A Modelica Library for Real-Time Coordination Modeling

Uwe Pohlmann[1], Stefan Dziwok[1], Julian Suck[1], Boris Wolf[1],
Chia Choon Loh[2], and Matthias Tichy[3]
[1]Software Engineering Group, [2]Control Engineering and Mechatronics Group,
Heinz Nixdorf Institute, University of Paderborn, Paderborn, Germany,
[upohl|stefan.dziwok|jsuck|borisw|chia.choon.loh]@upb.de
[3]Software Engineering Division, Chalmers University of Technology
and University of Gothenburg, Sweden, tichy@chalmers.se

Embedded software is an important part of today's life. One reason for the increasing trend of embedded systems is the introduction of coordination between previously autonomous systems resulting in complex systems of systems in order to realize functionality which cannot be achieved by each system alone [2]. The car industry is an example where vehicles communicate with other vehicles in order to extend the car's vision to areas obstructed by other vehicles. This coordination requires an intensive communication between the systems under real-time constraints.

Modelica in version 3.2 and the StateGraph2 library for state-based modeling lack appropriate support for the sketched case of modeling the real-time coordination between autonomous systems as this coordination is often realized by communication using asynchronous messages and complex state-based behavior.

In this paper, we present a Modelica library for modeling communication under hard real-time constraints. Our library extends the StateGraph2 library by providing support for (1) synchronous and asynchronous communication and (2) rich modeling of real-time behavior. These extensions are based on our previous work on the MECHATRONICUML modeling language [1]. We illustrate our extension using a robot platooning scenario.

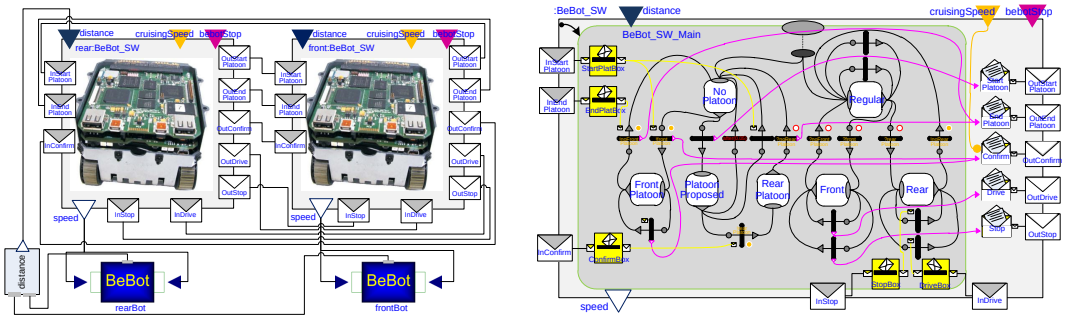

Figure 1: Platoon Scenario Instance and Behavior Model

This work was developed in the project ENTIME. The project ENTIME is funded by the state of NRW, Germany and the EUROPEAN UNION, European Regional Development Fund.

References

[1] S. Becker, C. Brenner, S. Dziwok, T. Gewering, C. Heinzemann, U. Pohlmann, C. Priesterjahn, W. Schäfer, J. Suck, O. Sudmann, and M. Tichy. The mechatronicuml method - process, syntax, and semantics. Technical Report tr-ri-12-318, Software Engineering Group, Heinz Nixdorf Institute, University of Paderborn, 2012.

[2] W. Schäfer and H. Wehrheim. The Challenges of Building Advanced Mechatronic Systems. In Lionel C. Briand and Alexander L. Wolf, editors, *FOSE*, pages 72–84, 2007.