



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

Domain Specific Languages with EMFText

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

1 EMFText

The purpose of this exercise is to understand how to build domain specific languages (DSL). To get started, first download Eclipse Modeling Tools and afterwards install EMFText¹ [1].

1.1 Task 1: Statechart DSL

- Use the metamodel for state charts from the first exercise
- Develop a DSL for statecharts using the keywords: `statechart`, `state` and `transition`.

1.2 Task 2: Class Diagram DSL

- Develop a metamodel for class diagrams, which supports
 - Classes,
 - Attributes,
 - Methods with parameters and return types, as well as
 - Inheritance, Associations, Aggregations, and Compositions.
- Develop a DSL for this metamodel using the keywords `attribute`, `method`, `class`, etc.
- Evolve your DSL so the keywords `attribute` and `method` are no longer needed.

1.3 Task 3: DSL Integration

- Integrate the two DSLs developed in Task 1 and 2.
- Each class can additionally have a statechart as shown in Listing 1.

All three DSL projects must be handed in as `*.zip` archives on the day before the next exercise.

¹<http://www.emftext.org>

Listing 1: Example DSL instance for Class Model+Statechart.

```
1 class Door {
2     attribute isOpen:boolean ;
3     method isOpen():boolean;
4     method open ();
5     method close ();
6     statechart {
7         state open ;
8         state closed ;
9         transition open ( close () [ isOpen ] / DoClose ) closed
10        transition closed ( open () [ isNotOpen ] / DoOpen ) open
11    }
12 }
```

References

- [1] Florian Heidenreich, Jendrik Johannes, Sven Karol, Mirko Seifert, and Christian Wende. Model-based language engineering with emftext. In *Generative and Transformational Techniques in Software Engineering IV*, pages 322–345. Springer, 2013.