Overview

Theoretical Foundations of the UML

Lecture 1: Introduction

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

moves.rwth-aachen.de/teaching/ss-16/theoretical-foundations-of-the-uml/

9. Oktober 2017

RWTHAACHEN UNIVERSITY

Target audience

You are studying:

- Master Computer Science, or
- Master Systems Software Engineering, or
- Bachelor Computer Science, or
- ......

Usage as:

- elective course Theoretical Computer Science
- not a Wahlpflicht course for bachelor students
- specialization MOVES (Modeling and Verification of Software)
- complementary to Model-based Software Development (Rumpe)

Target audience (contd.)

In general:

- interest in system software engineering
- interest in formal methods for software
- interest in semantics and verification
- application of mathematical reasoning

Prerequisites:

- mathematical logic
- formal language and automata theory
- algorithms and data structures
- computability and complexity theory

Organization

Schedule:

	Day	Time	Room
Lecture	Mon	10:15 - 11:45	9U09
	Tue	10:15 - 11:45	9U09
Exercises	Tue	14:15- 15:45	9U09

about 21 lectures in total; Keep track of website for precise dates!

People involved:

	Lecturer	EMail
Lectures	Joost-Pieter Katoen	`katoen@cs.rwth-aachen.de`
Exercises	Tim Quatmann	`tim.quatmann@cs.rwth-aachen.de`
	Matthias Volk	`matthias.volk@cs.rwth-aachen.de`

Organization (contd.)

Assignments:

- (almost) weekly assignments
- available from course web-site
- first assignment: Tuesday October 17
- hand in solution at start next exercise class
- groups of maximally two students
- first exercise class: Tuesday October 24

Organization (contd.)

Examination: (6 ECTS credit points)

- written exam: February 6, 2018, 11:00–13:00
- written re-exam: March 12, 2018, 13:00–15:00

Admission:

- at least 40% of exercise points

Motivation

Scope:

- Goal: formal description + analysis of (concurr.) software systems
- Focus: the Unified Modeling Language

More specifically:

- Sequence Diagrams (used for requirements analysis)
- Propositional Dynamic Logic
- Communicating Finite State Automata
- Statecharts (behavioral description of systems)

Aims:

- clarify and make precise the semantics of some UML fragments
- formal reasoning about basic properties of UML models
- convince you that UML models are much harder than you think

What this course is **NOT** about:

What is it **not** about?

- the use of the UML in the software development cycle
  - see the complementary course by Prof. Rumpe
- other notations of the UML (e.g., class diagrams, activity diagrams)
- what is precisely in the UML, and what is not
  - liberal interpretation of which constructs belong to the UML
- applying the UML to concrete SW development case studies
- empirical results on the usage of UML
- drawing pictures
- ...

RWTH AACHEN UNIVERSITY

Overview

Theoretical Foundations of the UML

Lecture 2: Message Sequence Charts

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

moves.rwth-aachen.de/teaching/ss-16/theoretical-foundations-of-the-uml/

9. Oktober 2017

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 ...

Variants of MSCs

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

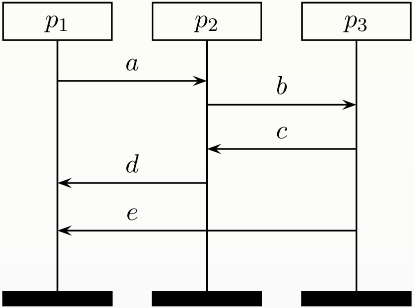
Characteristics

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

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



These pictures are formalized using partial orders.

Partial orders

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., $\forall e, e'.\, (e \preceq e' \;\wedge\; e' \preceq e) \Rightarrow e = e'$.

The pair $(E, \preceq)$ is called a partially ordered set (poset, for short).

Definition

Let $(E, \preceq)$ be a poset and let $e, e' \in E$. e and e' are comparable if $e \preceq e'$ or $e' \preceq e$. Otherwise, they are incomparable.

$\preceq$ is a non-strict partial order as it is reflexive. A strict partial order is a relation $\prec$ that is irreflexive, transitive and asymmetric (i.e., if $e \prec e'$ then not $e' \prec e$).

Hasse diagram

Definition

Let $(E, \preceq)$ be a poset.
The Hasse diagram $(E, \lessdot)$ of $(E, \preceq)$ is defined by:

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

Hasse diagrams can be used to visualize posets with finitely many elements in a succinct way.

Linearizations

Definition

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

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

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

Note that every partial order has at least one linearization.

Example

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:

$$\begin{tikzcd} & e_2 & & & \\ e_1 \arrow[ur] \arrow[dr] & & & & \\ & e_3 \arrow[r] & e_4 \arrow[r] & e_5 \arrow[r] & e_6 \end{tikzcd}$$

Linearizations:

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

No linearizations:

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

Processes and actions

Definition

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

Definition

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

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

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

Let Act denote the set of communication actions

Message Sequence Chart (MSC) (1)

Definition

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

- $\mathcal{P}$, a finite set of processes $\{p_1, p_2, \ldots, p_n\}$ with $n > 1$
- E, a finite set of events

$$E = \biguplus_{p \in \mathcal{P}} E_p = E_? \mathbin{\dot\cup} E_!$$

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

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

Message Sequence Chart (MSC) (2)

Definition

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

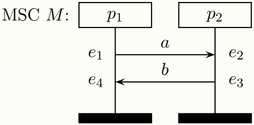
$$m(e) = e' \wedge l(e) = !(p, q, a) \text{ implies } l(e') = ?(q, p, a) \quad (p \neq q,\ a \in \mathcal{C})$$

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

$$\preceq \ = \Big(\underbrace{\bigcup_{p \in \mathcal{P}} <_p}_{<_p \text{ is a total order } = \text{ "top-to-bottom" order on process } p} \ \cup \ \underbrace{\{(e, m(e)) \mid e \in E_!\}}_{\text{communication order } <_c} \Big)^*$$

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

Example (1)



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

$$
\begin{array}{lcll}
\mathcal{P} & = & \{p_1, p_2\} & E_{p_1} = \{e_1, e_4\} \\
E & = & \{e_1, e_2, e_3, e_4\} & E_{p_2} = \{e_2, e_3\} \\
\mathcal{C} & = & \{a, b\} & E_! = \{e_1, e_3\}, \\
 & & & E_? = \{e_2, e_4\}
\end{array}
$$

$$
\begin{array}{ll}
l(e_1) = !(p_1, p_2, a) & m(e_1) = e_2 \\
l(e_2) = ?(p_2, p_1, a) & \\
l(e_3) = !(p_2, p_1, b) & m(e_3) = e_4 \\
l(e_4) = ?(p_1, p_2, b) &
\end{array}
$$

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

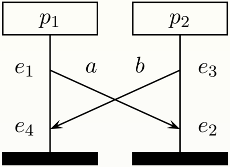
Hasse diagram of $(E, \preceq)$:

$$\boxed{e_1 \longrightarrow e_2 \longrightarrow e_3 \longrightarrow e_4}$$

Linearizations?

Example (2)

MSC M':



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

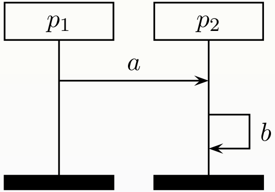
$<'_c$: $e_1 \longrightarrow e_2$, $e_3 \longrightarrow e_4$

$<'_{p_1}$: $e_1 \longrightarrow e_4$

$<'_{p_2}$: $e_3 \longrightarrow e_2$

$\preceq'$: $e_1 \longrightarrow e_2$, $e_1 \longrightarrow e_4$, $e_3 \longrightarrow e_2$, $e_3 \longrightarrow e_4$

This is not an MSC

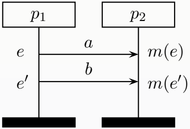


FIFO property

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

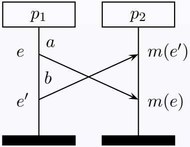
$$e \prec e' \wedge l(e) = !(p, q, a) \wedge l(e') = !(p, q, b) \text{ implies } m(e) \prec m(e')$$

i.e., "no message overtaking allowed"



FIFO

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



non-FIFO

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

Linearizations

Definition

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

MSCs and its linearizations are interchangeable

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

Thus:

$Lin(M)$ uniquely characterizes the MSC M.

From MSCs to its set of linearizations is straightforward. The reverse direction is discussed in the following. First: well-formedness.

Well-formedness

Let $Ch := \{(p,q) \mid p \neq q\,,\, p,q \in \mathcal{P}\}$ be the 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

Proposition

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

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

From linearizations to posets

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

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

such that:

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

$$\ell(i) = !(p, q, m) \text{ and } \ell(j) = ?(q, p, m) \text{ and}$$

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

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 every 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 $f : E \rightarrow E'$ such that $e \preceq e'$ iff $f(e) \preceq' f(e')$ and $\ell(e) = \ell'(f(e))$.

Linearizations yield isomorphic MSCs

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

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