

Software Design, Modelling and Analysis in UML  
 Lecture 17: Live Sequence Charts I

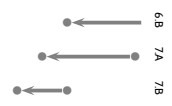
2017-01-17

Prof. Dr. Andreas Podelski, Dr. Bernd Westphal  
 Albert-Ludwigs-Universität Freiburg, Germany

The Plan

- Thu. 19. 1: Live Sequence Charts I  
 Priority: State-Machines/Rest, Code Generation
- Tue. 24. 1: Live Sequence Charts II
- Thu. 26. 1: Live Sequence Charts III
- Tue. 31. 1: Tutorial 7
- Thu. 2. 2: Model Based/ Driven SW Engineering
- Mon. 6. 2: Inheritance
- Tue. 7. 2: Meta-Modelling - Questions

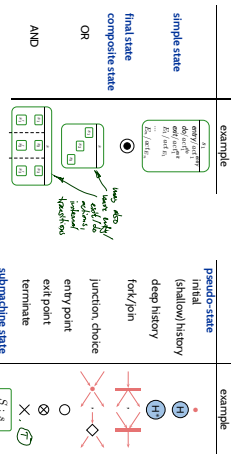
February 17th: The Exam.



Content

- Hierarchical State Machines
- Action vs. Passive Objects
- Methods / Behavioural Features
- Code Generation
- Discussion
- Performance Descriptors of Behaviour
- Interactions
- A Brief History of Sequence Diagrams
- Live Sequence Charts
- Abstract Syntax
- Well-Formedness

Hierarchical State Machines: Retrospective



UML distinguishes the following kinds of states:

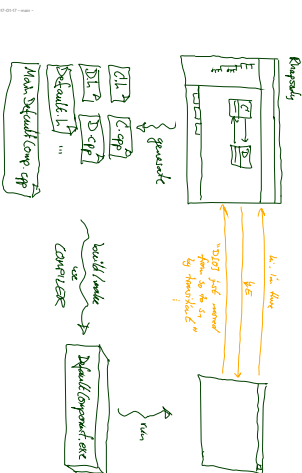
Exercise 4. (1)

nr.	action	edge	st.	stateId	inv.	st.	stateId	z	value
1	0	-1	0	0	1	0	0	1	0
2	4	-4	5	0	4	0	4	4	0
3	4	-4	5	1	4	0	4	4	0
4	1	-1	5	4	0	1	0	1	0
5	1	-1	5	4	0	1	0	1	0
6	1	-1	5	4	0	1	0	1	0
7	1	-1	5	4	0	1	0	1	0
8	1	-1	5	4	0	1	0	1	0
9	1	-1	5	4	0	1	0	1	0
10	1	-1	5	4	0	1	0	1	0
11	1	-1	5	4	0	1	0	1	0
12	1	-1	5	4	0	1	0	1	0
13	1	-1	5	4	0	1	0	1	0
14	1	-1	5	4	0	1	0	1	0
15	1	-1	5	4	0	1	0	1	0
16	1	-1	5	4	0	1	0	1	0
17	1	-1	5	4	0	1	0	1	0
18	1	-1	5	4	0	1	0	1	0
19	1	-1	5	4	0	1	0	1	0
20	1	-1	5	4	0	1	0	1	0
21	1	-1	5	4	0	1	0	1	0
22	1	-1	5	4	0	1	0	1	0
23	1	-1	5	4	0	1	0	1	0
24	1	-1	5	4	0	1	0	1	0
25	1	-1	5	4	0	1	0	1	0
26	1	-1	5	4	0	1	0	1	0
27	1	-1	5	4	0	1	0	1	0
28	1	-1	5	4	0	1	0	1	0
29	1	-1	5	4	0	1	0	1	0
30	1	-1	5	4	0	1	0	1	0
31	1	-1	5	4	0	1	0	1	0
32	1	-1	5	4	0	1	0	1	0
33	1	-1	5	4	0	1	0	1	0
34	1	-1	5	4	0	1	0	1	0
35	1	-1	5	4	0	1	0	1	0
36	1	-1	5	4	0	1	0	1	0
37	1	-1	5	4	0	1	0	1	0
38	1	-1	5	4	0	1	0	1	0
39	1	-1	5	4	0	1	0	1	0
40	1	-1	5	4	0	1	0	1	0
41	1	-1	5	4	0	1	0	1	0
42	1	-1	5	4	0	1	0	1	0
43	1	-1	5	4	0	1	0	1	0
44	1	-1	5	4	0	1	0	1	0
45	1	-1	5	4	0	1	0	1	0
46	1	-1	5	4	0	1	0	1	0
47	1	-1	5	4	0	1	0	1	0
48	1	-1	5	4	0	1	0	1	0
49	1	-1	5	4	0	1	0	1	0
50	1	-1	5	4	0	1	0	1	0





17/4



18/4

Tell Them What You've Told Them...

- Rhapsody also supports non-active objects – their instances share an event pool with an active object
  - Behavioural Features: exist
  - Semantic Variation Points are regions – but manageable, e.g. by appropriate modeling guidelines (stick to 'the beaten track')
  - Interactions can be used for **reflective** descriptions of behaviour, i.e.
    - describe what behaviour is (un)desired
    - without (yet) defining how to realize it
  - One visual formalism for interactions: **Live Sequence Charts**
  - partially ordered locations
  - instantaneous and asynchronous messages
  - conditions and local invariants
- Later pre-Charts

33/4

References

33/4

References

Cane, M. L. and Dringel, J. (2007). UML vs. classical vs. rhapsody stavecharts: not all models are created equal. *Software and System Modeling*, 6(4):415–435.

Damm, W. and Hund, D. (2001). LSCs: Breathing life into Message Sequence Charts. *Formal Methods in System Design*, 9(1):65–80.

Harel, D. (1997). *Semantic Foundations of Statecharts*. D. years later. In Gumberg, O., editor, *CMU*, volume 1354 of *LICS*, pages 226–231. Springer-Verlag.

Harel, D. and Geay, E. (1997). Executable object modeling with statecharts. *IEEE Computer*, 30(7):31–42.

Harel, D. and Peles, S. (2007). *Asert and negate revisited: Modal semantics for UML sequence diagrams. Software and System Modeling (SSSM)*. To appear. (Early version in SCEASO6, 2006, pp. 13–30).

Harel, D. and Mureşanu, R. (2003). *Came Let's Play... Scenario-Based Programming Using LSC and the Play-Engine*. Springer-Verlag.

Klopp, J. (2003). *LSC: A Graphical Formalism for the Specification of Communication Behaviour*. PhD thesis, Carl von Ossietzky Universität Oldenburg.

OMG (2007). *Unified modeling language: Superstructure, version 2.1.2*. Technical Report formal/07-11-Q2.

OMG (2010a). *Unified modeling language: Metastructure, version 2.4.1*. Technical Report formal/2010-08-05.

OMG (2010b). *Unified modeling language: Superstructure, version 2.4.1*. Technical Report formal/2010-08-06.

Stefan, H. (2003). *Asert graphs and references to UML 2 interactions*. In Jilpelt, J., George, B., France, R., and Fernandez, E. B., editors, *CSO/WC 2003*, number TUM-1033. Technische Universität München.

34/4