Concept

This is a rough concept of the aspects that should/can be covered in this project.

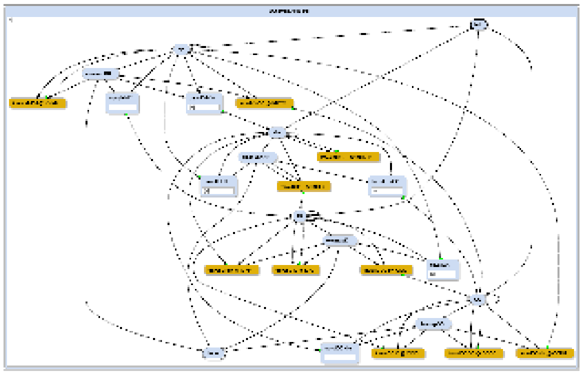
SCCharts Railway Controller

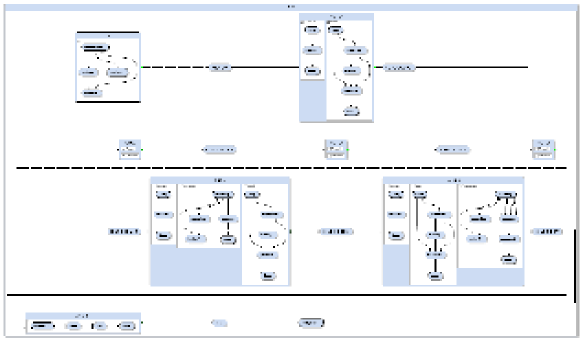
- **SCCharts Railway Controller**
  A controller modeled in SCCharts that runs the railway.
  The previous controller was able to control up to 11 trains with a dynamic schedule. However, it depends heavily on hostcode and is therefore not suited for different APIs or simulation.
  For this project, all previous Documentation and SCCharts are available for inspection.

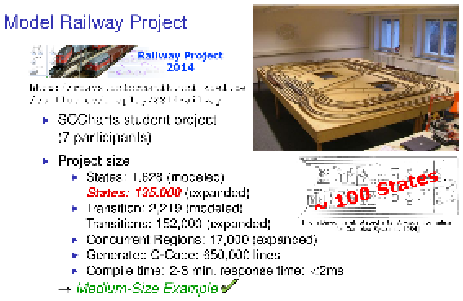
- **PROM Environment**
  The new SCCharts Controller should use the KIELER Template system for code generation to connect the model to the API.
  You have to write the API templates yourself. However, after creating them, it should be easy to reuse the templates and adjust them for simulation.
  You can read more about the template engine at our PROM environment page.

- **SCCharts Railway Simulation**
  Create an fully integrated simulation. Therefore, you have to develop an SCChart that represents the behaviour of the real model railway. It should produce the outputs that enable to run an SCCharts railway controller without the actual railway API in C.
  There exists a simulation model written in SCADE. You can similar features such as Dataflow Region in SCCharts. The Simlation should be integrated in the KIELER framework.

- **Railway Simulation and Visualization in KIELER**
  Create a vizualization of the simulation when connected to a controller. KIELER already provides ways to run and visualize models.
  You can find more information about the KIELER Environment Visualization at our KEV page.



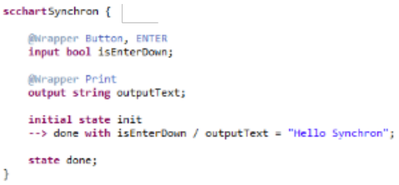




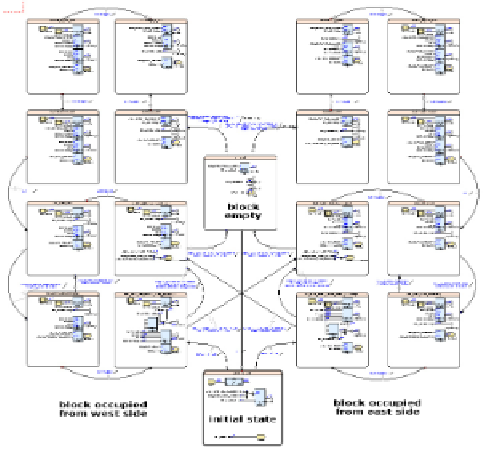
Model Railway Project

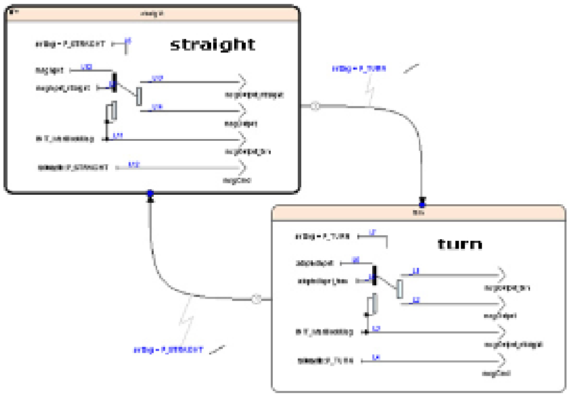
- SCCharts student project (7 participants)
- Project size
  - States: 1,678 (modeled)
    States: 135,000 (expanded)
  - Transition: 2,219 (modeled)
    Transitions: 152,000 (expanded)
  - Concurrent Regions: 17,000 (expanded)
  - Generated C-Code: 850,000 lines
  - Compile time: 2-3 min, response time: <2ms

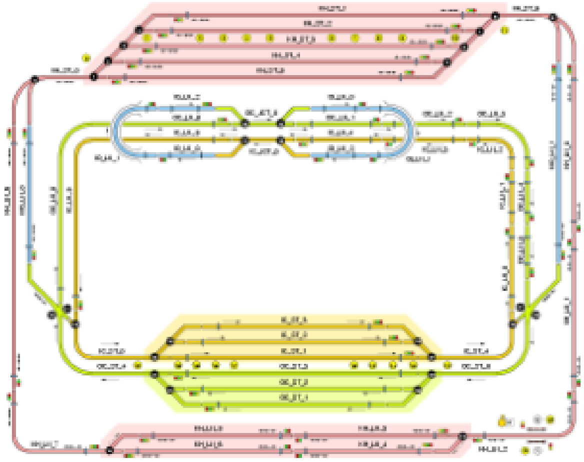
→ *Medium-Size Example* ✓



```
scchart Synchron {

    @Wrapper Button, ENTER
    input bool isEnterDown;

    @Wrapper Print
    output string outputText;

    initial state init
    --> done with isEnterDown / outputText = "Hello Synchron";

    state done;
}
```







Optional / additional aspects:

- **Railway Modeling DSL**
  A simplified modeling DSL, suited for occasions such as GirlsDay, that generates a controller SCChart to run the railway.

- **Lighthouses**
  Extend your controller to be able to control the two mini-lighthouses in H0 scale that were build In a previous student project
  They should be included in the railway installation physically.

- **Train Tracking**
  Develop a method to track the positions of the trains from start, giving the controller the information which train is where on the track.