Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

ENHANCING SDLPS WITH CO-SIMULATION

Pau Fonseca i Casas

Universitat Politècnica de Catalunya
Jordi Girona 1-3
Campus Nord.
Barcelona, 08034, SPAIN

ABSTRACT

The increasing complexity of the systems that can be analyzed using simulation techniques requires that the tools, not only become 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. An a second related to how can we use and reuse existing simulation models that answer many times, comprehensively but partially, specific aspects of certain systems In this paper we present a methodology based on Specification and Description Language, a formal and graphical language, that allows using in a single simulation model different simulators (co-simulation). This simplifies the interaction and the participation in a project of multidisciplinary teams. In addition, we cite a tool that implements this methodology.

1 INTRODUCTION

We propose to use Specification and Description Language (SDL) (Telecommunication standardization sector of ITU, 1999) 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, (i) meta-formalism, (ii) Common formalism, and (iii) co-simulation.

SDLPS (SDL Parallel Simulator, sdlps.upc.edu) (Fonseca i Casas, 2008) allows to use Specification and Description Language or DEVS to represent a simulation model. Specifically, SDLPS understand a XML representation of a SDL model and a XML representation of a DEVS model (Fonseca i Casas, 2009) (Fonseca i Casas & Casanovas, 2011). Common formalism techniques however are not enough always. If you have a legacy simulation model, that do not have a formal representation, or the simulation is something like a black box (very common in specialized simulation software), co-simulation is the only technique that can be used to combine different models in a single simulation model.

2 CO-SIMULATION ON SDLPS

Each one of the different agents in SDL defines, due to the structure of the language, if they can be executed in parallel or not. In order to define a set of SDL agents that must be executed sequentially it is only needed to define a set of PROCESS inside of a PROCESS agent. This clearly indicates that all these agents must wait for the completion of the execution of the other agents.

With this in mind the modeler can clearly express in a graphical way the execution mode (parallel or sequential) for each one of the different simulation sub-models (a SDL agent). This is very useful in order to define a distributed simulation model or to combine an existing simulation model with other simulation components that can actuate in real time.

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

2.1 Time management

On SDL standard exist a global clock that represents logically the time for the simulation (viewing the simulation as a whole without different components). In SDLPS this time is implemented and interpreted similar to the Lower Bound Time Stamp (LBTS in HLA1.3). This is the time that all the federates can advance securely without the need of implement a rollback mechanism. The value of this clock is stored in the SYSTEM agent. This value exists in all the other agents that compose the model. For all the other agents and components this time represents its LVT. From the point of view of the definition of the model it is not needed to understand the internal time management. This management differs depending on the method we use to perform the synchronization between all the elements that compose the system, conservative or optimistic. In our current implementation SDLPS only supports conservative synchronization.

2.2 Input and output variables

In order to define the communication mechanism between the different elements that compose the system SDL have two mechanisms depending on the nature of the sub-model we want to incorporate to the model. If the sub-model is defined following SDL or DEVS languages we have a complete definition of its behavior, hence we have a PROCESS. In that case we can use parameters that can be attached to the SIGNALS. When a PROCESS receives a new SIGNAL, if is defined in the model, its attached parameters can be used inside the PROCESS methods. Only is needed to specify what are the parameters, and the type of the parameters that the different PROCESS expect to receive. However often we do not have a formal representation of the model using SDL or DEVS. In that case it is needed to use a PROCEDURE CALL that enables the possibility to connect the model with other components. This is a very powerful approach that fully possibilities the co-simulation mechanism.

In the SDLPS implementation of the SDL PROCEDURE CALL is allowed to call other computer programs (or simulators) to obtain data and combine its information dynamically with the simulation model. The key element here is that the procedures always belongs to a PROCESS that defines its integration and interaction with the whole simulation model, defining as we said previously on the SIGNALS the parameters that we are using in the sub-models. Three different mechanisms to define the PROCEDURE are implemented on SDLPS, *full definition*, *API implementation* and *remote implementation*.

3 CONCLUDING REMARKS

Specification and Description Language is a graphical language easy to be understood by all the members of the team. Thanks to its modular structure not all the members needs to understand the complete system, simplifying its validation. SDL can be used to represent parallelism and synchronization in the model. This enables to use co-simulation technique. SDLPS is a simulation software that implements this approach. On http://sdlps.upc.edu you can find more information and the download page.

4 REFERENCES

Fonseca i Casas, P., 2008. *SDL distributed simulator.* Miami, INFORMS.

Fonseca i Casas, P., 2009. *Towards an automatic transformation from a DEVS to a SDL specification.* Istanbul, Turkey, s.n.

Fonseca i Casas, P. & Casanovas, J., 2011. *Towards a SDL-DEVS Simulator.* Barcelona, s.n.

Telecommunication standardization sector of ITU, 1999. *Specification and Description Language (SDL).* [Online] Available at: http://www.itu.int/ITU-T/studygroups/com17/languages/index.html [Accessed April 2008].

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.