

Tutorial 02: SCCharts

- Preliminaries
 - SCCharts
 - KIELER
- Tasks - DEADLINE: Mi, 30.04., 16:00
 - T2A1: Important Thoughts
 - T2A2: Tickets
 - T2A3: Modeling with SCCharts
 - T2A4: SCCharts Transformations
 - T2A5: Code Generation
 - T2A6: Hostcode (Bonus)

Preliminaries

SCCharts

Read and understand the following information about SCCharts:

- Lecture 20, Synchronous Languages WS13: <http://rtsys.informatik.uni-kiel.de/~rt-teach/ws13/v-synch/lectures/lecture20-handout4.pdf>
- Lecture 21, Synchronous Languages WS13: <http://rtsys.informatik.uni-kiel.de/~rt-teach/ws13/v-synch/lectures/lecture21-handout4.pdf>

Even so you must not understand every detail of the following two lectures you should read them to acquire additional information about the foundations of SCCharts. If you have questions about specific topics, don't hesitate to ask one of your supervisors.

- Lecture 22, Synchronous Languages WS13: <http://rtsys.informatik.uni-kiel.de/~rt-teach/ws13/v-synch/lectures/lecture22-handout4.pdf>
- Lecture 23, Synchronous Languages WS13: <http://rtsys.informatik.uni-kiel.de/~rt-teach/ws13/v-synch/lectures/lecture23-handout4.pdf>

Additional information about SCCharts within KIELER can be found here:

- PLDI Artifact submission: [PLDI'14 Artifact on SCCharts](#)

KIELER

Install and run a copy of KIELER. You should download the nightly semantics build <http://rtsys.informatik.uni-kiel.de/~kieler/files/nightly/kieler/>

(If you have already installed and set up your own KIELER development environment, you're free to use your own set up but be sure to update your code content to an actual state.)

Tasks - DEADLINE: Mi, 30.04., 16:00

T2A1: Important Thoughts

1. Write down the difference between Extended SCCharts and Core SCCharts. Use your own words.
2. What advantages has the Core SCCharts normalization?
3. Describe in own words what a basic block is.
4. Which constraints influence the ordering of a schedule?

T2A2: Tickets

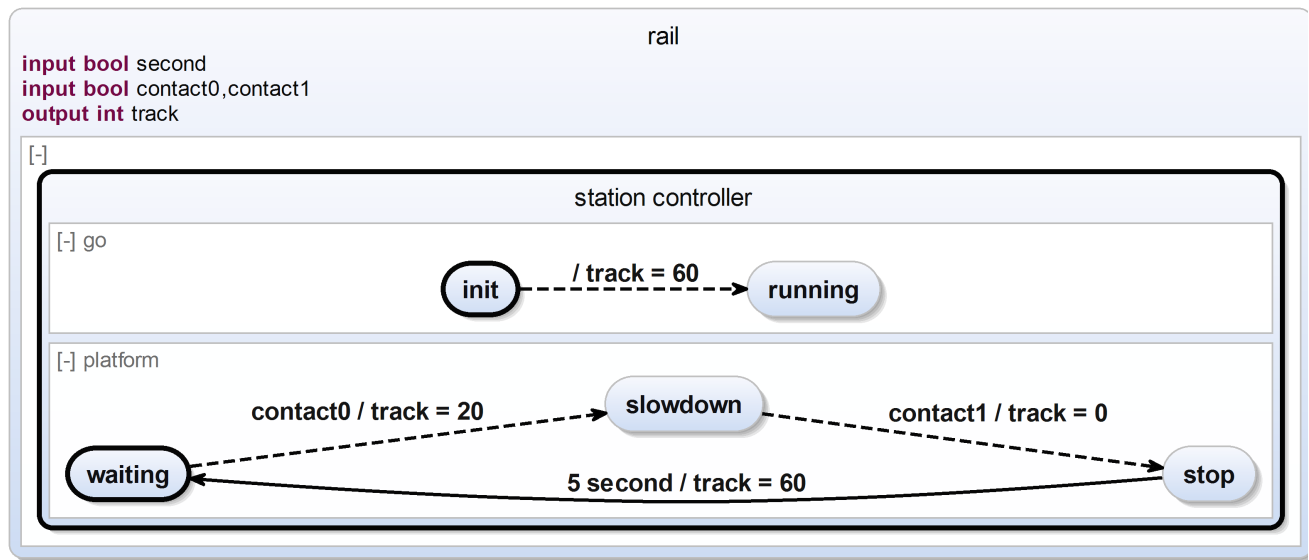
Create two tickets in the group's JIRA ticket system for two other course members of your choice. Be sure to select the semantics project and use Railway as component. Use your imagination for the actual ticket. Have fun! 🤪 React to the tickets that you receive within 24h and close the ticket with a useful comment.

Submit your ticket before Fr, 25.04., 12:00!

(You do not need to solve the actual issue. Just comment on the presented topic and close the ticket.)

T2A3: Modeling with SCCharts

Model the following SCChart using the textual SCChart language (*.sct).



The SCChart describes a simple railway station controller. Initially the track is powered with 60 units to allow any train standing on this platform to begin its journey. (We assume the train stands beyond contact0 and will not trigger it in the beginning.) Subsequently, the platform of this station slows down any train entering the station (indicated by `contact0`) and stops it when it passes `contact1`. After 5 seconds it may proceed its journey.

T2A4: SCCharts Transformations

- Transform all extended features of the SCChart in T2A3 to core variants. Explain in detail what features are extended features and how the particular transformation worked. Upload an image of the SCChart in Core SCChart form.
- Normalize your new SCChart. Upload an image of the SCCharts in normalized form.
- Convert the SCChart into its SCG representation. (You may use any valid SCG transformation KIELER offers.) Inspect the SCG and write down and explain all potential problems. Upload an image of the SCG.

T2A5: Code Generation

In order to generate executable code for the station controller, some potential problems need to be fixed. Describe these problems. Create a corrected version of the simple station controller and compile it with KIELER. Upload the fixed controller, the new SCG and the resulting c code. Explain which changes were necessary.

T2A6: Hostcode (Bonus)

Replace the contact and track variable assignments/reads with appropriate host code calls. Compile the SCChart again. Inspect the C code. Write down what changes to the code must be done to use this code in the railway simulation from tutorial 1 and the ModelGUI.