Prof. Dr. A. Podelski, Dr. B. Westphal
C. Schätzle

Wintersemester 2016/17

Software Design, Modeling, and Analysis in UML

http://swt.informatik.uni-freiburg.de/teaching/WS2016-17/sdmauml

Exercise Sheet 4

Early submission: Monday, 2016-12-05, 12:00 Regular submission: Tuesday, 2016-12-06, 8:00

Exercise 1 (7/20 Points)

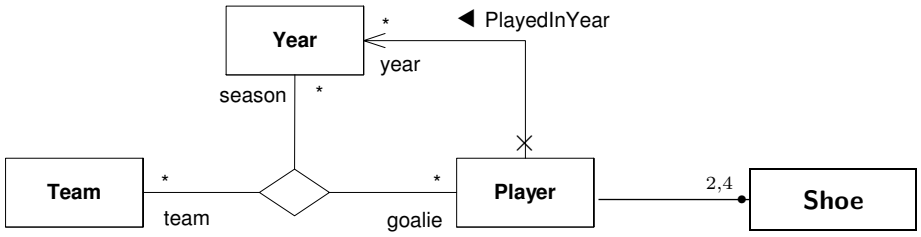


Figure 1: League model (OMG, 2011, 44).

(i) Provide the signature defined by the class diagram in Figure 1. Assume that class **Team** has the boolean basic type attribute '*hasLicence*' (not shown in the diagram). (2)

(ii) Give an interesting (!) system state (σ, λ) which illustrates the new component λ with a non-empty link for the ternary relation.

In addition, at least one Player of your λ should have played in at least one year. (3)

(iii) To which value does the OCL constraint

$$\mathsf{context}\ \mathit{Player}\ \mathsf{inv}: \mathit{team}\ \mathsf{->}\ \mathsf{forAll}(t \mid t.\mathit{hasLicence})$$

evaluate in your system state from (ii)?

(Argue based on the value of $\mathit{self}.\mathit{team}$, that is, compute $I[\![\mathit{self}.\mathit{team}]\!]((\sigma, \lambda), \beta)$ for at least one interesting binding of $\mathit{self}_{\mathit{Player}}$ by β.) (2)

Exercise 2 (6/20 Points)

Assume that the class diagram in Figure 1 is supposed to capture information about teams in a league. Assume that the diagram models a particular league where there is the (admittedly strange) rule saying that a player is not allowed to be the goalkeeper ("goalie") in more than 2 years. In order to reflect these rules in the diagram, a modeler proposes to change the multiplicity of the association end *season* accordingly.

(i) Provide the OCL constraints induced by the multiplicities of the ternary association *after the change*. (3)

(ii) Another modeler claims that the change does not have the desired effect, that is, that it is not the case that exactly those system states of the changed class diagram which are consistent with the constraints in the diagram do respect the league's rule.

Provide a convincing counter example. (2)

(iii) Do you have an own proposal to formalise the league's rule?

If you have a proposal, test it on the example from Task (ii). If not, argue why this requirement cannot be formalised using UML. (1)

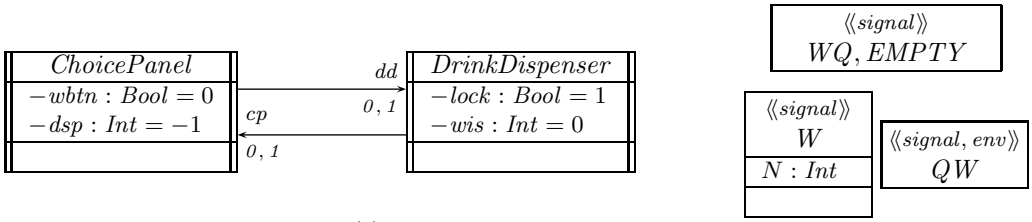
Exercise 3 (2/20 Points)

(a) Bidirectional. (b) Unidirectional.

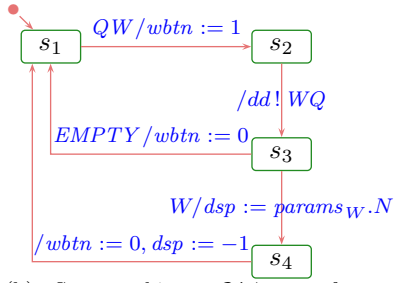
Figure 2: Bidirectional vs. unidirectional links.

In some older UML textbooks one finds the claim that the two class diagrams shown in Figure 2(a) and Figure 2(b) are equivalent in the sense that for each system state of one, there is a system state of the other which provides the same information, and vice versa.
One way to prove that diagrams are *not* equivalent in this sense is to give an OCL constraint which is satisfied by a system states of one but where there is no system state of the other which also satisfies the constraint.
Provide (and, of course, explain) such an OCL constraint.

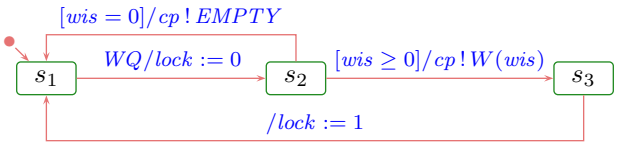
Exercise 4 (5/20 Points)



(a) Vending machine class diagram.



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



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

Figure 3: State machines.

(i) Give the core state machines corresponding to the diagrams shown in Figures 3(b) and 3(c). (3)

(ii) Provide a Rhapsody model of the core state machines in Figures 3(b) and 3(c). A Rhapsody model of the structure will be provided on the course's homepage as a starting point. (2)

Note: Rhapsody uses the following particular C++ types and expressions where Figure 3 uses the types and abstract action language from the course.

Abstract Action Language	Rhapsody in C++
$wbtn : Bool$	`wbtn : bool`
$dsp : Int$	`dsp : int`
$wbtn := 1$	`wbtn = 1`
$wis = 0$	`wis == 0`
$dd\,!\,WQ$	`dd->GEN(WQ)`
$cp\,!\,W(wis)$	`dd->GEN(W(wis))`
$dsp := params.N$	`dsp = params->N`

Exercise 5 (5 Bonus)

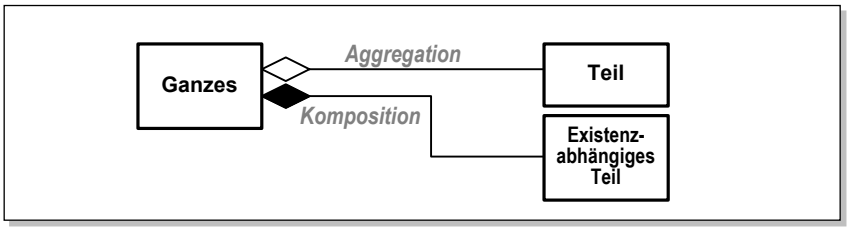


Figure 4: Aggregation and composition. (Oestereich (2006))

In addition to the decorations of association ends discussed in the lecture, UML admits at most one end to be decorated with a hollow or solid "diamond", to indicate "aggregation" and "composition" (cf. Figure 4).

Integrate the semantics of aggregation and composition into the course's formal framework.

Hint: That is, first assess what has to be covered (name it, cite it from the standard) and briefly explain its informal semantics as given by the standard.
Recall that we've seen that different "UML things" are of different semantical relevance: it ranges from, e.g., attribute types with prominent semantical relevance, over activeness which we postponed, to reading direction of association names which we don't even represent in the abstract syntax because of its weak semantical relevance.
Discuss into which class of relevance you think aggregation/composition belongs and treat it accordingly. For example, if you opt for prominent semantical relevance, you may have to extend the definition of signatures, to define the mapping from diagram to your extended signature, think about the impact on OCL and well-typedness etc.

References

Oestereich, B. (2006). *Analyse und Design mit UML 2.1, 8. Auflage.* Oldenbourg, 8. edition.

OMG (2011). Unified modeling language: Superstructure, version 2.4.1. Technical Report formal/2011-08-06.