

6 Conclusions

In this paper, we have shown how the introduction of clocks solves the issues encountered in event-based models written in Modelica. We have also shown that clocks could be harmoniously integrated into the language without compromising simplicity nor expressiveness, on the contrary: models could be made more generic thanks to the modular-friendly aspects of clock calculus (which would help a lot in the design of industrial-strength libraries) and, since it would be possible to express more subtle relationships between event sources, development, debugging and maintainance of models involving discrete-time aspects would be easier.

The Modelica community, in the course of the Modelica 4 design process, is going to consider the problem of synchrony. We hope that modular aspects and expressiveness resulting from the introduction of a full clock calculus will be retained as key features of the new language.

7 Acknowledgements

Many thanks to Ramine Nikoukhah for having pointed out issues related to synchrony and for having been the first to suggest having a look at the concept of clock in the context of Modelica. I would also like to thank Simon Bliudze and Marc Pouzet for interesting discussions about the theoretical aspects of hybrid systems.

References

- [1] R. Nikoukhah, *Activation Inheritance in Modelica*, EOOLT, 2008
- [2] R. Nikoukhah, S. Furic, *Synchronous and Asynchronous Events in Modelica: Proposal for an Improved Hybrid Model*, 6th international Modelica conference, 2008
- [3] A. Benveniste, P. Le Guernic, and C. Jacquemot, *Programming with events and relations: the Signal language and its semantics*, 1991
- [4] S. Bliudze., D. Krob, *Modelling of Complex Systems: Systems as dataflow machines*, 2009