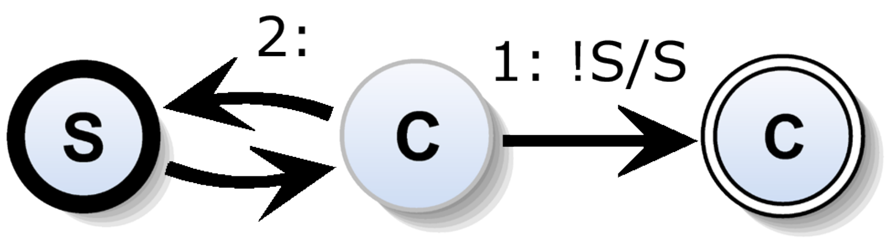
SCCharts (pre 1.0)



Sequentially Constructive Statecharts (SCCharts)

SCCharts [5] is a new visual synchronous language that is designed for specifying safety-critical reactive systems. SCCharts uses a new statechart notation similar to Harel Statecharts [3] and provides deterministic concurrency based on a synchronous model of computation (MoC), without restrictions common to previous synchronous MoCs like the Esterel constructive semantics [2]. Specifically, we lift earlier limitations on sequential accesses to shared variables, by leveraging the sequentially constructive MoC [4]. Thus SCCharts in short are SyncCharts [1] syntax plus Sequentially Constructive semantics.

The key features of SCCharts are defined by a very small set of elements, the Core SCCharts, consisting of state machines plus fork/join concurrency.

Conversely, Extended SCCharts contain a rich set of advanced features, such as different abort types, signals, history transitions, etc., all of which can be reduced via semantics preserving model-to-model (M2M) transformations into Core SCCharts. Extended SCCharts features are syntactic sugar because they can be expressed by a combination of Core SCCharts features.

On the one hand this eases the compilation and makes it more robust because it reduces its complexity. On the other hand, using Extended SCCharts features, a modeler is able to abstract away complexity of his or her SCCharts model which increases robustness and readability of a model. This approach enables a simple yet efficient compilation strategy and aids verification and certification.

For our KIELER SCCharts implementation, you may also like to download an SCCharts Tutorial or the SCCharts Cheat Sheet.

[1] 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.

[2] Christian Motika, Steven Smyth, and Reinhard von Hanxleden. Compiling SCCharts — A case-study on interactive model-based compilation. In Proceedings of the 6th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2014) , volume 8802 of LNCS, pages 443–462, Corfu, Greece, October 2014.

[3] Francesca Rybicki, Steven Smyth, Christian Motika, Alexander Schulz-Rosengarten, and Reinhard von Hanxleden. Interactive model-based compilation continued — interactive incremental hardware synthesis for SCCharts. In Proceedings of the 7th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2016), LNCS, 2016.

[4] Reinhard von Hanxleden, Michael Mendler, Joaquín Aguado, Björn Duderstadt, Insa Fuhrmann, Christian Motika, Stepjen Mercer, and Owen O'Brien. S equentially Constructive Concurrency - A conservative extension of the synchronous model of computation, In Proc. Design, Automation and Test in Europe Conference (DATE'13), Grenoble, France, March 2013.

[5] Charles André. Semantics of SyncCharts, Technical Report ISRN I3S/RR-2003-24-FR, I3S Laboratory, Sophia-Antipolis, France, April 2003.

[6] Gerad Berry. The foundations of Esterel, In G. Plotkin, C. Stirling, and M. Tofte, editors, Proof, Language, and Interaction: Essays in Honour of Robin Milner, pages 425-454, Cambridge, MA, USA, 2000.

[7] David Harel. Statecharts: A visual formalism for complex systems, Science of Computer Programming, 8(3):231-274, June 1987.

Downloads

Downloads - KIELER SCCharts Product

The KIELER SCCharts Product includes the SCCharts editor and compiler.

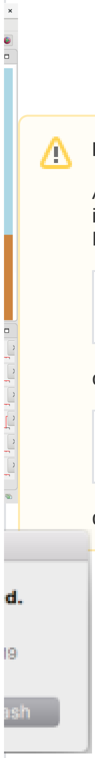
Release 1.2.0

KIELER SCCharts Product v. 1.2.0 (2021-07-05, based on Eclipse 2021-06). Available platforms:

- Linux: x86 64bit
- Mac OS X: x86 64bit
- Windows: x86 64bit

Nightly Builds

- http://rtsys.informatik.uni-kiel.de/~kieler/files/nightly/sccharts/



KIELER App on Mac OS

After downloading KIELER, Mac OS quarantines the application and, for some reason, considers the app broken and wants you to move it into the Trash.
Hence, if you want to use KIELER on your Mac, you have to remove the quarantine flag manually. You can use the following command:

```
xattr -rc com.apple.quarantine ./Kieler.app
```

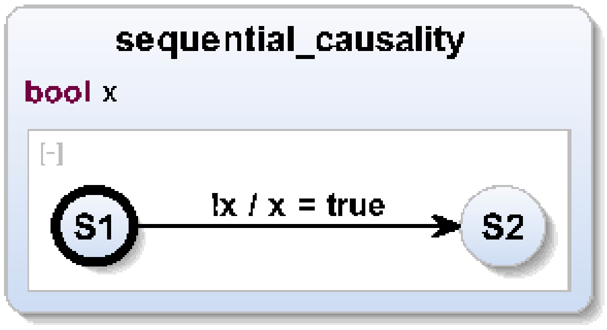
or

```
xattr -rd com.apple.quarantine ./Kieler.app
```

depending on the Mac OS version

Sequentially Constructive MoC

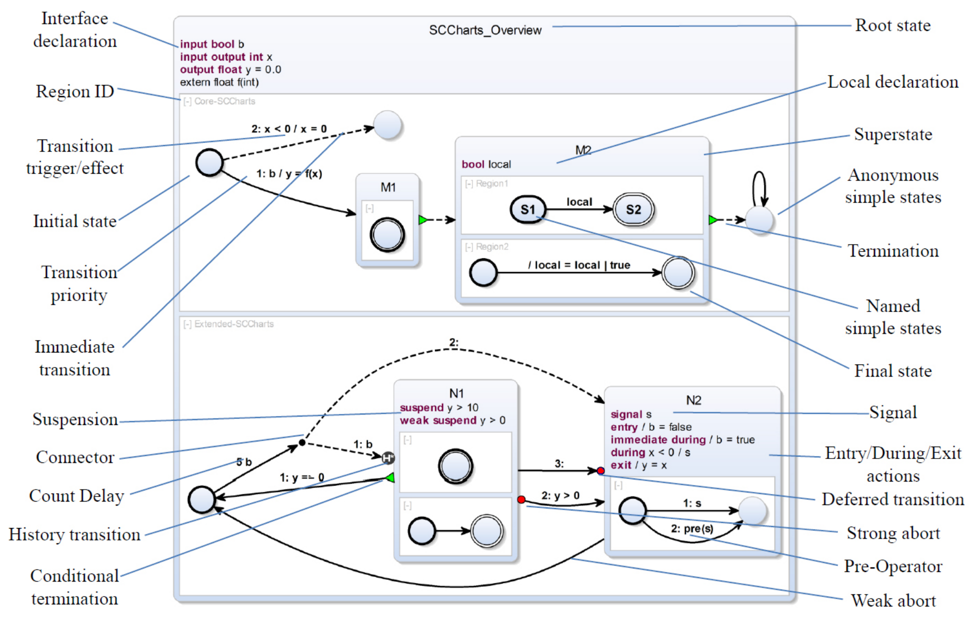
In contrast to SyncCharts (see [1] Charles André) a signal (or variable) in SCCharts is allowed to be emitted with different values in the same macro tick (if the emissions are schedulable according to the SC MoC). The following example is forbidden in SyncCharts but not in SCCharts.



SyncCharts: x cannot be absent and present in the same macro tick.
SCCharts: Deterministic ordering possible: If x is false **then** take the transition and set x to true.

Core & Extended SCCharts

A core SCChart is composed of elements of a minimal set of constructs. Additional constructs and syntactical sugar (f.e. actions, suspend) are introduced in extended SCCharts. Every extended SCCharts can be transformed into a core SCChart.

Core SCChart	Extended SCCharts	Graphical comparison
• Simple States • Hierarchical States (aka Composite States) • Regions • Transitions (weak abort) • Immediate Transitions • Normal Termination • Variables (primitive types) • Interface Declaration	Core SCCharts + • Connector (aka Choice) • Strong Abort • Weak Abort • History Transition • Suspension • Entry Action • During Action (aka Inner Action) • Exit Action • Signal • Pre • Count Delay • Conditional Termination • Initialization • Complex Final State • Deferred Transition	 Interface declaration Region ID Transition trigger/effect Initial state Transition priority Immediate transition Suspension Connector Count Delay History transition Conditional termination SCCharts_Overview input bool b input output int x output float y = 0.0 extern float f(int) [-] Core-SCCharts 2: x < 0 / x = 0 1: b / y = f(x) M1 M2 bool local [-] Region1 S1 local S2 [-] Region2 / local = local | true [-] Extended-SCCharts 2: N1 suspend y > 10 weak suspend y > 0 N2 signal s entry / b = false immediate during / b = true during x < 0 / s exit / y = x 5 b 1: b 1: y =- 0 3: 2: y > 0 1: s 2: pre(s) Root state Local declaration Superstate Anonymous simple states Termination Named simple states Final state Signal Entry/During/Exit actions Deferred transition Strong abort Pre-Operator Weak abort [click to enlarge]

Modeling & Compiling SCCharts

SCCharts can be modeled using our KIELER SCChrats editor and compiler (download). The modeling editor is a textual editor based on the itemis Xtext framework. The language used to model SCCharts textually is called SCT and documented here. A quick start guide introducing first steps from downloading over modeling to compiling SCCharts can be found here.

Project Status

Subproject/Extension	Progress		Released
SCCharts Editor (*.sct)	Implemented and tested		0.9.0
SCG Editor	Implemented and tested		0.9.0
SCL Editor	Implementation not yet finished		cancelled
Extended 2 Core SCCharts	Implemented, not yet fully tested		0.9.0
Core 2 Normalized SCCharts	Implemented, not yet fully tested (some known bugs)		0.9.0
Normalized SCCharts 2 SCG	Implemented, not yet fully tested (some known bugs)		0.9.0
SCG 2 Sequential SCG	Implemented and partly tested, straightforward scheduler for 0.9.0 release, enhanced scheduler planned for 0.10.0 release.		0.9.0

SCG 2 C	Implemented by transformation via common S language (this can also be translated into Java -> SJL)		0.9.0
Online Compiler and Command Line Tools	Implemented and partly tested		0.10.0
Simulation	Implemented and partly tested		0.10.0
Hardware Circuit Synthesis and Simulation	Implemented and partly tested		0.11.0

Known Limitations

- **Normalization** may result in conditions where there actually is no conditions, this should optimized manually
- **SCG Generation** currently produces unoptimized hierarchy levels, e.g., fork nodes with just one successor node should be eliminated
- **Scheduling** of unconnected SCG exit nodes is currently not possible