

The KIELER SCCharts Editor

A Modular Open-Source Modeling Suite with Automatic Diagram Synthesis

Steven Smyth*, Alexander Schulz-Rosengarten*, Christian Motika[†], Reinhard von Hanxleden*

*Department of Computer Science, Kiel University, Kiel, Germany, E-mail: {ssm, als, rvh}@informatik.uni-kiel.de

[†]Lufthansa Technik, Hamburg, Germany, E-mail: christian.motika@lht.dlh.de

By using high-level domain specific languages for designing programs, for example in the hardware domain, the model-based approach promises a more transparent and efficient development process. The main task is to synthesize the program into the target language, e. g. a circuit, which is done by the compiler. Designing, understanding, and maintaining a compiler still remains a difficult task. Although the user has some degree of control over the compilation, compilers are usually black boxes. It is relatively hard to grasp beforehand how the final or intermediate results will look like. In addition, the results are not necessarily executable models or comprehensible in a human-readable way.

By leveraging the complete compilation workflow of chained model-to-model transformation systems to a meta level, both, the tool developer and the modeler, can benefit from an interactive development process. Combined with modern transient view and automatic layout technologies, creating and working with model-to-model transformation systems becomes transparent and less time consuming. We argue that modern development tools, such as model-based compilers, besides producing the correct result, can and should also guide the modeler to potential issues and provide means to understand what is happening during the transformations.

The KIELER¹ SCCharts Editor is such a modular, open-source modeling suite, using the synchronous language SCCharts² [5] as main demonstrator, but can handle arbitrary many languages. The SCCharts statechart dialect is a successor of SyncCharts, which can be seen as a graphical version of Esterel. The editor supports high- and low-level language compilation [1], automatic syntheses of intermediate results

using a transient view framework [3] with automatic layout via the Eclipse Layout Kernel (ELK)³, and deployment to different platforms, both software and hardware [2]. The modular concept of the compiler framework [4] allows for rapid application and prototype development. The concepts used to build the reference SCCharts compiler can be applied to various other languages or domains and are not bound to SCCharts.

REFERENCES

- [1] 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 *LNCIS*, pages 443–462, Corfu, Greece, October 2014.
- [2] 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)*, volume 8802 of *LNCIS*, pages 443–462, Corfu, Greece, October 2016.
- [3] Christian Schneider, Miro Spönemann, and Reinhard von Hanxleden. Just model! – Putting automatic synthesis of node-link-diagrams into practice. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC '13)*, pages 75–82, San Jose, CA, USA, September 2013.
- [4] Steven Smyth, Alexander Schulz-Rosengarten, and Reinhard von Hanxleden. Towards interactive compilation models. In *Proceedings of the 8th International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2018)*, Limassol, Cyprus, November 2018.
- [5] Reinhard von Hanxleden, Björn Duderstadt, Christian Motika, Steven Smyth, Michael Mender, 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)*, pages 372–383, Edinburgh, UK, June 2014. ACM.

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

²<http://www.sccharts.com>

³<https://www.eclipse.org/elk>

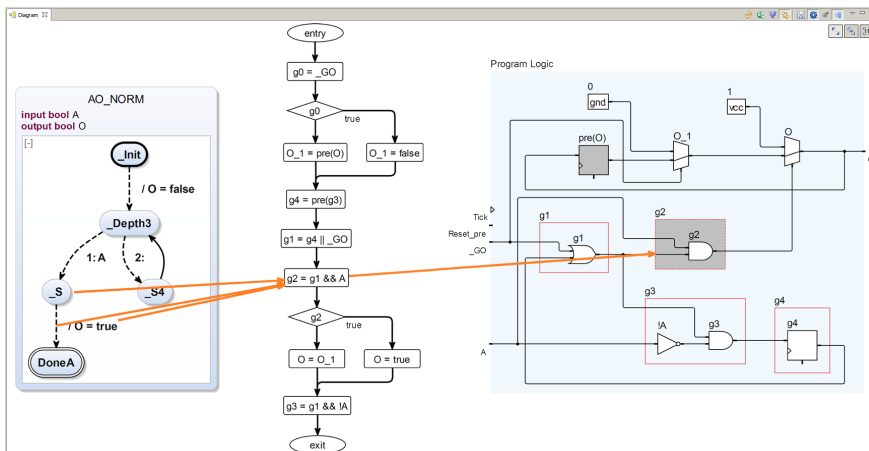


Fig. 1. Different life cycles of an SCCharts model in KIELER. The graphical views (here (1) the normalized model, (2) a controlflow graph representation, and (3) a generated circuit) are synthesized automatically from the textual model and the intermediate results of the transformation steps. The developer can select specific transformation steps (e. g. 1-3) and then inspect model element relations (orange arrows) between the selected intermediate results interactively. Since the compiler framework is fully modular and generic, arbitrary transformation steps can be performed.