

# Sequential Constructiveness and SCCharts for Safety-Critical Applications\*

Reinhard von Hanxleden, Björn Duderstadt, Insa Fuhrmann, Christian Motika, Steven Smyth  
Dept. of Computer Science, Kiel University, 24098 Kiel, Germany  
{rvh, bdu, ima, cmot, ssm} @informatik.uni-kiel.de

Michael Mendler, Joaquín Aguado  
Dept. of Computer Science, Bamberg University, 96045 Bamberg, Germany  
{michael.mendler, joaquin.aguado} @uni-bamberg.de

Stephen Loftus-Mercer, Owen O'Brien  
National Instruments, Austin, TX 78759, USA  
{stephen.loftus-merc, owen.o'brien} @ni.com

**Abstract:** We present a new visual language, SCCharts, designed for the model-driven engineering of safety-critical reactive systems. SCCharts provide determinate concurrency based on a synchronous model of computation (MoC), without restrictions common to previous synchronous MoCs.

SCCharts [vHDM<sup>+</sup>14] use a statechart notation [Har87] and provide determinate concurrency, based on a synchronous model of computation (MoC) [And04]. We lift earlier limitations on sequential accesses to shared variables, by leveraging the sequentially constructive (SC) MoC [vHMA<sup>+</sup>14, AMvHF14]. In essence, the SC MoC extends the classical synchronous MoC by allowing variables to be read and written arbitrarily as long as sequentiality expressed in the program provides sufficient scheduling information to exclude race conditions. Previous approaches have excluded test-then-set code such as “if (!initialized) { init(); initialized = true; }” because it requires a write after a read. The SC MoC permits a compiler to safely admit such statements.

The semantics and key features of SCCharts are defined by a very small set of language 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., as illustrated in Fig. 1; all of these can be reduced via model-to-model transformations into Core SCCharts [MSvH14]. This approach enables a simple yet efficient compilation strategy and aids verification and certification. An open-source, class-room tested SCChart editor/compiler is available in the Eclipse-based modeling framework KIELER<sup>1</sup>.

---

\*This work was supported by the German Science Foundation (DFG HA 4407/6-1 and ME 1427/6-1) as part of the PRETSY project.

<sup>1</sup><http://www.informatik.uni-kiel.de/rtsys/kieler/>

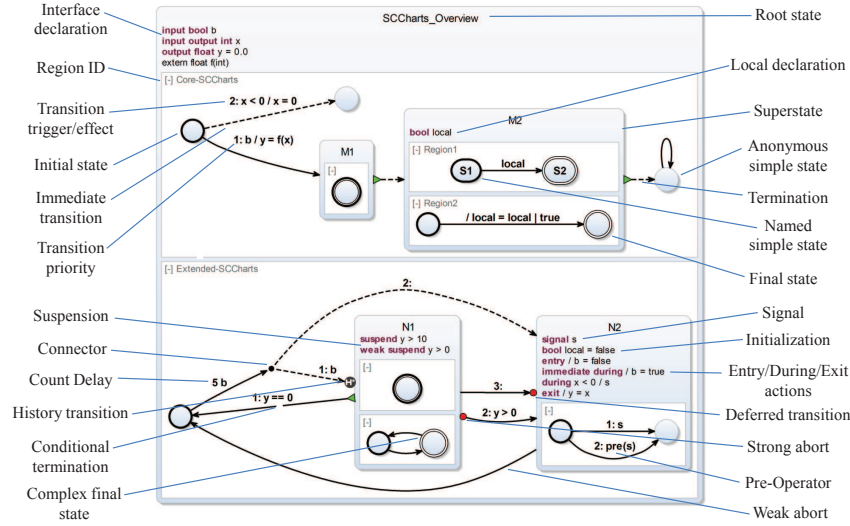


Figure 1: Syntax overview of SCCharts (from [vHDM<sup>+</sup>14]). The upper region uses Core SCCharts language elements only, the lower region illustrates Extended SCCharts features.

## References

- [AMvHF14] Joaquín Aguado, Michael Mendler, Reinhard von Hanxleden, and Insa Fuhrmann. Grounding Synchronous Deterministic Concurrency in Sequential Programming. In *Proceedings of the 23rd European Symposium on Programming (ESOP'14)*, LNCS 8410, pages 229–248, Grenoble, France, April 2014. Springer.
- [And04] Charles André. Computing SyncCharts Reactions. *Electronic Notes in Theoretical Computer Science*, 88:3–19, October 2004.
- [Har87] David Harel. Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming*, 8(3):231–274, June 1987.
- [MSvH14] 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.
- [vHDM<sup>+</sup>14] 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. ACM.
- [vHMA<sup>+</sup>14] Reinhard von Hanxleden, Michael Mendler, Joaquín Aguado, Björn Duderstadt, Insa Fuhrmann, Christian Motika, Stephen Mercer, Owen O’Brien, and Partha Roop. Sequentially Constructive Concurrency—A Conservative Extension of the Synchronous Model of Computation. *ACM Transactions on Embedded Computing Systems, Special Issue on Applications of Concurrency to System Design*, 13(4s):144:1–144:26, July 2014.