



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

Domain Specific Languages with Xtext

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

1 Domain Specific Languages

The purpose of this exercise is to introduce **Domain Specific Languages (DSL)** and creating a text-based Editor for our Statecharts using the *Xtext*¹ tool suite for Eclipse.

1.1 Xtext

To start with the exercise, please download Xtext via the Eclipse Marketplace and create an Xtext project based on your Statechart metamodel (Exercise 1).

Install Xtext:

- Start *Eclipse*.
- Open the *Eclipse Marketplace* with *Help->Eclipse Marketplace..*
- Search for Xtext, install *Eclipse Xtext 2.XX.X* and restart Eclipse

Create the project:

- Open the workspace with your solution of Exercise 1.
- Open the Creation Wizard with *File->New->Other* and choose *Xtext Project from Existing Ecore Models*.
- On the next page *Add* the `genmodel` file of your metamodel.
- Last, create your project by providing a name.
- Find the `*.xtext` grammar file under the `src` folder of the newly created main project.

This grammar file will be the core of your new DSL. In it, you will describe the rules that define the parsing of structured text to our statechart model elements. To get an overview of the syntax of Xtext rules, please read through this documentation². After this step, you can build and start the new DSL text editor.

¹<https://www.eclipse.org/Xtext/>

²https://www.eclipse.org/Xtext/documentation/301_grammarlanguage.html

Start the editor after working on the grammar file:

- Rightclick on the `*.mwe2` file next to the grammar file and choose *Run As...->MWE2 Workflow* to create the DSL editor.
- After the editor is created, rightclick on the model project and choose *Run As* and *Eclipse Application*. A new Eclipse application will open. There, create a new general *Project* and a `*.statechart` file. You should now be able to model your statechart with the DSL editor. If the editor did not open, you can open it with a rightclick on the `*.statechart` file and *Open With->-your- Editor*.

1.2 Statechart DSL Editor

Create a DSL Text Editor based on the statechart metamodel from Exercise 1. After this, test the editor by recreating your example statechart models in a human-readable form. If you could not finish Exercise 1, an example solution will be provided.

If you wish a review of your solution, please hand in the project as a `*.zip` file till the day before the next exercise (E-Mail: Markus.Hamann1@tu-dresden.de). Also, on the day of the next exercise, an example solution will be published.

1.3 (Optional) A DSL Based on Your Project

Like in the last exercise, we want to encourage you to create your own model-driven toolchain. If you created a metamodel from one of your projects, try to create a DSL text editor for it.