Prof. Dr. A. Podelski, Dr. B. Westphal
M. Vujinovic

Wintersemester 2015/16

Software Design, Modeling, and Analysis in UML

http://swt.informatik.uni-freiburg.de/teaching/WS2015-16/sdmauml

Exercise Sheet 5

Early submission: Wednesday, 2016-01-06, 12:00 Regular submission: Thursday, 2016-01-07, 10:00

Exercise 1 — Ether (5/20 Points)

Consider the FIFO queue ether

$$\varepsilon = \underline{(u_2, e_1)}.(u_1, e_2)$$

with respect to the system state defined by the complete object diagram in Figure 3. (First entry of the FIFO underlined for clarity.)

(i) What is $ready(\varepsilon, u_2)$? (1)

(ii) What is $ready(\varepsilon, u_1)$? (1)

(iii) What is $\varepsilon' := \ominus(\varepsilon, e_1)$? (1)

(iv) What is $\varepsilon'' := \ominus(\varepsilon', e_2)$? (1)

(v) What is $\varepsilon''' := \oplus(\varepsilon'', u_1, e_3)$, given e_3 is the identity of an instance of signal WQ? (1)

Exercise 2 (2/20 Points)

The class diagram in Figure 1 induces a signature $\mathscr{S}_0$ with *explicitly* given basic types and attributes, while the complete object diagram in Figure 3 shows a system state with respect to a signature $\mathscr{S}$ which in particular comprises *implicit* attributes.

Provide the signature $\mathscr{S}$ according to the definition from the lecture.

Exercise 3 (3/20 Points)

Consider the system configuration (σ, ε) where system state σ is defined by the complete object diagram in Figure 3 and ε is the ether from Exercise 1.

> The intuition of the signals and attributes is as follows: If a customer presses the water button on the choice panel without having inserted money beforehand, it works as a query for whether and how much water is in stock. The button on the vending machine then sends an instance of signal QW ("water query") to the drink dispenser controller, here modelled by making QW an environment signal.
>
> *wbtn* is supposed to model a light behind the button for water, it is switched on (set to 1) while the query is processed. The choice panel needs to ask the drink dispenser because attribute *wis* is private. This is done by signal QW. The drink dispenser is supposed to reply using signals $EMPTY$ or W, where W has an attribute which carries the amount of water in stock.
>
> If the drink dispenser responds with $EMPTY$, then the water button light is just switched off, otherwise the amount of water in stock is shown on a display of the vending machine, here modelled by an attribute *dsp*. We assume that assigning a number to

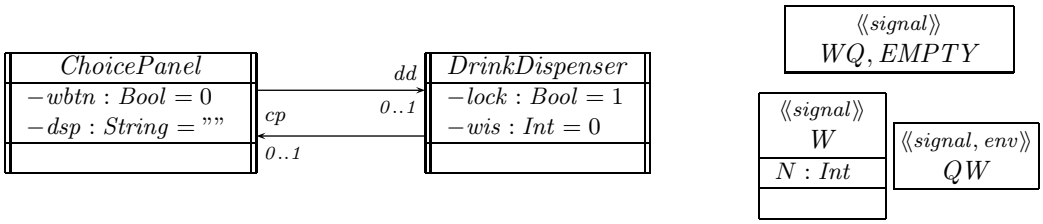
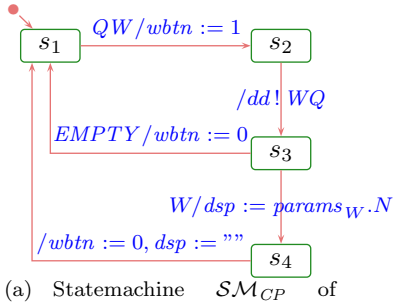
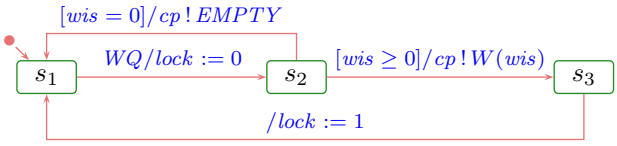


Figure 1: Vending machine class diagram.



(a) Statemachine $\mathcal{SM}_{CP}$ of choice panels.



(b) Statemachine $\mathcal{SM}_{DD}$ of drink dispensers.

Figure 2: State machines.

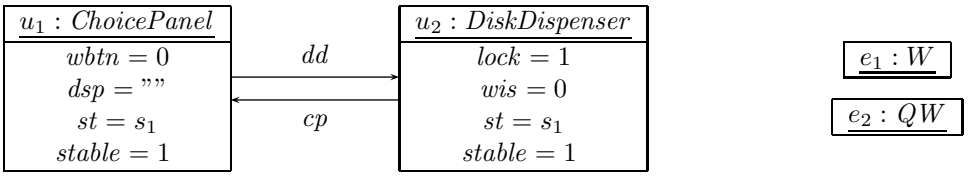


Figure 3: Complete object diagram.

the string-type attribute *dsp* implicitly converts the number to a corresponding string representation.

To make the behaviour a bit more interesting, we assume that the drink dispenser needs to unlock the water storage before being able to count the water units. This is modelled by the boolean attribute *lock*. In case the drink dispenser replies with *EMPTY*, it doesn't bother to re-lock the water storage, we assume that it will be locked again when filling up water (not in the model).

(i) What are the possible effects of u_1 executing action `skip`? (1)

(ii) What are the possible effects of u_1 executing action $wbtn := 1$? (1)

(iii) What are the possible effects of u_1 executing action $dd\,!\,WQ$? (1)

Exercise 4 (10/20 Points + 5 Bonus)

Consider the system configuration (σ, ε) where system state σ is defined by the complete object diagram in Figure 3 and ε is the ether from Exercise 1.

(i) How can the system evolve from (σ, ε) when not using the rule for environment interaction? (For each step, note down the rule by which it is justified.) (5)

Hint: choose a convenient and readable representation of the possible steps, e.g., a table similar to the one in the lecture. Note that information which does not change at all.

(ii) Explain the difference between step and RTC-step using your results for the previous task. (1)

Hint: how are step and RTC-step related? Can you point out a step which is also an RTC-step? An RTC-step which is (or is not) a step?

(iii) Is it possible to reach the designated error configuration from (σ, ε)? If yes, point out how – if not, propose a modification of the state machines or of the considered system configuration such that the error configuration is reached. (1)

(iv) If we also consider the rule for environment interaction, how does the possible behaviour change? (1)

(v) Consider the OCL constraint

$$\mathsf{context}\ CP\ \mathsf{inv} : st = s_4\ \mathsf{implies}\ dd.wis > 0$$

Is it an invariant of the behaviour which is possible from system configuration (σ, ε)?

If yes, argue why, if no, point out a counter-example. (1)

(vi) The state names s_1, s_2, etc. are not very intuitive, so the diagram is not very friendly to the readers in this aspect. Can you suggest more intuitive names? (1)

(vii) Reconsider Task (v). If you think that the OCL constraint is an invariant of the considered behaviour: how would you prove that? If not: can you propose a change to the state machines (which is still consistent with the purpose of the model given above) such that the constraint becomes an invariant? (And how would you prove that your change is correct, i.e., ensures that the constraint is an invariant.) (5 Bonus)

Exercise 5 (10 Bonus)

(i) Enter the UML model considered in this exercise sheet into Rhapsody, generate code, and demonstrate that your Rhapsody model is "not completely broken" by simulating at least one interesting evolution. Record your simulation as a sequence diagram, and submit your model, including the sequence diagram, and an explanation why it is interesting. (5)

(ii) How does the behaviour that you can simulate with Rhapsody relate to your results for the first task of Exercise 4? Is all the behaviour you claimed there available in Rhapsody? Is it more, is it less? Why? (5)