Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

CO-SIMULATION USING SPECIFICATION AND DESCRIPTION LANGUAGE

Pau Fonseca i Casas

Universitat Politècnica de Catalunya
Jordi Girona 31
Campus Nord.
Barcelona, 08034, SPAIN

ABSTRACT

When faced with complex problems we need powerful tools. These tools often provide answers to partial questions satisfactorily, but sometimes is required to integrate other tools and knowledge to obtain a complete answer. Many times different actors are involved in these projects, which complicates the definition of the model. In these cases the use of a formal language to define the structure and behavior of the model is important. However often the teams involved in the project are not used with the formal language, or some models are formalized in other formal language. In this article we present an infrastructure based on a formal language (SDL) that allows using in a single simulation model different simulators (Co-simulation) , in order to simplify the interaction and participation in a project of multidisciplinary teams.

1 INTRODUCTION

The increasing complexity of the systems that are to be analyzed using simulation techniques requires that the tools, not just becomes more powerful, but can better express the relationships between the various components comprising the model. This presents two problems. The first related to how to express the relationships between the different elements of the model. A second related to how can we use and reuse existing simulation models that answer many times, comprehensively but partially, specific aspects of particular systems.

In this framework we propose to use Specification and Description Language to represent not only the simulation models in a no ambiguous way, but also establish a mechanism to reuse the existing simulation models with the technique of Co-simulation. The ultimate goal is that the system will be able to reuse simulation models defined with different formal languages and if that is not possible, usually because the model lacks of a formal representation, use this models in a Co-simulation environment.

To allow the combination of different models in a single simulation model on (Vangheluwe, 2000) are proposed three main mechanisms:

1. *meta-formalism*: This approach is based on the idea that a formalism subsumes the different formalisms of the sub-models that compose the system representation.
2. *Common formalism*: This approach is based on the transformation of the different sub models to a common formalism.
3. *Co-simulation*: Where different simulators are working together. In this case the representation of the different sub models does not matter, since we are analyzing the inputs and the outputs of each sub-model.

Common formalism technique implies the definition of several transformation algorithms between the different formal languages we want to use.

2 SDLPS

SDLPS is a simulator that have has a main goal to provide an infrastructure to execute discrete simulation models defined using SDL or DEVS. As we said previously, SDL have a textual representation. Although SDL-PR can be used to represent our model some technical reasons lead us to use a new XML representation of SDL language.

1. We need to describe information that not belong to the model. As an example in a diagram we need to relate an spatial position to each one of the different SDL blocks in order to allow a representation of the model. With XML is easy to add an optional tag that can carry this information.
2. XML have a lot of different libraries that allows a good manipulation of the information. This simplifies the codification and the maintenance of the parser, and the code related.
3. XML can be validated, using an XSD schema. This is useful in order to detect some structural problems when we write the model before to run it on the simulator.
4. XML can be easily transformed to other representations. Some programs allows to implement this transformations between XML files in a graphical way (Stylus Studio ® as an example).
5. XML representation of the model helps us in the representation of it in a web site.
6. In our XML representation of the SDL language the code related to the tasks can be C code. Also, the code related to the procedures can be a diagram or, like in the task blocks, C code. This simplifies the implementation of the DLL in the compiler.

Since for SDL and DEVS no exists an specific representation using XML it is needed to establish a mechanism for, at least for SDL, transform from SDL-PR to SDL-XML (to use models defined in SDL on other tools).

3 CONCLUDING REMARKS

When we deal with a complex system, where different personnel is involved in its definition, and where we do not have a complete picture of the system, formal languages are the best tools in order to understand the model. This enables the operational validation, simplifying the detection of errors and accelerating the implementation. Problems arises when we want to connect different legacy models or specific simulation tools to the model. In that case it is needed to define a Co-simulation methodology that possibilities this. Specification and Description Language is a graphical language that can be understood easily by all the members of the team. Thanks to its modular structure not all the members must understand the whole system, simplifying its validation.

We present an infrastructure named SDLPS that implements *Co-simulation* and *Common Formalism* techniques. Some projects where initial versions of this methodology have been used can be reviewed on (Fonseca i Casas, et al., 2011). On http://sdlps.upc.edu you can find more information and the download page.

4 REFERENCES

Fonseca i Casas, P., Casanovas, J., Montero, J. & Guasch., A., 2011. Experiences of Simulation Use in Industrial Projects. *SCS M&S Magazine,* July.Volume 3.

Vangheluwe, H. L. M., 2000. *DEVS as a common denominator for multi-formalism hybrid systems modelling.* s.l., IEEE Computer Society Press, pp. 129--134.