Model-based Software Engineering

Summer Term 2016

Exercise Sheet 01: **Metamodeling and UML**

Note: *The questions on this sheet will be discussed in the tutorial on April 26. We will not grade your exercises, but we still encourage you to try solving them by yourself, as similar or related questions may be asked in the exam. Discussions on the exercise tasks with other students via the Stud.IP forum are highly encouraged as well: Share your solutions or questions and comment on the solutions and questions posted by others!*

Questions marked as "presentation bonus" can be used to get your presentation credit, which is necessary in order to obtain the full bonus in the mini-projects. To do so, you are required to prepare a few presentation slides that illustrate your solution and present the solution in the tutorial session. Also, after giving your presentation, please send an email with your name and your presentation slides in PDF format to `greenyer@inf.uni-hannover.de`.

1 UML metamodel and abstract syntax (presentation bonus)

Figure 1 shows a UML class diagram (it models part of the Petri net meta model from the lecture).

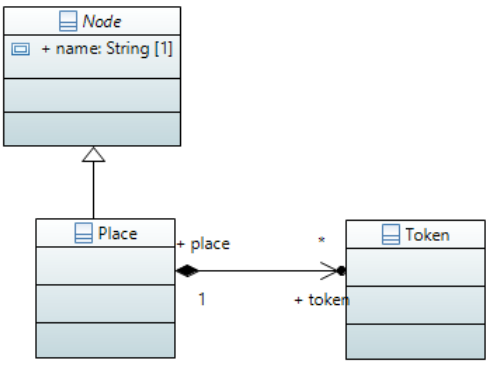

Figure 1: Class diagram

Perform the following **steps** and answer the following **questions**:

Steps:

1. Download and install Eclipse Modeling Tools (`http://www.eclipse.org/downloads/`).

2. Install the Papyrus UML editor (via the menu Help->Install Modeling Components...)
3. Model the class diagram shown in Fig. 1 or import the archived project that accompanies this exercise sheet.
4. Open the corresponding .uml file in the UML Model Editor, which displays the model in a tree. You may have to right-click on the UML file and select "Open with", "UML Model Editor".
5. Explore the model's tree structure in the editor and also the attributes and reference links among the model elements, which you can see in the Properties View.
6. Also take a look at the .uml file using a text editor ("Open with", "Text Editor") and see how the structure of the nested XML elements corresponds to the structure shown in the tree editor and diagram.
7. Download the UML specification from `http://www.omg.org/spec/UML/2.5/PDF` and have a look into it.
8. Open the UML metamodel in Eclipse: Go to "File", "Import", "Plug-In Development", "Plug-Ins and Fragments", "Next", "Next", select "org.eclipse.uml2.uml" and "Finish". This plug-in should now appear in your explorer. In the folder "model", you find the file UML.ecore. Have a look at this model and see how it corresponds to the UML specification document, i.e., try to find some metaclasses that you find in the UML specification and vice versa, and compare attributes and references.

Questions (to be presented):

a) Draw an object diagram for the abstract syntax of the above UML class diagram according to the structure that you find in the .uml file that you inspected. (Use the notation used for object diagrams in the lecture, see for example slide 13 of lecture 02.) The object diagram must show the following information:

i) all objects that comprise the class model, labeled with :metaclass.

ii) all reference links labeled with the name of their metareference (metaassociation). If a link is a containment, also add a filled diamond to the link arrow in order to illustrate that one object contains the other.

iii) all attribute values for attributes that are set to a value other than their default value. (If an attribute is set to a value other than the default, it shows up in the XML representation. For example, you should see that for the Node class, there is an attribute

`isAbstract="true"`, but there is no `isAbstract="false"`) for the other classes, since `false` is the default value.)

Use the UML specification and the other artifacts that you discovered following the steps above where you think they help you understand the UML metamodel and the abstract syntax of the class diagram.

b) For three of the UML metaclasses that you see instantiated in this class diagram, find the corresponding definition in the UML specification. Present and cite each definition (or at least the first one or two paragraphs, as you see fit, if the definition is really long).

c) Find and present class diagrams of the UML metamodel in the UML specification that show these metaclasses and their relationships.

2 UML metamodel and abstract syntax II (presentation bonus)

Answer the same questions as in Exercise 1 with the UML diagram shown in Fig. 2 below. Note here that the action "Prepare Sauce" in the activity "Prepare Spaghetti with Sauce" is a Call Behavior Action that references the activity "Prepare Sauce" shown below.

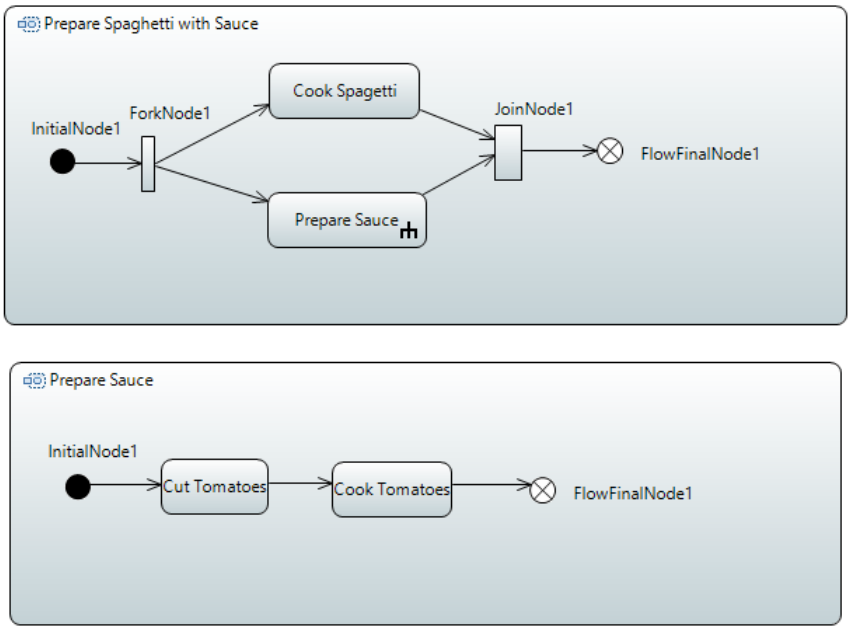

Figure 2: Activity diagram

3 UML metamodel: Classes, InstanceSpecifications, Roles

Answer the following questions based on the UML specification `http://www.omg.org/spec/UML/2.5/PDF`.

a) *Classes*: Read the definition of a *Class* in Sect. 11.4. of the UML specification.

- **i)** What are variants in the notation of classes?
- **ii)** From which metaclasses does the metaclass Class inherit?
- **iii)** How is the metaclass Class related to the metaclass StructuredClassifier?

b) *Roles*: Read the definition of *Structured Classifiers* in Sect. 11.2.

- **i)** What is a Structured Classifier?
- **ii)** What is a Role?
- **iii)** Look at Figure 11.5 (i) and (ii): what are the boxes and lines show in these two subfigures? How are they related? Try to draw an abstract syntax diagram to understand the relationship better.

c) *InstanceSpecification*: Read the definition of *Instances* in Sect. 9.8.

- **i)** What is a Instance Specification?
- **ii)** What is a Slot and an Instance Value?
- **iii)** What are the rules for the notation of Instance Specifications and Instance Values? What is the notation for an Instance Specification whose classifier is an Association?

d) What is the difference between an Instance Specification and a Role?

- **i)** with respect to their semantics?
- **ii)** with respect to their notation?