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

Model-to-Text Transformations

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

1 Acceleo

The purpose of this exercise is to understand how to realize a model to text transformation. In fact, this exercise focuses on template-based code generation utilizing *Acceleo*[1] [1]. In general, the task is to create three model-to-text transformators, to generate valid *Java* source code from state machines and class diagrams, defined in the previous exercise.

1.1 Task 1: Basic Code Generation

- Install and understand the *Acceleo* toolkit.[2]
- Write *Acceleo* templates to generate valid *Java* code from *class diagrams*, defined in the previous exercise.

1.2 Task 2: Code Generation for State Machines

- Write *Acceleo* templates to generate valid *Java* code from *state machine* models, defined in the previous exercise.

  Hint: *Use the state pattern to translate state machines into code.*

1.3 Task 3: Complex Code Generation

- Combine both generators to generate valid Java code from the integrated DSL, i.e., combining class diagrams and state machines.

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

[1] https://www.eclipse.org/acceleo/

[2] https://wiki.eclipse.org/Acceleo/Getting_Started

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

```
class Door {
    boolean isOpen;
    void doClose();
    void doOpen();

    void open();
    void close();
    void look();
    void unlock();

    statechart Lifecycle {
        state open;
        state closed;
        state locked;
        transition open   ( close [ isOpen ] / doClose ) closed;
        transition closed ( open [ !isOpen ] / doOpen ) open;
        transition closed ( lock ) locked;
        transition locked ( unlock ) closed;
        initial open;
    }
}
```

1.4 Additional Information

- *Acceleo*,[3] is a pragmatic implementation of the Object Management Group (OMG) MOF Model-to-Text Language (MTL) standard.
- *Acceleo Getting Started*,[4] is a basic tutorial on the use of Acceleo.

References

[1] Jonathan Musset, Étienne Juliot, Stéphane Lacrampe, William Piers, Cédric Brun, Laurent Goubet, Yvan Lussaud, and Freddy Allilaire. Acceleo user guide. *Acceleo*, 2, 2006.

[3] https://www.eclipse.org/acceleo/

[4] https://wiki.eclipse.org/Acceleo/Getting_Started

Listing 2: Example Java Code Generated from Listing 1.

```java
abstract class Door {
    private boolean isOpen;
    public abstract void doOpen()
    public abstract void doClose()
    public abstract Door open();
    public abstract Door close();
    public abstract Door lock();
    public Door unlock();

    public Door (boolean open){ isOpen=open; }
    public void setOpen (boolean open) { isOpen = open; }
    public boolean isOpen() { return isOpen; }

}

class OpenDoor extends Door {
    // ...
}
// ...
```