From UML/SysML to Matlab/Simulink: Current State and Future Perspectives

Yves Vanderperren, Wim Dehaene

EE Dept., ESAT-MICAS, Katholieke Universiteit Leuven, B-3001 Leuven, Belgium

1. Motivation

Several recent EDA surveys [1-2] confirm that The Mathworks Matlab/Simulink and the Unified Modelling Language (UML) are both gaining increased attention as Electronic System Level (ESL) languages. While Matlab is commonly used to model signal processing intensive systems, UML has the potential to support innovative ESL methodologies which tie the architecture, design and verification aspects in a unified perspective. Integrated design flows which exploit the benefits of the complementarity between UML and Matlab provide an interesting answer to the issues of mono-disciplinary modeling and the necessity of moving beyond point-tool solutions [3]. This paper summarizes how UML and Matlab/Simulink can be associated and what is the impact of SysML, a new modeling language based on UML to describe complex heterogeneous systems.

2. UML and Matlab/Simulink

Two different approaches allow coupling UML and Matlab/Simulink models: co-simulation, and integration based on a common underlying executable language.

In case of co-simulation, the Simulink and the UML simulations communicate via an intermediate coupling tool. Ensuring a consistent notion of time is crucial to guarantee proper synchronization between the UML tool and Simulink. Both simulations exchange signals and run concurrently in the case of duplex synchronization, while they run alternatively in the sequential case. The former solution increases the simulation speed, whereas the time precision of the exchanged signals is higher in the latter case. As an example, the co-simulation approach is implemented in the Exite tool from Extessy AG, which allows coupling a.o. a Simulink model with ARTiSAN Software Real-Time Studio or I-Logix Rhapsody. A similar simulation platform is proposed in [4] for IBM Rational Rose RealTime.

The alternative approach is to resort to a common execution language. In absence of support for Matlab code generation from UML, the classical solution is to generate C/C++ code from Matlab, using Matlab Compiler or Real-Time Workshop, and link it to a C++ implementation of the UML model. This solution is adopted, for instance, in the Constellation framework from Real-Time Innovation and in the GeneralStore integration platform [5]. Both tools provide a unified representation of the system at model level on top of at code level. The Simulink subsystem appears in Constellation as a component which can be opened in Matlab, whereas a UML representation of the Simulink subsystem is available in GeneralStore, based on precise bidirectional transformation rules.

The co-simulation approach requires special attention to the synchronization aspect, but allows better support for the most recent advances in UML 2.0, the SoC profile and SysML, by relying on the latest UML CASE tools. On the other hand, specific development frameworks which ease the creation of a C++ executable model from UML and Matlab/Simulink allow faster simulation speed.

3. SysML

The Systems Modeling Language [6] is intended to unify the various modeling languages used by systems engineers. Several similarities exist between the methods applied in the area of Systems Engineering and complex embedded systems design [7]. SysML extends the application of UML to systems which are not purely software based, and can in particular be applied to design heterogeneous embedded systems. As an example, it provides support for the representation of continuous behavior and flow rates. SysML also introduces a requirement diagram to structure the requirements and link these to the system architecture and test procedures.

However, it is essential to remember that a sound development process which suits the peculiarities of embedded system design is necessary to complement the use of UML/SysML, Matlab/Simulink, and coupling tools.

4. References

[1] Electronics Weekly & Celoxica, "Survey of System Design Trends 2005", http://www.celoxica.com, 2005.

[2] "ESL Now survey", http://www.esl-now.com, 2005.

[3] Deutsche Bank F.I.T.T., "EDA Survey Results", 2005.

[4] J. Hooman et al., "Coupling Simulink and UML Models", Proc. Symposium FORMS/FORMATS, 2004.

[5] C. Reichmann et al., "Model Level Coupling of Heterogeneous Embedded Systems", Proc. 2nd RTAS Workshop on Model-Driven Embedded Systems, 2004.

[6] OMG Systems Engineering Domain Special Interest Group, http://syseng.omg.org/SysML.htm.

[7] Y. Vanderperren, W. Dehaene, "SysML and Systems Engineering Applied to UML-Based SoC Design", Proc. 2nd UML for SoC Design Workshop, 42nd DAC, 2005.