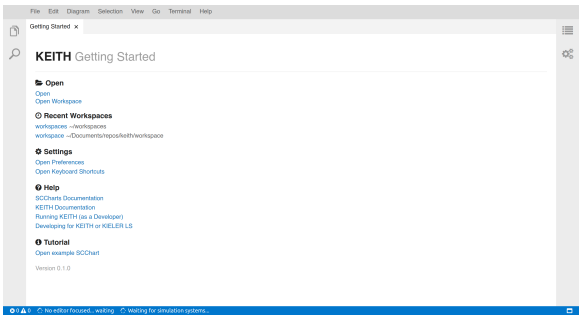
KEITH

KEITH[1][2] is an IDE for Model -Driven Development (MDE) based on KIELER[3]. It uses KIELER as a language server (LS) to provide rich client features for SCCharts[4], Lustre, Esterel, ELK, KGraph, and some intermediary languages.

KEITH runs in a browser. You can either start KEITH and connect to an URL (i.e. localhost:3000) via a browser (this will henceforth be called the browser version of KEITH), or start an electron app with a browser that runs KEITH inside (henceforth called the electron version of KEITH). Since the electron app might use a different chromium browser some UI elements might be displayed differently, but both version work the same. The electron version is bundled via the electron-builder framework to build an appimage or an archive into the henceforth called product. Both version have two alternatives to connect to the language server (LS). KEITH can connect to the LS via a socket. This is the only way to debug the LS. In the product KEITH connects to the LS via stdio/out and has to know the relative path to the LS application.



General Usage Philosophy

KEITH is developed via the Theia framework that is based on VSCode. It should be designed to integrate in the look and feel of VSCode. The user should be able to use this tool without clicking buttons in extensive dialogs as it can be the case in Eclipse. Since KIELER is an Eclipse application KEITH should not only focus on adapting everything that is already in KIELER and bring it to another platform but focus on the main features and make them easier accessible and minimize the clutter created via extensive button landscapes and dialogs.

Developing and designing the UI via web technologies promises to make it easier to design user interfaces with Human-computer interaction (HCI) in mind. The KIELER-LS for KEITH requires to separate UI and functionality also in KIELER and allows to develop a KEITH client with less dependencies to language specific implementations and promises are more modular framework. Moreover, the resulting LS could be used to adopt SCCharts toolchain for different IDEs.

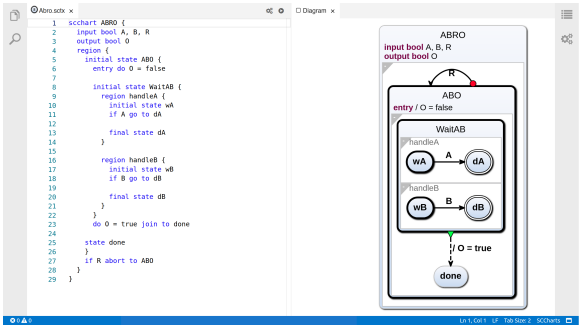
As mentioned before, KEITH is able to run in your browser. However, the backend of KEITH and the language must not run on your system. One goal of KEITH is to use it as an online Editor such as JSFiddle for JS and use tools such as Gitpod which uses the kubernetes framework to manage dockerized workspace and IDE instances. This project is currently work in progress.

> Normally the browser version is used for debugging and the electron version is delivered as download, but both work the same and you can also use the developer tools inside the electron version to debug the application.

> You can download the nightly build for the KEITH electron app and the language server here (currently no working build for MacOS).

Error rendering macro 'toc'

null



Downloads

KEITH Nightly: https://rtsys.informatik.uni-kiel.de/~kieler/files/nightly/sccharts-integration/

KEITH Language Server: https://rtsys.informatik.uni-kiel.de/~kieler/files/nightly/sccharts-integration/

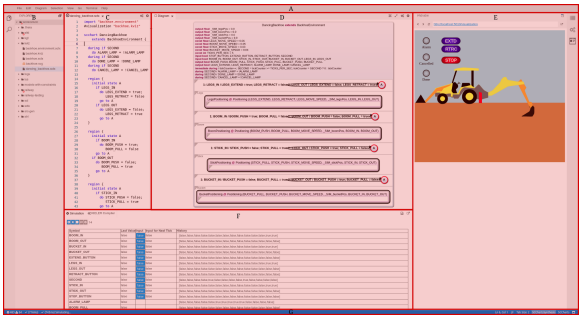
[1] Sören Domrös. *Moving Model-Driven Engineering from Eclipse to Web Technologies*, November 2018

[2] Niklas Rentz. *Moving Transient Views from Eclipse to Web Technologies*, November 2018

[3] Reinhard von Hanxleden, Hauke Fuhrmann, and Miro Spönemann. “KIELER—The KIELIntegrated Environment for Layout Eclipse Rich Client”. In: Proceedings of the Design,Automation and Test in Europe University Booth (DATE ’11). Grenoble, France, 2011.

[4] Reinhard von Hanxleden, Björn Duderstadt, Christian Motika, Steven Smyth, Michael Mendler, Joaquín Aguado, Stephen Mercer, and Owen O’ Brien. SCCharts: Sequentially Constructive Statecharts for Safety-Critical Applications. In Proc. ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI’14), Edinburgh, UK, June 2014.

Using KEITH



Theia has four bars: The left var (B), the main bar (C and D), the right bar (E), and the bottom bar (F). All bars are splittable and all widgets can be moved to any bar.

A: Menu bar: Holds the main menu.

B: Left bar: The left bar usally holds the explorer widget and the search widget (only seen as an icon).

C: Left part of Main bar, the editor widget: In the left part of the main bar resides the editor widget. In the top right of the editor widget, in the so called tab bar are some editor specific commands accessible. It is possible to compile or simulate the currently opened model via these buttons. These buttons will not directly invoke the compilation but redirect the user to the command palette.

D: Right part of main bar, the diagram widget: Shows the current model as diagram. Can also be used to show snapshots of a model to model transformation. During simulation the current state is highlighted in red and the last state and taken transition are highlighted in light blue. In the current diagram all "A" states are currently active. In the top right of the diagam widget is the tab bar of this widget. It includes commands to:

- Center the diagram
- Fit the diagram to the widget size
- Compile the shown mode
- Simulate the shown model

E: The right bar: The right bar currently holds the browser preview widget. It can be used to display a visualization of a simulated model. This view can also be opened in another browser tab. The right bar does usually also hold the outline widget, the output widget, and the diagram option widget.

F: The bottom bar: Holds the simulation widget and the compiler widget (in the background). The simulation widget shows the values of a simulated model in form of a table and allows the user to set inputs. Playing, pausing, stepping, and stopping a simulation is supported. On the top right of the simulation widget is the tab bar of this widget. It includes buttons to open the simulation visualization which is currently seen in the right bar.

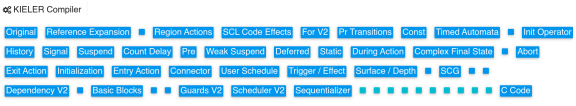
G: Status bar: Shows the status of tool including:

- errors and warnings in workspace
- Status of compilation and simulation
- Current line number and column
- The end of line sequence (LF)
- Current indentation style (here Tab Size 2)
- Current diagram synthesis (SCChartsSynthesis), which can be changed via clicking it
- The language in the active editor (here SCCharts)

The blue color indicates that the tool has a stable connection to its language server. Yellow indicates a connection loss.

KEITH has some widgets by default, such as the problems-widget, the outline-widget, the output-widget, and the search-widget. All existing widgets are found via the *View* menu entry or via the command palette.

KIELER compiler widget

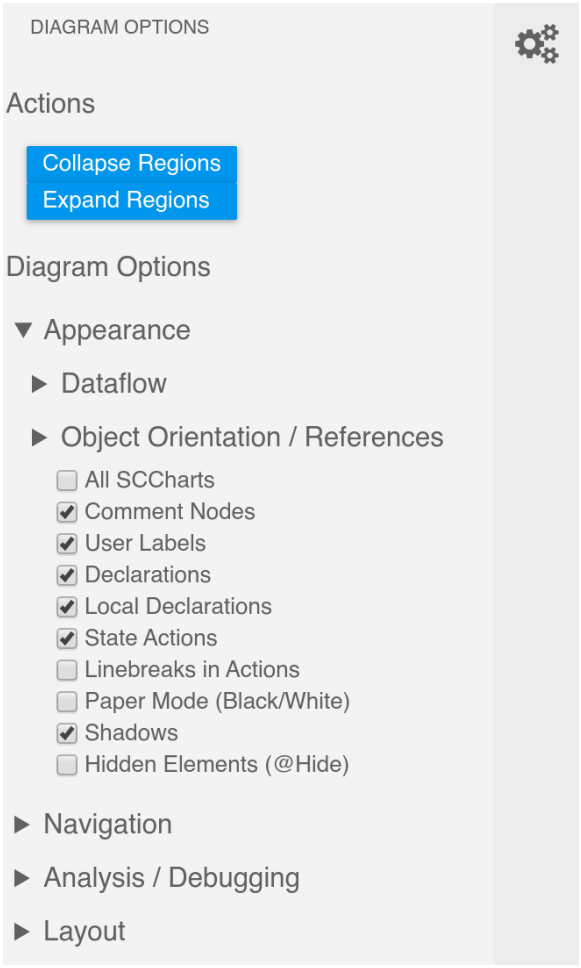


The compiler widget shows all snapshots created via the model to model transformations of the currently model. These buttons are clickable and send a request to the LS to draw the clicked snapshot in the diagram view. Snapshots are labeled according to their processor label. Snapshots without labels are models of the previous processor.

This widget is only used for displaying these snapshots. All compilation features can be sued via the tab bar of the editor or diagram widget or via the command palette.

> **Hint**
>
> You can navigate through the snapshots via *Alt+G*/*Alt+J*

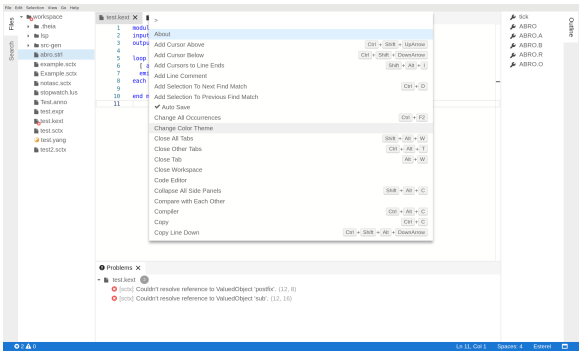
Diagram-option widget



The diagram option widget allows to set the diargam options for the currently shown diagram.

On change of the diagram model it updates the options of the diagram via the LSP. The options seen here are SCCharts options.

The command palette



The command palette opens by pressing *F1* or opening it via the menu *View>Find Command...*

On open it first displays the recently used commands and after that all commands in lexicographic order. The command palette can be searched via fuzzy search and supports regular expressions. The displayed commands have of a name (on the left side) and optionally a shortcut to invoke them (on the right side of the command palette). Some commands, such as Auto Save also have an icon associated to them.

Via *Ctrl+P* the workspace can be searched for files to open. This can also be access via *F1* and deleting the > character in the command palette.

Modeling SCCharts

See Quick Start Guide#ModelingSCCharts for details.

Known Problems

Mac

If the app does not start open it via the terminal with

```
./keith.app/Contents/MacOS/keith
```