Softwaretechnik / Software-Engineering

Lecture 08: Scenarios and Use Cases

2015-06-08

Prof. Dr. Andreas Podelski, **Dr. Bernd Westphal**

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

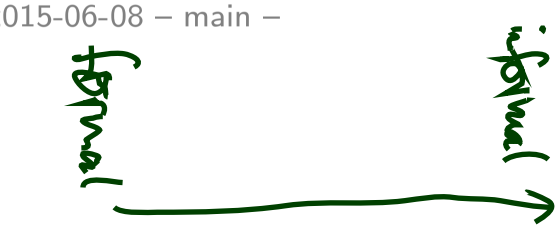
Last Lecture:

- Consistency, Completeness, etc. for Decision Tables.

This Lecture:

- **Educational Objectives:** Capabilities for following tasks/questions.
  - What is a scenario/an anti-scenario?
  - What is included in a use case? In a use case diagram?
  - What is the abstract syntax of this Live Sequence Chart (LSC)?
  - Which are the cuts and firedsets of this LSC?
  - Construct the TBA of a given LSC body.
  - Given a set of LSCs, which scenario/anti-scenario/requirement is formalised by them?
  - Formalise this positive scenario/anti-scenario/requirement using LSCs.
- **Content:**
  - Scenarios in Requirements Engineering
  - User Stories; Use Cases and Diagrams
  - Live Sequence Charts

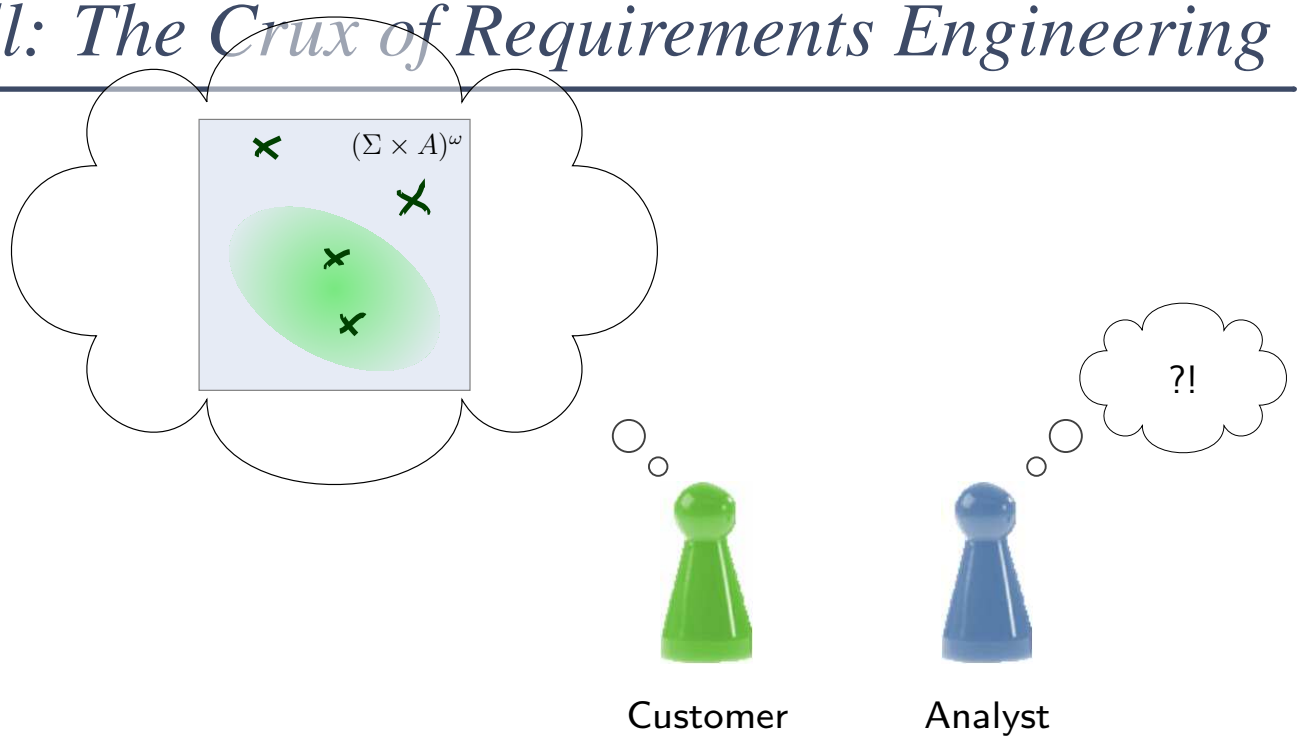
You Are Here



Introduction	L 1:	20.4., Mo
	T 1:	23.4., Do
Development Process, Metrics	L 2:	27.4., Mo
	L 3:	30.4., Do
	L 4:	4.5., Mo
	T 2:	7.5., Do
	L 5:	11.5., Mo
	-	14.5., Do
Requirements Engineering	L 6:	18.5., Mo
	L 7:	21.5., Do
	-	25.5., Mo
	-	28.5., Do
	T 3:	1.6., Mo
	-	4.6., Do
	L 8:	8.6., Mo
Architecture & Design	L 9:	11.6., Do
	L 10:	15.6., Mo
	T 4:	18.6., Do
Constructive Models	L 11:	22.6., Mo
	L 12:	25.6., Do
	L 13:	29.6., Mo
	T 5:	2.7., Do
Testing, Formal Verification	L 14:	6.7., Mo
	L 15:	9.7., Do
	L 16:	13.7., Mo
Invited Talks	L 17:	16.7., Do
	T 6:	20.7., Mo
Wrap-Up	L 18:	23.7., Do

Scenarios

Recall: The Crux of Requirements Engineering



One quite effective approach:
try to **approximate** the requirements with positive and negative **scenarios**.

- Dear customer, please describe example usages of the desired system.

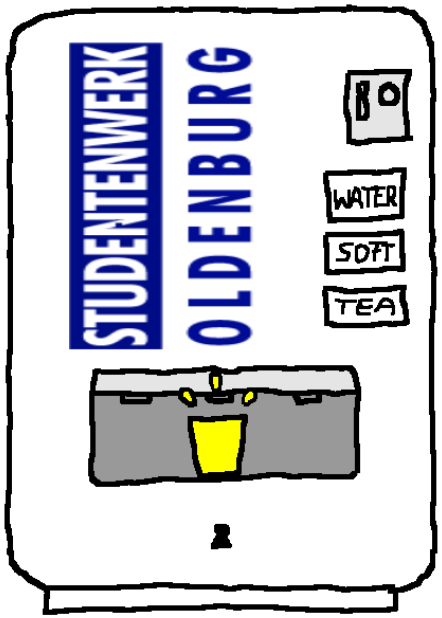
  "If the system is not at all able to do this, then it's not what I want."

- Dear customer, please describe behaviour that the desired system must not show.

  "If the system does this, then it's not what I want."

- From there on, refine and generalise:
  what about exceptional cases? what about corner-cases? etc.

Example: Vending Machine



- Positive scenario:
  - (i) Insert one 1 euro and one 50 cent coin.
  - (ii) Press the ‘softdrink’ button.
  - (iii) Get a softdrink.
  - (iv) Get 50 cent change.
- Negative scenario:
  - (i) After switching on, insert no money.
  - (ii) Press the ‘tea’ button.
  - (iii) Get a tea.
  - (iv) Get 100 € change.

Notations for Scenarios

- The idea of scenarios (sometimes without negative or anti-scenarios) (re-)occurs in many process models or software development approaches.
- First prominent recognition: OOSE (Jacobson, 1992)
- In the following, we will discuss two and a half notations (in increasing formality):
  - **User Stories** (part of **Extreme Programming**)
  - **Use Cases** and Use Case Diagrams (**OOSE**)
  - **Sequence Diagrams** (here: **Live Sequence Charts** (Damm and Harel, 2001))

User Stories

User Stories (Beck, 1999)

"A User Story is a concise, written description of a piece of functionality that will be valuable to a user (or owner) of the software."

Per user story, one **file card** with the user story, e.g. following the pattern:

- *As a [role] I want [something] so that [benefit].*

and in addition:

- **unique identifier** (e.g. unique number),
- **priority** (from 1 (highest) to 10 (lowest)) assigned by customer,
- **effort**, estimated by developers,
- back side of file card:
  (acceptance) **test case(s)** — how to tell whether the user story has been realised.

User Stories: Discussion

Proposed card layout (front side):

priority	unique identifier, name	estimation
As a [role] I want [something] so that [benefit].		
risk		real effort

- ✔ easy to create
- ✔ close contact to customer
- ✘ customers are usually not trained in writing requirements
- ✘ may get difficult to keep overview over whole system to be developed
- ✘ strong dependency on competent developers
- ✘ estimation of effort may be difficult
- ✘ not easy to cover non-functional requirements and restrictions

(Balzert, 2009)

Recall:

Natural Language Patterns

Natural language requirements can be written using A, B, C, D, E, F where

A	clarifies when and under what conditions the activity takes place
B	is MUST (obligation), SHOULD (wish), or WILL (intention); also: MUST NOT (forbidden)
C	is either "the system" or the concrete name of a (sub-)system
D	one of three possibilities: • "does", description of a system activity, • "offers", description of a function offered by the system to somebody, • "is able if", usage of a function offered by a third party, under certain conditions
E	extensions, in particular an object
F	the actual process word (what happens)

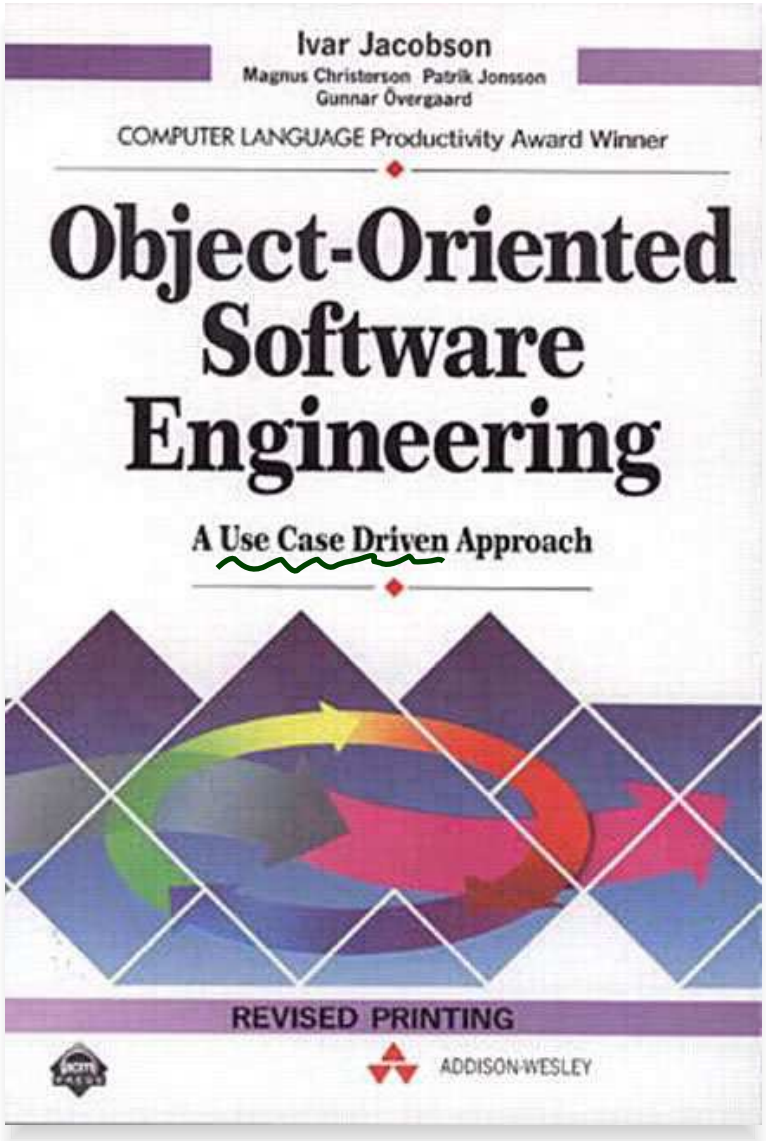
(Rupp and die SOPHISTen, 2009)

Example:

After office hours ($= A$), the system ($= C$) should ($= B$) offer to the operator ($= D$) a backup ($= F$) of all new registrations to an external medium ($= E$).

37/90

Use Cases


Ivar Jacobson
Magnus Christerson Patrik Jonsson
Gunnar Övergaard
COMPUTER LANGUAGE Productivity Award Winner
Object-Oriented Software Engineering
A Use Case Driven Approach
REVISED PRINTING
ADDISON-WESLEY

Use Case: Definition

use case — A sequence of interactions between an actor (or actors) and a system triggered by a specific actor, which produces a result for an actor.

(Jacobson, 1992)

- **participants**: the **system** and at least one **actor**,
- **actor**: an actor represents what interacts with the system.
  - An actor is a **role**, which a **user** or an **external system** may assume when interacting with the system under design.
  - Actors are not part of the system, thus they are **not described in detail**.
  - Actions of actors are **non-deterministic** (possibly constrained by domain model).
- A use case is triggered by a **stimulus** as input by the **main actor**.
- A use case is **goal oriented**, i.e. the main actor wants to reach a particular goal.
- A use case describes **all interactions** between the system and the participating actors that are needed to achieve the goal (or fail to achieve the goal for reasons).
- A use case **ends** when the desired goal is achieved, or when it is clear that the desired goal cannot be achieved.

Use Case Example

name	Authentication
goal	the client wants access to the ATM
pre-condition	the ATM is operational, the welcome screen is displayed, card and PIN of client are available
post-condition	client accepted, services of ATM are offered
post-cond. in exceptional case	access denied, card returned or withheld, welcome screen displayed
actors	client (main actor), bank system
open questions	none
normal case	1. client inserts card 2. ATM read card, sends data to bank system 3. bank system checks validity 4. ATM shows PIN screen 5. client enters PIN 6. ATM reads PIN, sends to bank system 7. bank system checks PIN 8. ATM accepts and shows main menu
exception case 2a	**card not readable** 2a.1 ATM displays "card not readable" 2a.2 ATM returns card 2a.3 ATM shows welcome screen

http://commons.wikimedia.org (CC-by-sa 4.0, Dirk Ingo Franke)

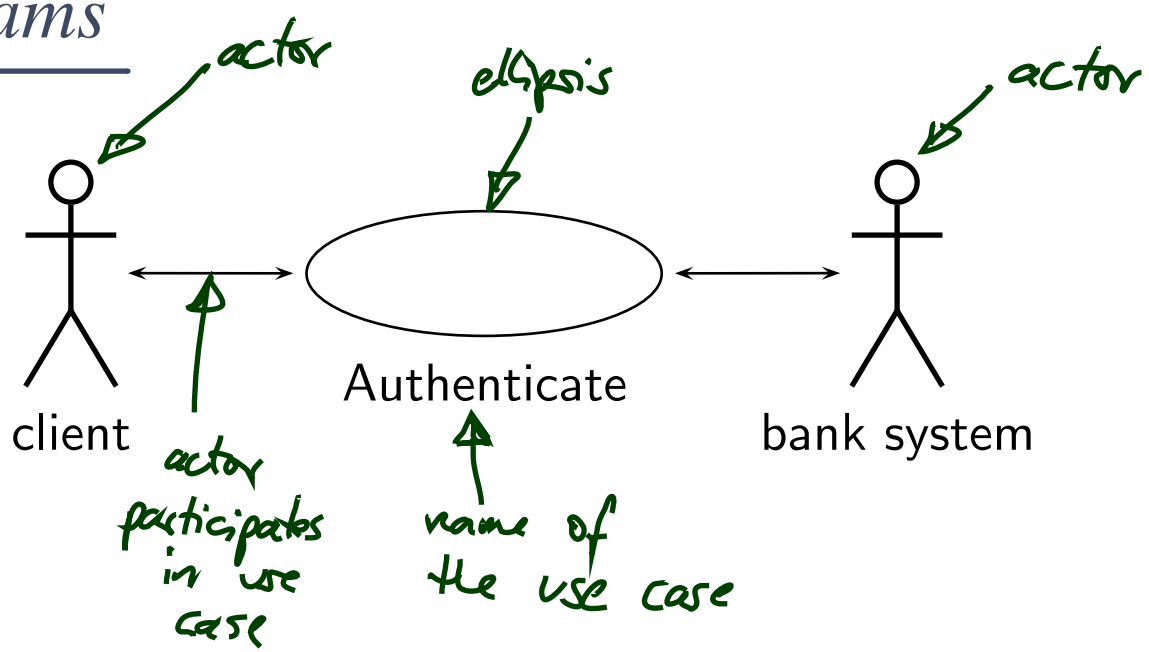
open questions	none
normal case	1. client inserts card 2. ATM read card, sends data to bank system 3. bank system checks validity 4. ATM shows PIN screen 5. client enters PIN 6. ATM reads PIN, sends to bank system 7. bank system checks PIN 8. ATM accepts and shows main menu
exception case 2a	**card not readable** 2a.1 ATM displays “card not readable” 2a.2 ATM returns card 2a.3 ATM shows welcome screen
exception case 2b	**card readable, but not ATM card**
exception case 2c	**no connection to bank system**
exception case 3a	**card not valid or disabled**
exception case 5a	**client cancels**
exception case 5b	**client doesn’t react within 5 s**
exception case 6a	**no connection to bank system**
exception case 7a	**first or second PIN wrong**
exception case 7b	**third PIN wrong**

http://commons.wikimedia.org (CC-by-sa 4.0, Dirk Ingo Franke)

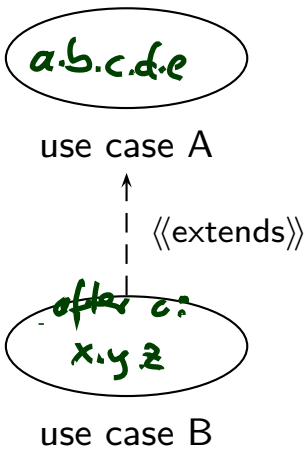
(Ludewig and Lichter, 2013; V-Modell XT, 2006)

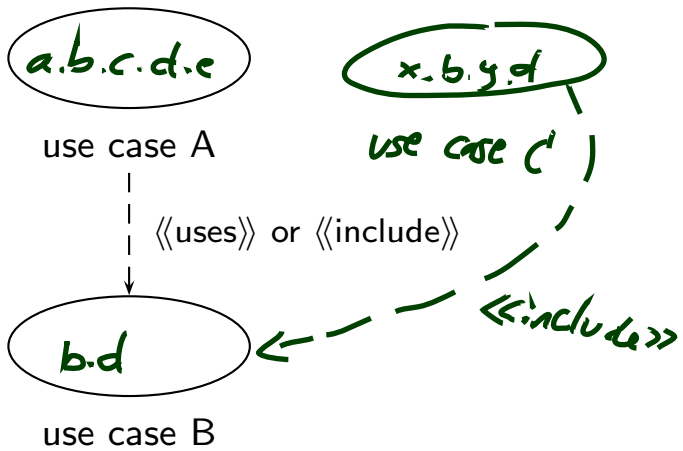
Use Case Diagrams

Use Case Diagrams

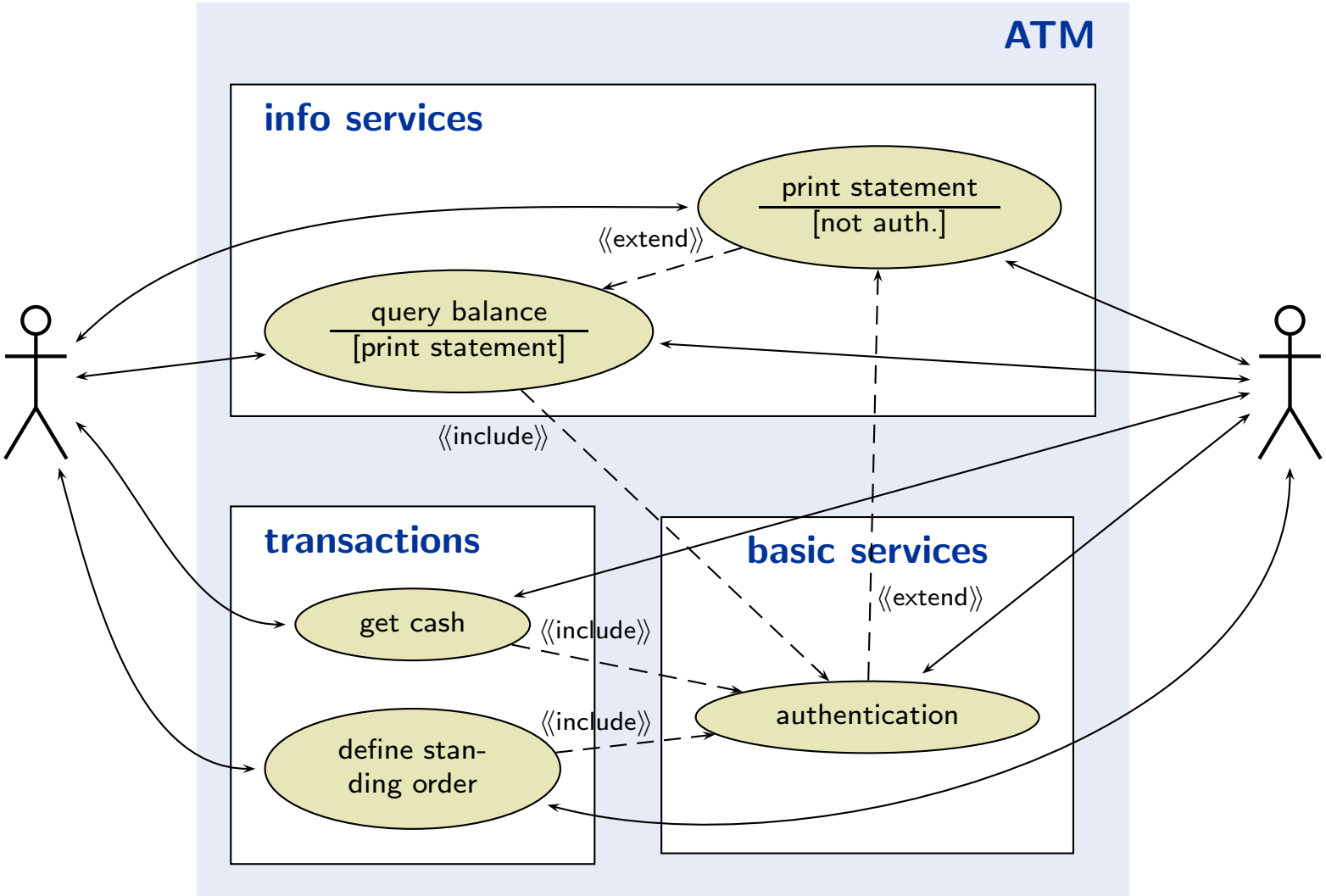


More notation:



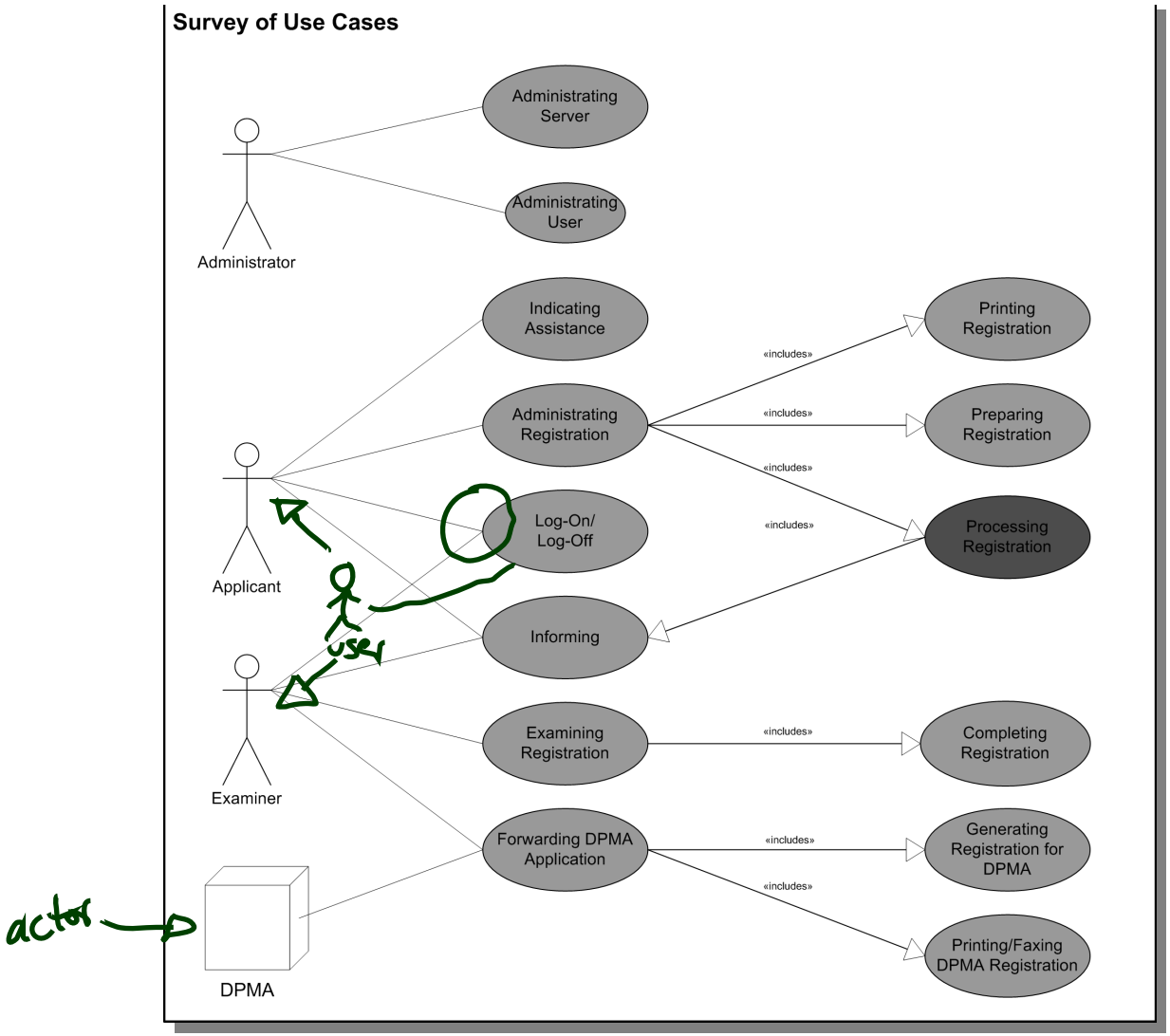


Use Case Diagram: Bigger Examples



(Ludewig and Lichter, 2013)

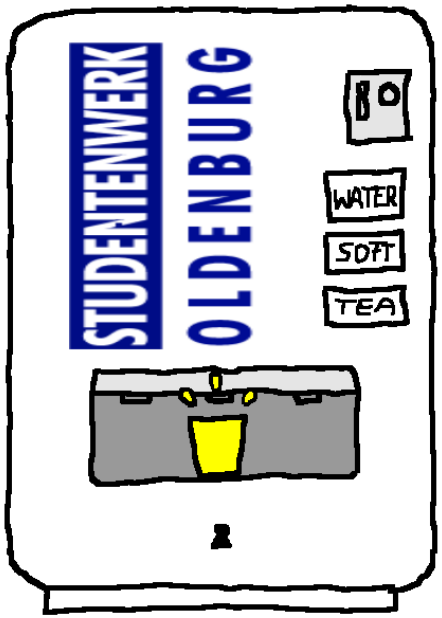
Use Case Diagram: Bigger Examples



(V-Modell XT, 2006)

Sequence Diagrams

Recall: Vending Machine Example



- Positive scenario:
  - (i) Insert one 1 euro and one 50 cent coin.
  - (ii) Press the ‘softdrink’ button.
  - (iii) Get a softdrink.
  - (iv) Get 50 cent change.
- Negative scenario:
  - (i) After switching on, insert no money.
  - (ii) Press the ‘tea’ button.
  - (iii) Get a tea.
  - (iv) Get 100 € change.

notation	understandable by customer	precision	complexity of definition
natural language	feels like ++	--	+++···
visual formalism	**(+)**	**+ +**	**+ +**
(temporal) logic	--	++	-

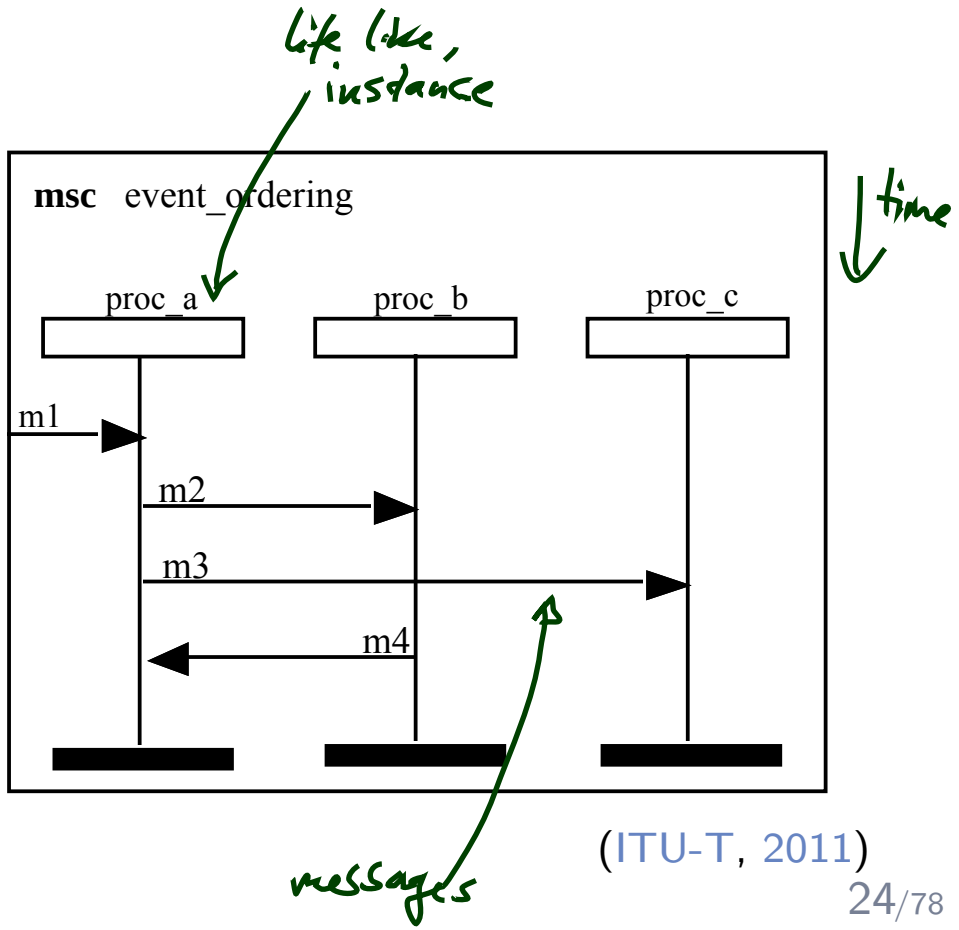
History

Most Prominent: Sequence Diagrams — with **long history**:

- **Message Sequence Charts**,
  standardized by the ITU in different versions (ITU Z.120, 1st edition: 1993),
  often accused to lack a formal semantics.
- **Sequence Diagrams** of UML 1.x (one of three main authors: I. Jacobson)

Most severe **drawbacks** of these formalisms:

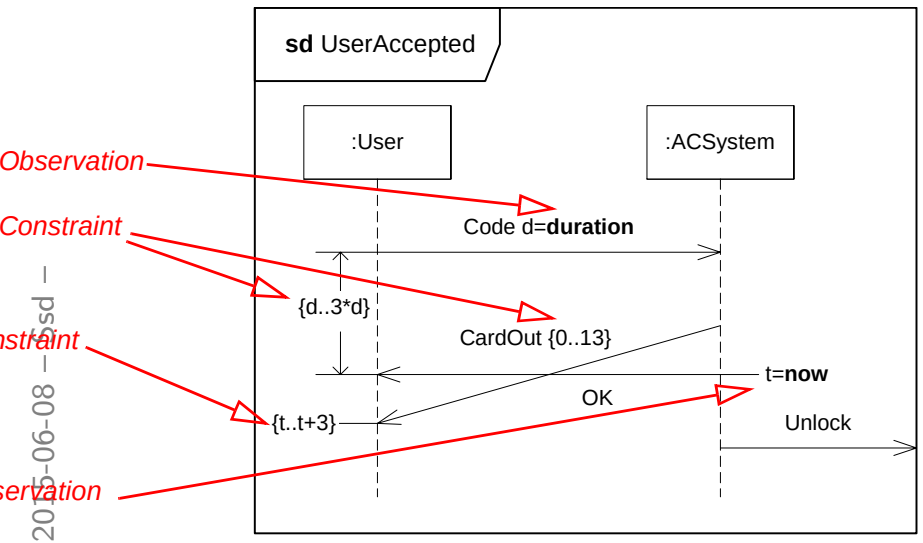
- unclear **interpretation**:
  example scenario or invariant?
- unclear **activation**:
  what triggers the requirement?
- unclear **progress** requirement:
  must all messages be observed?
- **conditions** merely comments
- no means (in language) to express
  forbidden scenarios

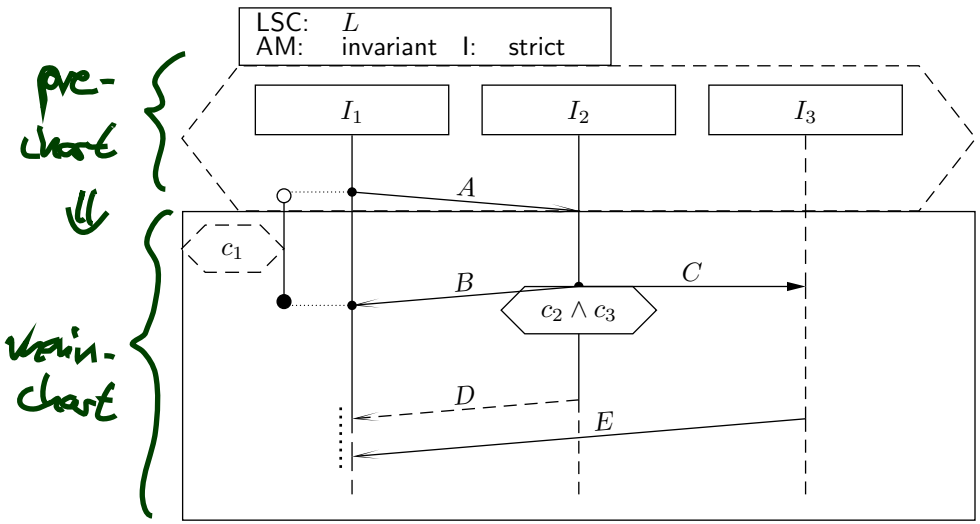


(ITU-T, 2011)

Thus: Live Sequence Charts

- **SDs of UML 2.x** address **some** issues, yet the standard exhibits unclarities and even contradictions (Harel and Maoz, 2007; Störrle, 2003)
- For the lecture, we consider **Live Sequence Charts** (LSCs) (Damm and Harel, 2001; Klose, 2003; Harel and Marelly, 2003), who have a common fragment with UML 2.x SDs (Harel and Maoz, 2007)
- **Modelling guideline**: stick to that fragment.





(OMG, 2007)

The Plan

LSCs as another **formal software specification**:

- LSC body abstract and concrete **syntax**,
- **Excursion**: Büchi automata (TBA),
- **TBA construction**: Cuts, Firedsets, Transitions,
- **Language** of an LSC body,
- **Putting it all together**:
  Activation modes, Pre-charts, Existential and universal LSCs,
- Some bigger LSC **examples**,
- Discussion of **MSC** issues named on previous slide.

- Some concluding notes on the block Requirements Engineering.

- **Next block**: design, architecture...

Live Sequence Charts: Syntax (Body)

LSC Body: Abstract Syntax

Definition. [*LSC Body*] Let $\mathcal{E}$ be a set of **events** and C a set of **atomic propositions**. An **LSC body** over $\mathcal{E}$ and C is a tuple

$$((\mathcal{L}, \preceq, \sim), \mathcal{I}, \mathsf{Msg}, \mathsf{Cond}, \mathsf{LocInv}, \Theta)$$

where

- $\mathcal{L}$ is a finite, non-empty of **locations** with
  - a **partial order** $\preceq \subseteq \mathcal{L} \times \mathcal{L}$,
  - a symmetric **simultaneity relation** $\sim \subseteq \mathcal{L} \times \mathcal{L}$ disjoint with $\preceq$, i.e. $\preceq \cap \sim = \emptyset$,
- $\mathcal{I} = \{I_1, \ldots, I_n\}$ is a partitioning of $\mathcal{L}$; elements of $\mathcal{I}$ are called **instance line**,
- $\mathsf{Msg} \subseteq \mathcal{L} \times \mathcal{E} \times \mathcal{L}$ is a set of **messages** with $(l, E, l') \in \mathsf{Msg}$ only if $(l, l') \in \prec \cup \sim$; message (l, E, l') is called **instantaneous** iff $l \sim l'$ and **asynchronous** otherwise,
- $\mathsf{Cond} \subseteq (2^{\mathcal{L}} \setminus \emptyset) \times \Phi(C)$ is a set of **conditions** with $(L, \phi) \in \mathsf{Cond}$ only if $l \sim l'$ for all $l \neq l' \in L$,
- $\mathsf{LocInv} \subseteq \mathcal{L} \times \{\circ, \bullet\} \times \Phi(C) \times \mathcal{L} \times \{\circ, \bullet\}$ is a set of **local invariants** with $(l, \iota, \phi, l', \iota') \in \mathsf{LocInv}$ only if $l \prec l'$, $\circ$: exclusive, $\bullet$: inclusive,
- $\Theta : \mathcal{L} \cup \mathsf{Msg} \cup \mathsf{Cond} \cup \mathsf{LocInv} \to \{\mathsf{hot}, \mathsf{cold}\}$ assigns to each location and each element a **temperature**.

powerset of $\mathcal{L}$

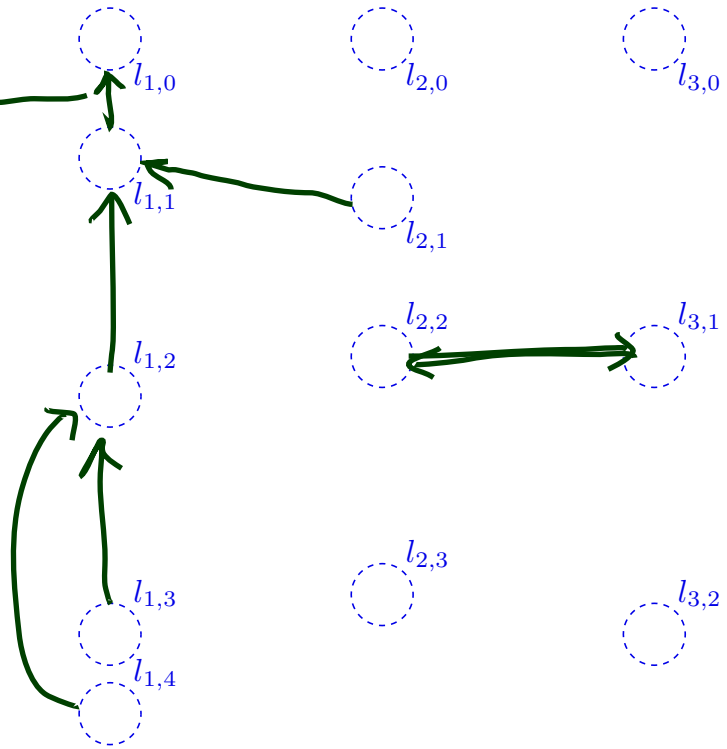
prop. formula over C

LSC Body Example

- $\mathcal{L} : l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,2} \sim l_{3,1},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$
- $\mathsf{Msg} = \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{1,3}), (l_{3,2}, E, l_{1,4})\}$
- $\mathsf{Cond} = \{(\{l_{2,2}\}, c_2 \wedge c_3)\},$
- $\mathsf{LocInv} = \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$

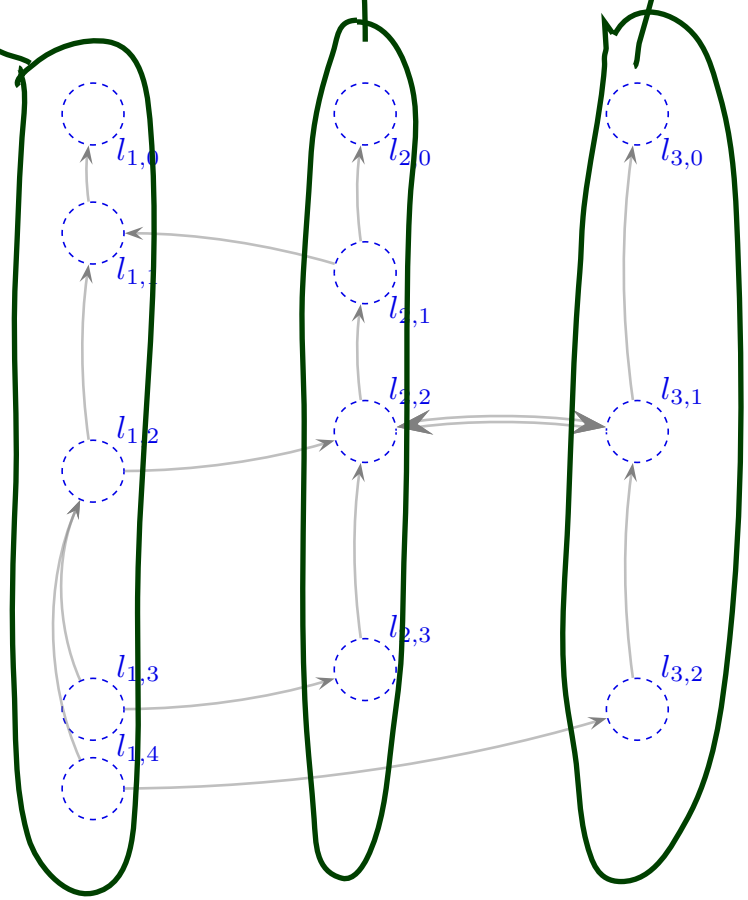
LSC Body Example

- $\mathcal{L}: l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,2} \sim l_{3,1},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$
- Msg $= \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{1,3}), (l_{3,2}, E, l_{1,4})\}$
- Cond $= \{(\{l_{2,2}\}, c_2 \wedge c_3)\},$
- LocInv $= \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$



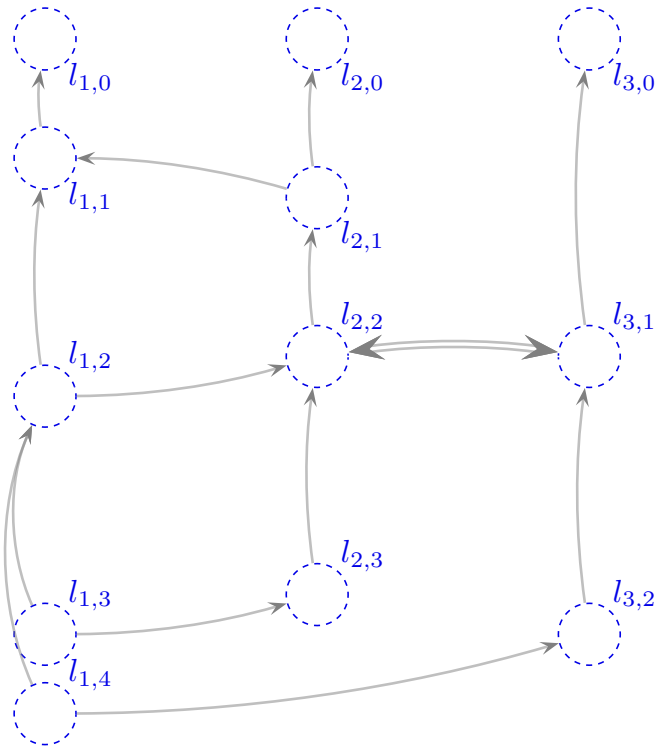
LSC Body Example

- $\mathcal{L} : l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,2} \sim l_{3,1},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$
- Msg $= \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{1,3}), (l_{3,2}, E, l_{1,4})\}$
- Cond $= \{(\{l_{2,2}\}, c_2 \wedge c_3)\},$
- LocInv $= \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$



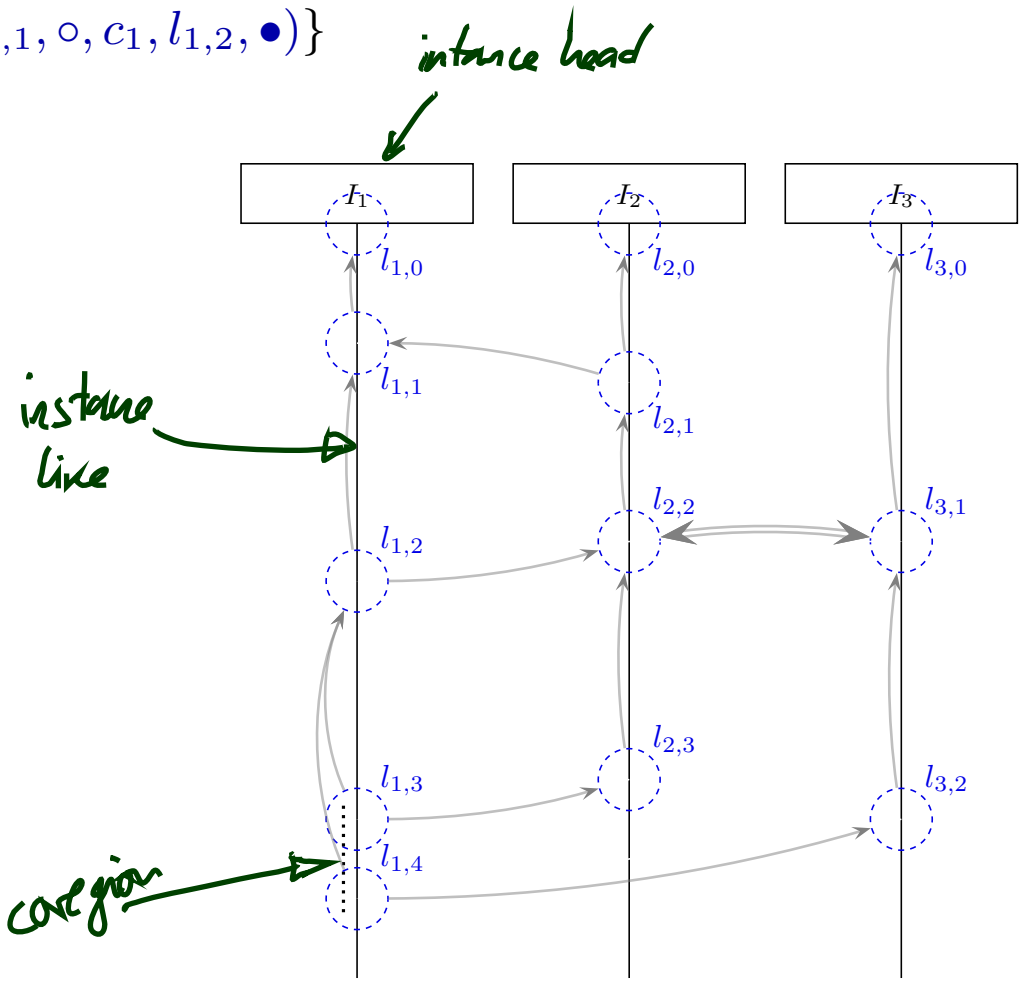
LSC Body Example

- $\mathcal{L} : l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,2} \sim l_{3,1},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$
- $\mathsf{Msg} = \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{1,3}), (l_{3,2}, E, l_{1,4})\}$
- $\mathsf{Cond} = \{(\{l_{2,2}\}, c_2 \wedge c_3)\},$
- $\mathsf{LocInv} = \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$



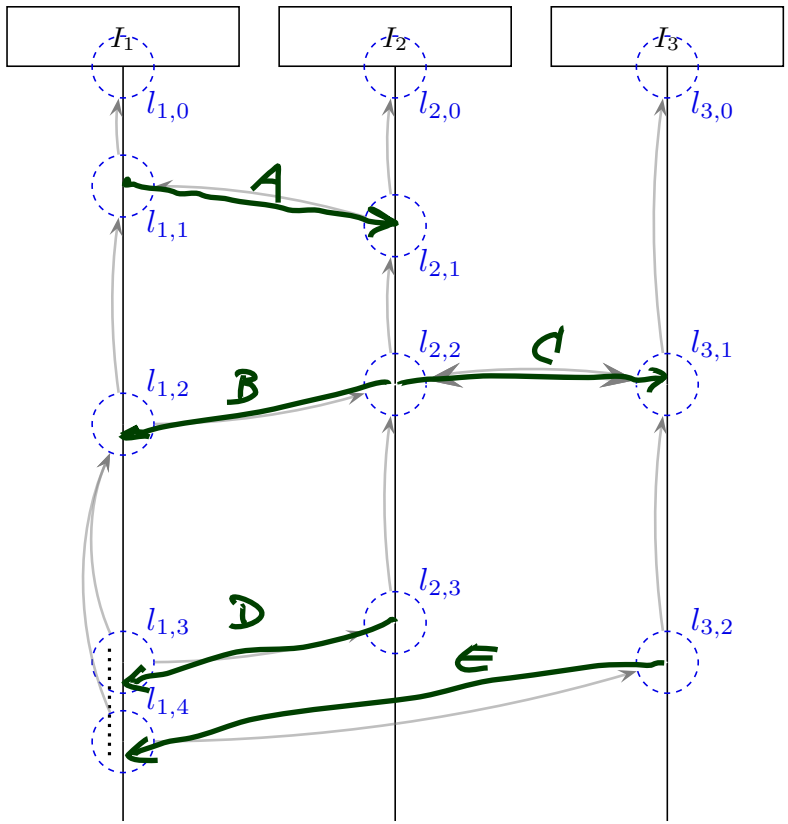
LSC Body Example

- $\mathcal{L} : l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,2} \sim l_{3,1},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$
- $\mathsf{Msg} = \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{1,3}), (l_{3,2}, E, l_{1,4})\}$
- $\mathsf{Cond} = \{(\{l_{2,2}\}, c_2 \wedge c_3)\},$
- $\mathsf{LocInv} = \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$



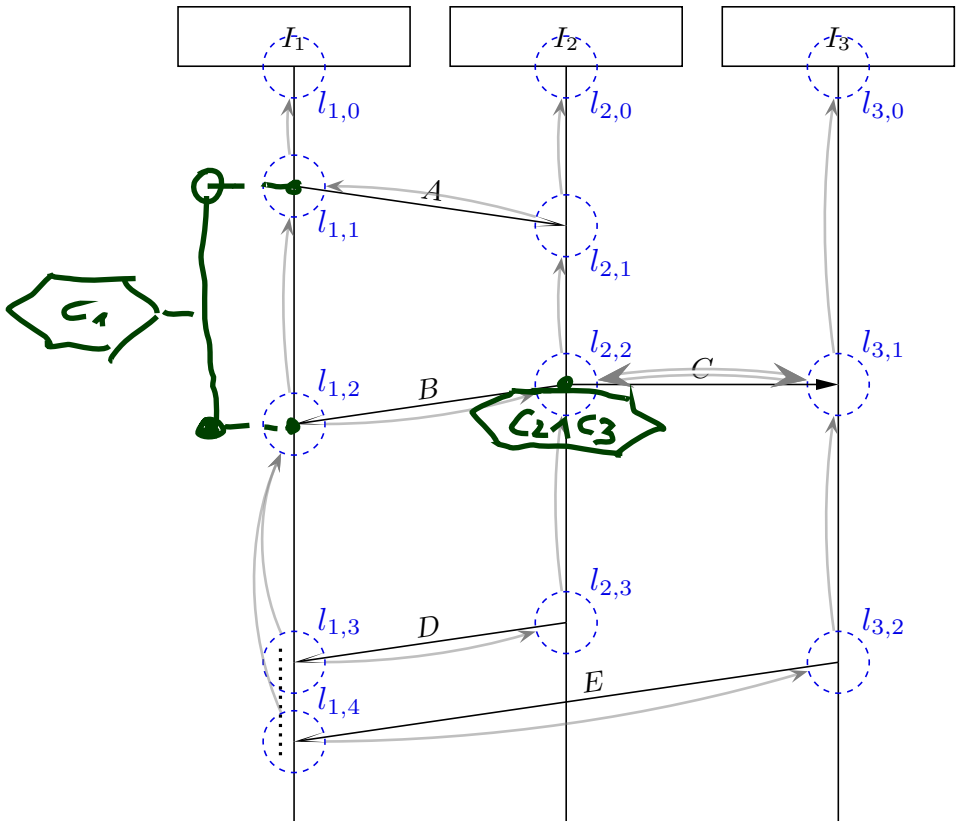
LSC Body Example

- $\mathcal{L} : l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,2} \sim l_{3,1},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$
- Msg $= \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{1,3}), (l_{3,2}, E, l_{1,4})\}$
- Cond $= \{(\{l_{2,2}\}, c_2 \wedge c_3)\},$
- LocInv $= \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$



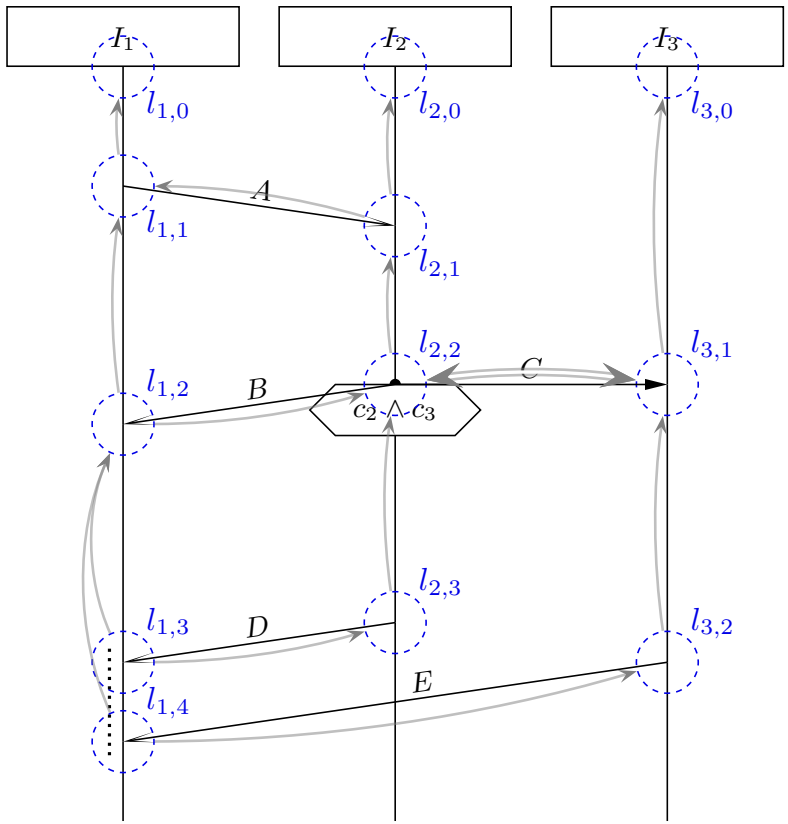
LSC Body Example

- $\mathcal{L} : l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,2} \sim l_{3,1},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$
- Msg $= \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{1,3}), (l_{3,2}, E, l_{1,4})\}$
- Cond $= \{(\{l_{2,2}\}, c_2 \wedge c_3)\},$
- LocInv $= \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$



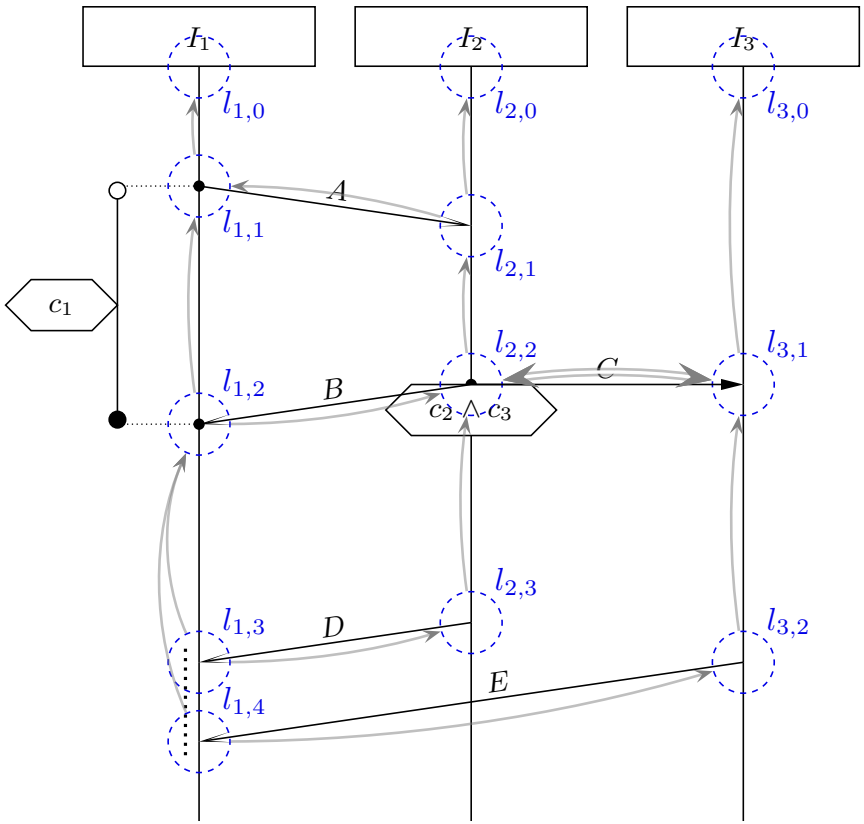
LSC Body Example

- $\mathcal{L} : l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,2} \sim l_{3,1},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$
- $\mathsf{Msg} = \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{1,3}), (l_{3,2}, E, l_{1,4})\}$
- $\mathsf{Cond} = \{(\{l_{2,2}\}, c_2 \wedge c_3)\},$
- $\mathsf{LocInv} = \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$



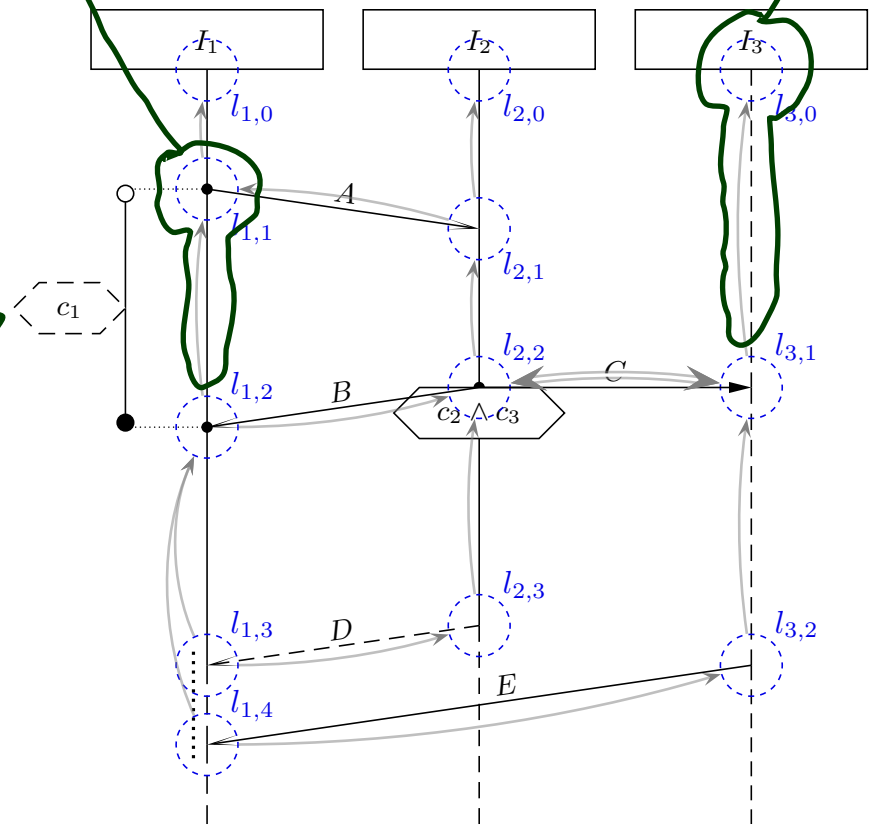
LSC Body Example

- $\mathcal{L} : l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,2} \sim l_{3,1},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$
- Msg $= \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{1,3}), (l_{3,2}, E, l_{1,4})\}$
- Cond $= \{(\{l_{2,2}\}, c_2 \wedge c_3)\},$
- LocInv $= \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$



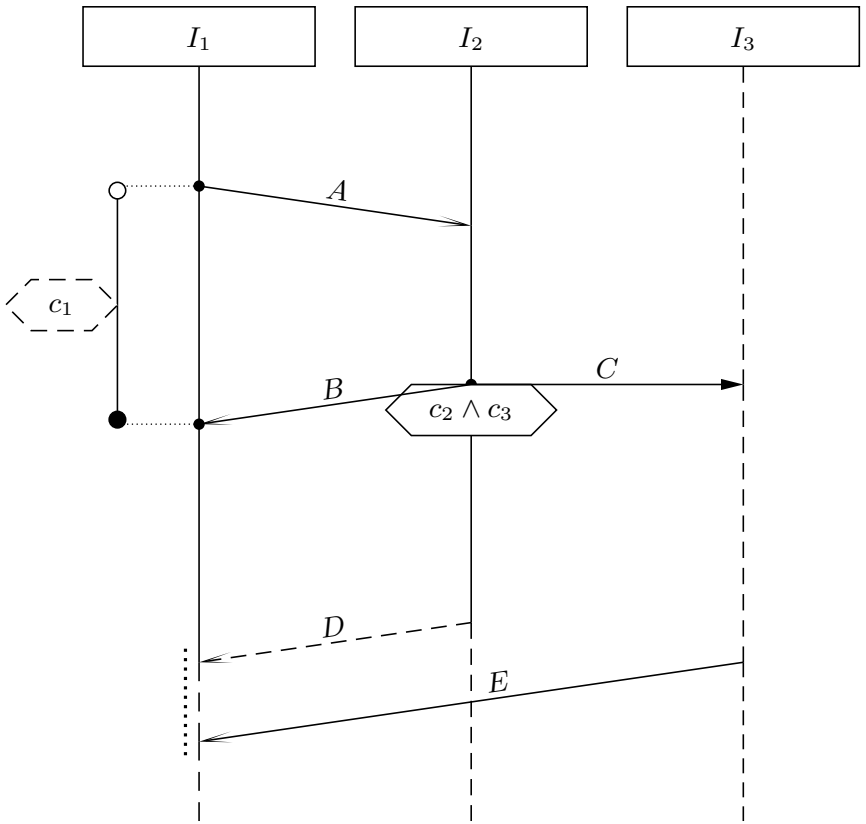
LSC Body Example

- $\mathcal{L} : l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,2} \sim l_{3,1},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$
- Msg $= \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{1,3}), (l_{3,2}, E, l_{1,4})\}$
- Cond $= \{(\{l_{2,2}\}, c_2 \wedge c_3)\},$
- LocInv $= \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$

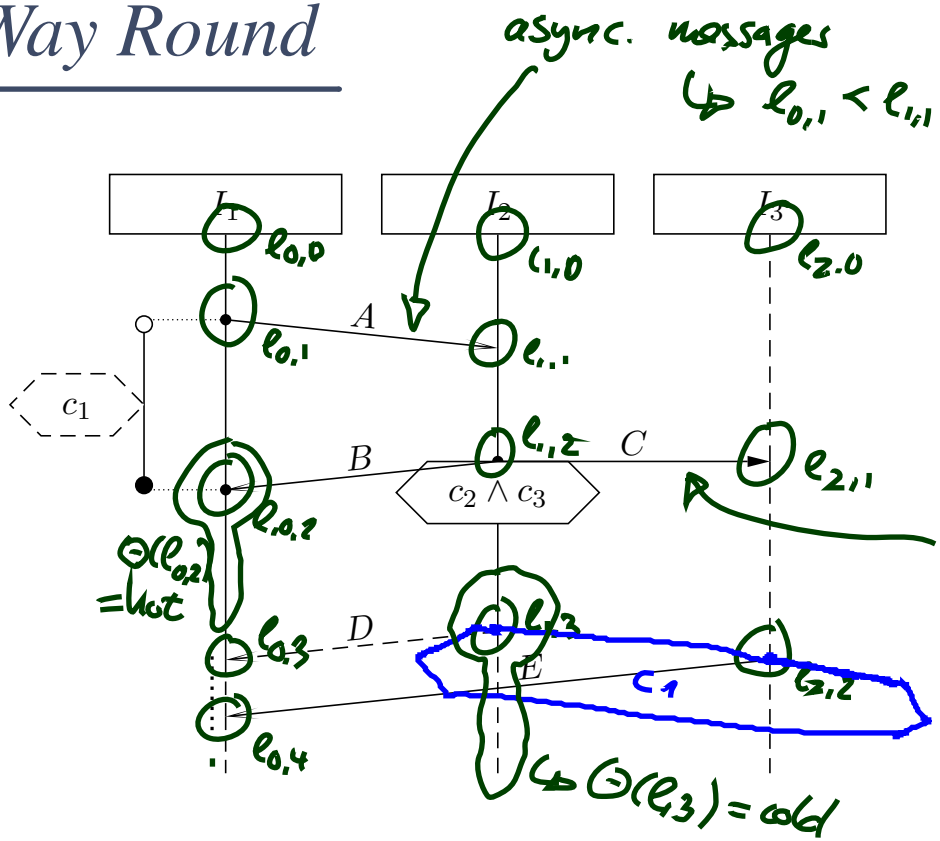


LSC Body Example

- $\mathcal{L} : l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,2} \sim l_{3,1},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$
- $\mathsf{Msg} = \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{1,3}), (l_{3,2}, E, l_{1,4})\}$
- $\mathsf{Cond} = \{(\{l_{2,2}\}, c_2 \wedge c_3)\},$
- $\mathsf{LocInv} = \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$



And The Other Way Round

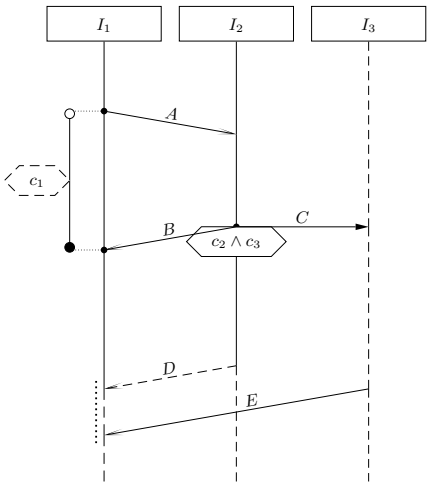


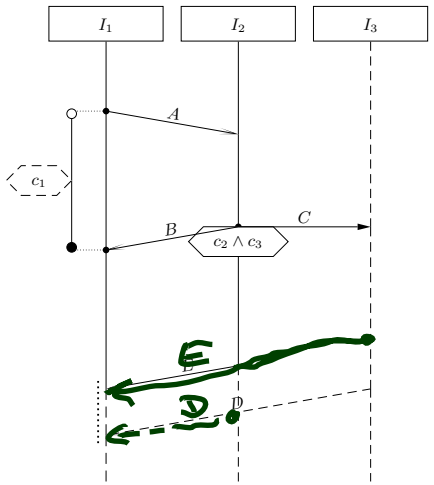
- $\mathcal{L}$: $\ell_{0,0} < \ell_{0,1} < \ell_{0,2} < \ell_{0,3}$, $\ell_{0,2} < \ell_{0,4}$ (along instance line)

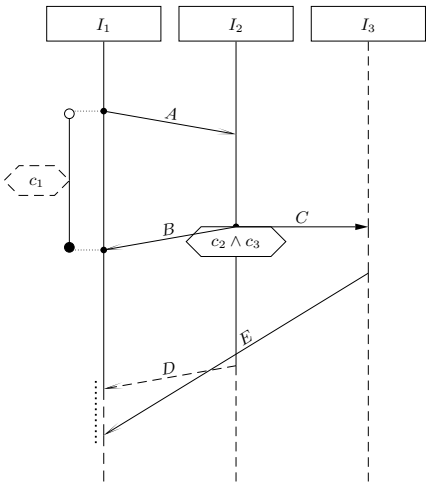
- $\mathcal{I} = \{\{\ell_{0,0}, .., \ell_{0,4}\}, \{\ell_{1,0}, .., \ell_{1,3}\}, \{\ell_{2,0}, .., \ell_{2,2}\}\}$
- Msg $= \{(\ell_{0,1}, A, \ell_{1,1}), \ldots\}$
- Cond $= \{(\{\ell_{1,2}\}, c_2 \wedge c_3)\}$
- LocInv $= \{(\ell_{1,0}, c_1, \ell_{0,2}, \bullet)\}$

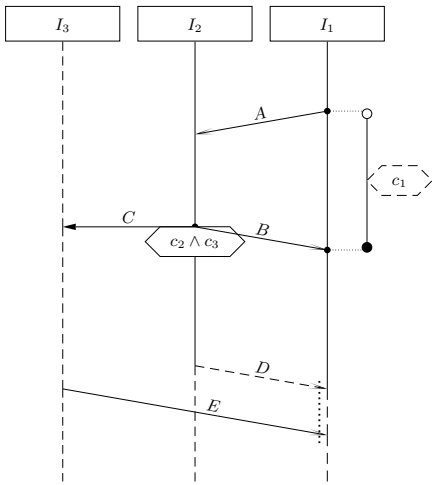
Concrete vs. Abstract Syntax

- $\mathcal{L} : l_{1,0} \prec l_{1,1} \prec l_{1,2} \prec l_{1,3}, \quad l_{1,2} \prec l_{1,4}, \quad l_{2,0} \prec l_{2,1} \prec l_{2,2} \prec l_{2,3}, \quad l_{3,0} \prec l_{3,1} \prec l_{3,2},$
  $l_{1,1} \prec l_{2,1}, \quad l_{2,2} \prec l_{1,2}, \quad l_{2,3} \prec l_{1,3}, \quad l_{3,2} \prec l_{1,4}, \quad l_{2,2} \sim l_{3,1},$
- $\mathcal{I} = \{\{l_{1,0}, l_{1,1}, l_{1,2}, l_{1,3}, l_{1,4}\}, \{l_{2,0}, l_{2,1}, l_{2,2}, l_{2,3}\}, \{l_{3,0}, l_{3,1}, l_{3,2}\}\},$
- $\mathsf{Msg} = \{(l_{1,1}, A, l_{2,1}), (l_{2,2}, B, l_{1,2}), (l_{2,2}, C, l_{3,1}), (l_{2,3}, D, l_{1,3}), (l_{3,2}, E, l_{1,4})\}$
- $\mathsf{Cond} = \{(\{l_{2,2}\}, c_2 \wedge c_3)\},$
- $\mathsf{LocInv} = \{(l_{1,1}, \circ, c_1, l_{1,2}, \bullet)\}$





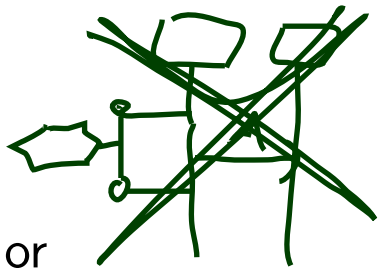




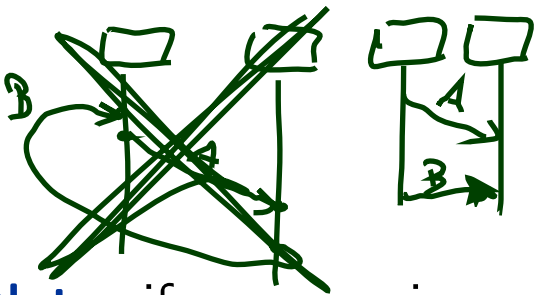
Well-Formedness

Bondedness/no floating conditions: (could be relaxed a little if we wanted to)

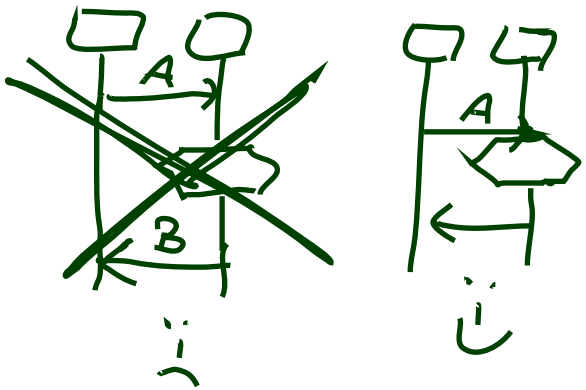
- For each location $l \in \mathcal{L}$, **if** l is the location of
  - a **condition**, i.e. $\exists\,(L, \phi) \in \mathsf{Cond} : l \in L$, or
  - a **local invariant**, i.e. $\exists\,(l_1, \iota_1, \phi, l_2, \iota_2) \in \mathsf{LocInv} : l \in \{l_1, l_2\}$, or



  then there is a location l' **simultaneous** to l, i.e. $l \sim l'$, which is the location of
  - an **instance head**, i.e. l' is minimal wrt. $\preceq$, or
  - a **message**, i.e.



$$\exists\,(l_1, E, l_2) \in \mathsf{Msg} : l \in \{l_1, l_2\}.$$



Note: if messages in a chart are **cyclic**, then there doesn't exist a partial order (so such charts **don't even have** an abstract syntax).

References

References

Balzert, H. (2009). *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. Spektrum, 3rd edition.

Beck, K. (1999). *Extreme Programming Explained – Embrace Change*. Addison-Wesley.

Damm, W. and Harel, D. (2001). LSCs: Breathing life into Message Sequence Charts. *Formal Methods in System Design*, 19(1):45–80.

Harel, D. and Maoz, S. (2007). Assert and negate revisited: Modal semantics for UML sequence diagrams. *Software and System Modeling (SoSyM)*. To appear. (Early version in SCESM'06, 2006, pp. 13-20).

Harel, D. and Marelly, R. (2003). *Come, Let's Play: Scenario-Based Programming Using LSCs and the Play-Engine*. Springer-Verlag.

ITU-T (2011). *ITU-T Recommendation Z.120: Message Sequence Chart (MSC)*, 5 edition.

Jacobson, I. (1992). *Object Oriented Software Engineering – A Use Case Driven Approach*. ACM Press.

Klose, J. (2003). *LSCs: A Graphical Formalism for the Specification of Communication Behavior*. PhD thesis, Carl von Ossietzky Universität Oldenburg.

Ludewig, J. and Lichter, H. (2013). *Software Engineering*. dpunkt.verlag, 3. edition.

OMG (2007). Unified modeling language: Superstructure, version 2.1.2. Technical Report formal/07-11-02.

Rupp, C. and die SOPHISTen (2014). *Requirements-Engineering und -Management*. Hanser, 6th edition.

Störrle, H. (2003). Assert, negate and refinement in UML-2 interactions. Technical Report TUM-I0323, Technische Universität München.

V-Modell XT (2006). *V-Modell XT*. Version 1.4.