DSL4CS: Co-simulation Language for Heterogeneous CPS based on Gemoc*

Abstract—Domain specific language is ... co-simulation: We propose a co-simulation language, introduce the metamodel, concrete syntax and semantics. We implement it based on Gemoc framework.

Index Terms—co-simulation,DSL, metamodel, Gemoc, CPS

I. Introduction

Cyber-physical systems (CPSs) are integration of computation with physical processes whose behavior is defined by both computational and physical parts of the system [1]. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. The heterogeneity is one of the main characteristics of CPSs. The components of CPSs are of various types, requiring interfacing and interoperability across multiple platforms and different models of computation. Verifying heterogeneous CPSs requires the use of heterogeneous simulation environments. One emerging industry standard is the Functional Mock-up Interface (FMI) [2] [3]. It is a standard to support simulation of complex systems composed of heterogeneous components, by coupling the different models with their own solver in a co-simulation environment.

The FMI standard was first developed in the MODELISAR project started in 2008 and supported by a large number of software companies and research centers [4]. FMI offers the means for model based development of systems and is particularly appropriate way to develop complex CPSs.

II. Co-simulation Language for Heterogeneous CPS (DSL4CPS)

The design of complex systems often relies on several Domain Specific Modeling Languages (DSMLs) that may pertain to different theoretical domains with different expected expressiveness and properties. As a result, several models conforming to different DSMLs are developed and the specification of the overall system becomes heterogeneous.

Meta-models, as defined here, provide: A language representation with strict notation and grammar. This is used to represent and store the meta-model and to share meta-model information between different tools and people. Model abstraction, where each external model is represented individually in a uniform and simulation tool independent way. A generic and uniform way to connect various simulation tools. Platform independent models. No operating system, network, or other co-simulation platform dependent parameters are stored in the meta-model. A single meta-model simulator application can start the simulation tools and control the co-simulation based on the meta-model.

A. Meta Model for DSL4CPS

B. Concrete Syntax for DSL4CPS

rules, director, coordinate pattern

C. Semantics for DSL4CPS

extend Gemoc to support DSL4CS

III. Extend Gemoc to Support DSL4CPS

A. Design Meta Model Using EMF

B. Generate Concrete Syntax Based on Xtext

C. Define Semantics with CCSL

IV. Implementation and Case Study

A. Tool Implementation based on Gemoc

B. Case Study

V. Related Works

VI. Conclusions and Acknowledgement

References

References

[1] S. Zanero, "Cyber-physical systems," IEEE Computer, vol. 50, no. 4, pp. 14–16, 2017. [Online]. Available: https://doi.org/10.1109/MC.2017.105

[2] T. Blochwitz, "The functional mockup interface for tool independent exchange of simulation models," no. 2011-03-22, pp. 105–114, 2011.

[3] D. Broman, C. X. Brooks, L. Greenberg, E. A. Lee, M. Masin, S. Tripakis, and M. Wetter, "Determinate composition of fmus for co-simulation," in Proceedings of the International Conference on Embedded Software, EMSOFT 2013, Montreal, QC, Canada, September 29 - Oct. 4, 2013, 2013, pp. 2:1–2:12. [Online]. Available: http://dx.doi.org/10.1109/EMSOFT.2013.6658580

[4] C. Clauß, "Modelisar: From system modeling to s/w running on the vehicle," Co-Simulation.

Identify applicable funding agency here. If none, delete this.