



WS2017/18 – Model-driven Software Development in Technical Spaces

Graphical Modeling Languages

Professor: Prof. Dr. Uwe Aßmann
Tutor: Dr.-Ing. Thomas Kühn

1 Domain Specific Modeling Languages with Sirius

This exercise provides a brief introduction to the design of graphical representations for a given domain model. In particular, this exercise focuses on **Sirius**¹ [1], a model-driven development framework for graphical modeling editors based on Eclipse and EMF. In general, the task will be to create graphical representations of domain models by means of *Viewpoint Specification Models*.

1.1 Task 1: State Machines

Create a graphical modeling editor for UML state machines (cf. EMF exercise).

- Install *Sirius* and understand the capabilities of *Viewpoint Specification Models*, especially for generating diagrams.
- Specify a *Viewpoint Specification Model* (*.odesign) for the state machine meta-model featuring *transitions*, *guards*, *regular*, *initial*, and *final states*.
- Test the resulting graphical editor by creating several *state machines*.

1.2 Task 2 : Class Diagrams

Define a graphical modeling editor for UML class diagrams (cf. DSL exercise).

- Declare a *Viewpoint Specification Model* (*.odesign) for the class diagram meta-model featuring *classes*, *attributes*, *methods*, *inheritance*, *associations*, *aggregations* and *composition*.
- Test the resulting graphical editor by creating several *class diagrams*.

¹<https://www.eclipse.org/sirius>

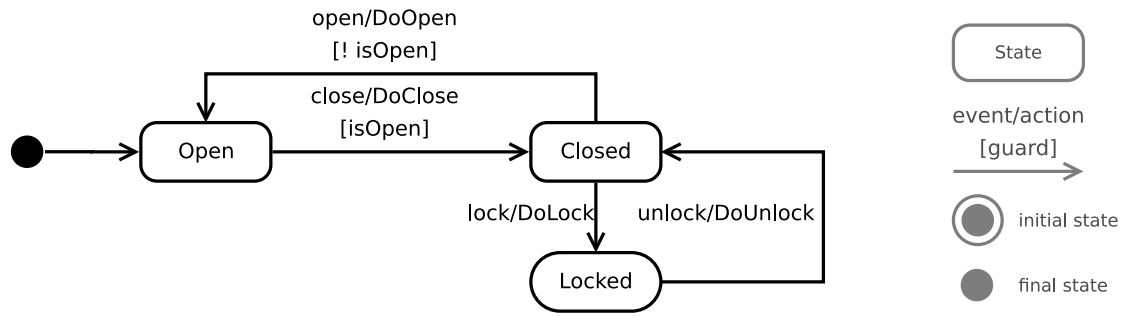


Figure 1: Possible visual representation of UML state machines.

1.3 Task 3 : Integrated Domain Modeling

Optionally, extend the graphical modeling editor for UML class diagrams to support embedding of state machines into classes.

- Declare a *Viewpoint Specification Model* (*.odesign) for the integrated class diagram (cf. Model-to-Text exercise) that allows for specifying state machines for individual classes.
- Reuse the *Viewpoint Specification* for state machines to permit the creation of separate state machine models per class. Support the creation of individual state machine models for each class in the *integrated class diagram*.

All three visual representations must be created as *.odesign files and instances as *.statechart, *.classdiagram, *.classstatediagram files with embedded graphical representations, respectively. These files must be handed in on the day before the next exercise.

1.4 Additional Information

- Sirius², is a model-driven framework for the development of graphical editors.
- Sirius Tutorial³, is an advanced tutorial on Sirius and the creation of *Viewpoint Specification Models*.

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.

²<https://www.eclipse.org/sirius>

³<https://wiki.eclipse.org/Sirius/Tutorials/AdvancedTutorial>