Real-Time Project SS 15 Home

This project is offered as Inf-AP-ES: Projektmodul - Echtzeitsysteme/Eingebettete Systeme (Layout oder Echtzeit) and as MSP1101: Masterprojekt - Echtzeitsysteme/Eingebettete Systeme (Layout oder Echtzeit). Both Bachelor and Master students may subsequently write a thesis building on results from the project.

Kick-Off Meeting

If you would like to join this project, you need to attend the kick-off meeting on April 15th at 10:15am. The meeting will be held in CAP4, room 1115.

What is this whole Wiki thing?

- It's the place where we post important information on the practical, such as due dates and similar information.
- It's also the place where we post tutorials. You will spend the first part of the practical working through the tutorials before starting your individual projects.
- And finally, it's the place where each group will document their project.

This Wiki is split into two halfs: the real-time and embedded systems project and the layout project. Depending on which of these projects you are part of, you will only contribute to that project's part of the Wiki.

Real-Time and Embedded Systems Project

Safety-critical systems

Reactive systems are computer systems that compute a function in a cyclic fashion. They continuously interact with their environment and react to inputs with computed outputs. Additionally reactive systems have to complete their computations in a certain amount of time which is typically a requirement derived from (physical) properties of their environment. Usually they are also embedded into their environment and hence also termed *embedded systems*. Often reactive systems control environments whose overall correct behavior is crucial. Failing in the sense of reactive systems not only means computing an incorrect output reaction but also not meeting the correct timing constraints. The failure of such systems or parts of it typically results in loss of human life or a financial disaster. It is essential that these systems do not fail and guarantee safety which is the most critical aspect. Hence, many real-time and embedded systems also are *safety-critical systems*. As the complexity of these systems rises, handling safety-critical tasks becomes more and more challenging. At the same time, the amount of embedded and critical systems is increasing. The model-based approach combined with synchronous languages aim to give such system designers better methods and tools to overcome this enormous task.

KIELER Miniature Wonderland - Demonstrating complexity

The Real-Time and Embedded Systems Group regularly demonstrates the complexity of such systems with the railway installation (see railway project 2014) of the Computer Science Department and/or with Lego Mindstorms. As part of this project you are going to extend our range of demonstrators for complex real-time systems.

- You are going to evaluate and choose appropriate demonstrators for a complex real-time system. An example for such a system is the Miniatur Wunderland in Hamburg,
- integrate new hardware (if required by your chosen demonstrator),
- and model your system within the model-based approach.

For now, it is planned to add mini RC cars and nano copters (see images) to our assortment but as this is *your* project, we also always appreciate further suggestions.

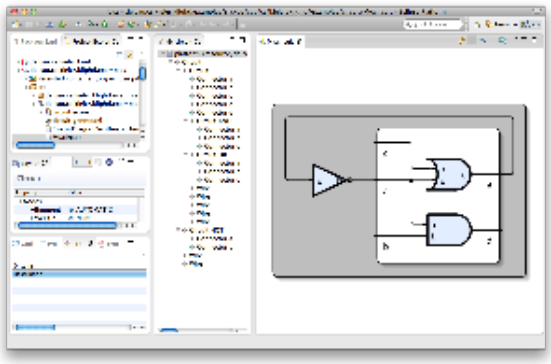
Layout Project

Graph layout algorithms are widely used to have computers generate visualizations of graph-like information. To that end, a graph layout algorithm generates a two-dimensional layout that consists of positioning data for nodes (represented as closed shapes) and routing data for edges (represented as curves). There are several approaches for designing graph layout algorithms [1,2].

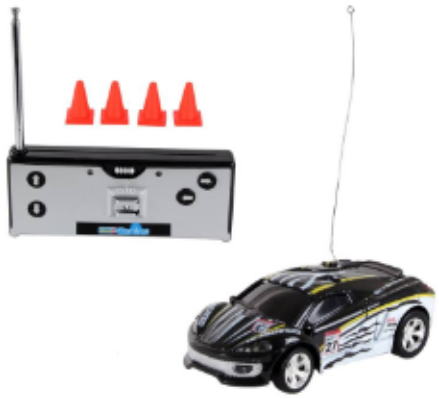
As a participant of this practical, you will work together with other students in order to extend specific layout algorithms or design visualizations using the Java programming language.

The algorithmic focus will be 1) on extending the layering phase of KLay Layered, a layout algorithm based on the approach of Sugiyama et al. [3], and 2) on the compact packing of unconnected subgraphs. The visualization focus will be on designing and implementing a visualization of UML Sequence Diagrams using KLighD, a framework for easily synthesizing graphical views.

The algorithms will be implemented according to the layout interface of the KIELER project, which is based on the Eclipse platform. The results may be published as part of the KIELER open source project.



This is KLighD showing a graphical view of a textually specified electric circuit.



Example: Mini RC Car by Revell (product details)

Example: Estes 4606 Proto X Nano R/C Quadcopter (product details)

Example: Estes Mad Cat RC Copter (product details)

Supervisors of this project are Steven Smyth, Christian Motika, Nis Boege Wechselberg, and Reinhard von Hanxleden. If you have questions or need assistance, please feel free to send us an E-mail and /or ask for an appointment.

A diagram layouted with our KLay Layered algorithm.

Responsible for this project are Christoph Daniel Schulze and Ulf Rüegg. If you need assistance, feel free to send us an email or ask for an appointment.

References

[1] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall, 1998.

[2] Tamassia, Roberto. Handbook of graph drawing and visualization. Chapman and Hall/CRC, 2013. (Online: https://cs.brown.edu/~rt/gdhandbook/)

[3] Christoph Daniel Schulze and Miro Spönemann and Reinhard von Hanxleden. Drawing Layered Graphs with Port Constraints. Journal of Visual Languages and Computing, Special Issue on Diagram Aesthetics and Layout, 2014.