TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science Institute of Software and Multimedia Technology, Software Technology Group

WS2020/21 – Model-driven Software Development in Technical Spaces

Graphical Modeling Languages

Professor: Prof. Dr. Uwe Aßmann
Tutor: Markus Hamann

1 Domain Specific Modeling Languages with Sirius

This exercise provides a brief introduction to the design of *graphical Domain Specific Languages* (DSL) for a given domain model. In this exercise we focus on **Sirius**[1] [1], a model-driven development framework for graphical modeling editors based on Eclipse and EMF.

1.1 Sirus

Sirus is a development framework to create model editors from a given *metamodel* and a *view specification*. The view specification has a similar purpose as the grammars used in the last exercise and describes e. g. how model elements should be rendered in the editor. We will use Sirius to generate a graphical DSL for our statecharts.

To start the exercise you must install Sirus and familiarize yourself with the Sirus Editor Development Toolkit.

- Install *Sirius 6.3* from the *Eclipse Marketplace*.
- Familiarize yourself with Sirius and its capabilities to create graphical model editors based on a given metamodel. Good starting points are the following Tutorials[2][3].
- Open a new eclipse workspace and import your statechart metamodel projects (Metamodel, Edit, and Editor) in that workspace. Do not use the same workspace as the previous exercise since *XText* and Sirus need some further configuration to work together.
- *Run* the Metamodel Project *as* an *Eclipse Application* to register your metamodel in Eclipse. In the new Eclipse instance create a new *Viewpoint Specification Project* and open the *View Specification* (`*.odesign`) file. Edit this file like in the tutorials to develop the graphical statechart editor.
- To test the developed editor create a *Modeling Project* in the same Eclipse instance and add some `*.statechart` files to it (Or use the ones from Exercise 1).

[1] https://www.eclipse.org/sirius
[2] https://wiki.eclipse.org/Sirius/Tutorials/StarterTutorial
[3] https://wiki.eclipse.org/Sirius/Tutorials/AdvancedTutorial

1.2 Graphical Statechart Editor

Define a graphical modeling editor for your statechart models by using the statechart metamodel from *Exercise 1*. The editor should be able to render, create, and edit all elements of the statechart model. Additionally, Try to add the following semantics:

- A *Transition* can only start from a *Normal* or *Initial State*.
- A *Transition* can only end at a *Normal* or *Final State*.
- The *name* of the *Initial State* must always be "initial" and can not be changed.

The visual representation must be created as a `*.odesign` file and instances as `*.statechart` files with embedded graphical representations, respectively. These files must be handed in on the day before the next exercise. If you wish a review of your solution, please hand in the `*.odesign` and `*.statechart` files as a `*.zip` file till the day before the next exercise (E-Mail: `Markus.Hamann1@tu-dresden.de`). If you did not hand in your metamodel after exercise 1, you need to hand it in too. Also, on the day of the next exercise, an example solution will be published.

References

[1] Vladimir Viyović, Mirjam Maksimović, and Branko Perisić. Sirius: A rapid development of dsm graphical editor. In *Intelligent Engineering Systems (INES), 2014 18th International Conference on*, pages 233–238. IEEE, 2014.