Overview

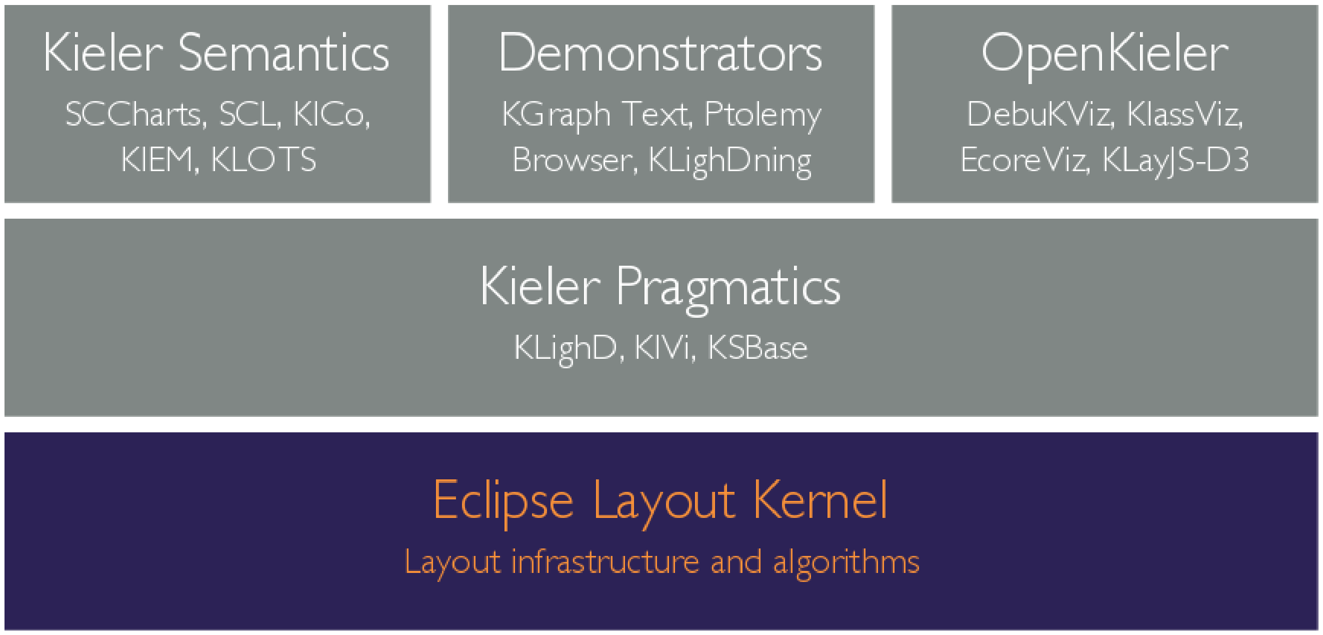
KIELER: The Kiel Integrated Environment for Layout (Eclipse Rich Client)

The *Kiel Integrated Environment for Layout Eclipse Rich Client*, or short *KIELER*, is a research project about enhancing the graphical model-based design of complex systems. The basic idea is to consistently employ automatic layout in all graphical components of the diagrams within the modeling environment. This opens up new possibilities for diagram editing, browsing, and dynamic visualizations (e.g. for simulation runs). Hence the focus of this project is the *pragmatics* of model-based system design, which can improve comprehensibility of diagrams, improve development and maintenance time, and improve the analysis of dynamic behavior.

This project is a continuation of the old KIEL project. The main differences are that KIELER aims to integrate a variety of modeling languages (not just Statecharts), and that it integrates into the rich client platform Eclipse. It attempts a tight integration with the modeling projects of Eclipse (EMF, GMF, Xtext, etc.) to bring new modeling approaches to a broad set of modeling paradigms and user communities.

KIELER is is developed as open source software, licensed under the Eclipse Public License and can thus be **downloaded and used** free of charge.

Being a research project, there are different people working in different areas of the project. The project is basically divided into the Layout, Pragmatics, and Semantics, but we also have a bunch of demonstrators and even a number of projects useful to end users published as part of OpenKieler:


Kieler Semantics
SCCharts, SCL, KICo, KIEM, KLOTS
Demonstrators
KGraph Text, Ptolemy Browser, KLighDning
OpenKieler
DebuKViz, KlassViz, EcoreViz, KLayJS-D3
Kieler Pragmatics
KLighD, KIVi, KSBase
Eclipse Layout Kernel
Layout infrastructure and algorithms

(Note: this image is also available as a PDF file and as the original SVG file and can freely be used in theses of our students. If you are not a student at our group but want to use this image, feel free to get in touch.)

Layout

The layout area provides the foundation for pretty much everything else in KIELER. Much effort in the Pragmatics area has gone into freeing the user from so-called "enabling steps": things a developer has to do to prepare for the model change he actually wants to make. This will usually be things like moving nodes around to make space for other nodes, taking care of edge routing etc. Good layout algorithms are necessary to be able to free the user from tasks like this that don't actually contribute anything to the final model.

The core component is KIML, the Kieler Infrastructure for Meta Layout. KIML provides the connection between graphical editors on the one hand, and layout algorithms on the other hand. Besides integrating various open source graph layout libraries such as OGDF and Graphviz, KIELER also includes custom layout algorithms in KLay, the Kieler Layouters component. The most important and most advanced algorithm is KLay Layered, a layer-based layout algorithm targeted primarily at the layout of data flow diagrams. However, KIELER also includes KLay Force, which provides implementations of force-based layout algorithms, as well as a KLay Planar, a planarization-based algorithm. Our custom layout algorithms are implemented in pure Java and can easily be adapted to be used outside of Eclipse as well. There's also the KWebS component, which makes our layout algorithms available as a web service. Finally, GrAna, our graph analysis component, allows us to compute a whole lot of different metrics for diagrams, which is handy when it comes to evaluating the performance of our layout algorithms.

Learn more about layout

Pragmatics

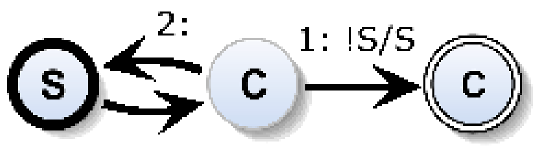
With the term 'pragmatics' (don't confuse it with 'pragmatism') we denote the set of all practical aspects that influences the daily work of modelers in the context of Model-Driven Engineering (MDE). This comprises the creation and modification of models (abstract syntax as in the sense of the MVC) as well as the synthesis of diverse views on those models tailored to different stakeholders and different purposes.

Exemplary projects addressing these topics are the KIELER Structure-based Editing (KSbasE) and the KIELER Lightweight Diagrams (KLighD). The first one aims at providing operations for changing models based on its abstract syntax (introduce new elements, modify them in some way, connect them, or delete them). The second one targets the synthesis of graphical representations of models or model excerpts in a lightweight and transient way. This means, representations are synthesized on the modeler's demand and dismissed later on without saving them on the disk, unless the modeler wants to do that. Both techniques are enabled by the ability of automatically arranging diagram elements as provided by our layout corner.

Learn more about pragmatics

Semantics / SCCharts

We also provide an infrastructure to define execution semantics for a meta model. This can be a simulator based on C (see S/SC projects) or Ptolemy (see KlePto project). Simulators can easily be integrated using the KIELER Execution Manager (KIEM). We also support textual simulations (see Esterel and S project). Another recent main contribution are SCCharts. To get an overview about all projects concerning semantics in KIELER follow the link below.



Learn more about semantics

Demonstrators

Demonstrators are editors we use to test and demonstrate the technologies developed in the other three areas.

Learn more about demonstrators

OpenKieler

A number of projects that started out as demonstrators were so useful to end users that they ended up in a separate area, *OpenKieler*. This project is hosted on GitHub and everyone is invited to contribute. *DebuKViz* helps programmers debug applications visually by generating diagrams of data structures on the fly. *KlassViz* is a lightweight class diagram generator. *EcoreViz* displays Ecore models graphically on the fly. KLayJS-D3 finally provides a binding between the KLayJS layout algorithm and the popular D3 graph drawing library.

Go to OpenKieler at GitHub

Discontinued Projects

Since this is a research project, we cannot provide support for all subprojects forever. Those that are no longer considered in active development and future releases are listed on the discontinued projects page. We have still some archived documentation for these old subprojects: