal{O}(n^3)$ where $n = |X|$

Proof:

follows from the fact that each loop has at most n iterations.

Efficiency improvement [Alur et al. '96] (1)

Warshall's algorithm determines R^* for any binary relation R.

Recall: our interest is in determining R^* for $R = \ll$

Using some properties of $\ll$ the complexity can be improved.

Exploit that for $\ll$:

- $\ll$ is acyclic (as it is a partial order)
- number of immediate predecessors of $e \in E$ under $\ll$ is at most two (why?)

  Recall that e is an immediate predecessor of e' (under $\ll$) if:

$$e \ll e' \text{ and } \neg(\exists e'' \notin \{e, e'\}.\ e \ll e'' \ll e')$$

Efficiency improvement [Alur et al. '96] (2)

Body of the algorithm for detecting races now becomes:

for $e := 1$ to n do
  for $e' := e - 1$ downto 1 do
    if $C[e', e]$ then
      for $e'' := 1$ to $e' - 1$ do
        if $C[e'', e']$ then
          $C[e'', e] :=$ `true`

      for $e'' := 1$ to $e' - 1$ do
        if $C[e'', e']$ then
          $C[e'', e] :=$ `true`

this part is executed for (e, e') only if e' is an immediate predecessor of e, i.e., number of loops per outermost iteration is $\leq 2 \cdot n \Longrightarrow$ time complexity $\mathcal{O}(n^2)$