Business Process Simulation: Transformation of BPMN 2.0 to Discrete Event System Specification

[1]Jayesh Gamar, [2]Dr.B.S.Agrawal.

Jayesh J. Gamar
Supervisor Instructor
ITI Pavi-Jetpur, Dist: Chhotaudepur.
jayesh_gamar@yahoo.co.in
ORCID-0000-0002-3921-9953

Dr. B.S. Agrawal
Director
Rollwala Computer Centre,
Gujarat University, Navarangpura, Ahmedabad.
bijendra.agrawal@gmail.com

ABSTRACT: Theoretical modeling is a complicated characteristic of a simulation study that straight affects the quality and effectiveness of simulation projects. This paper presents a model to model transformation from a conceptual modeling language to a simulation model specification. BPMN (Business Process Model and Notation) is worn for theoretical modeling and DEVS (Discrete Event System Specification) is elected for simulation model requirement. Simulation is a dynamic feature of MDSE and which explains the need of coherent M&S formalisms for simulation activities.Accordingly, this paper presents the simulation of service systems based on DEVS models. It defines a transformation approach of BPMN models into DEVS simulation models based on the metamodel approach, and describes the enrichment of obtained DEVS models through performance indicators (time and costs).

Key words: *BPMN, DEVS, Model transformation, Business Process simulation, MDD4MS*

*****

1. INTRODUCTION

To remain competitive, a company must differentiate itself from other competitors. Since improving the product'sperformance can reach some limits,one open solution is toimprove the enterprise service system, redefine its business processes and share more information (considered as additional services) with customers and suppliers.

Despite the fact that conceptual modeling is an important process in a simulation study, there are many simulation projects that have no explicit conceptual model, a poorly or only partially developed conceptual model, or incomplete documentation of the simulation conceptual model. However, appropriate expansion of a theoretical model is grave for expressing the situation, elements, relationships, boundaries and purpose of the simulation study [1].

In the frame of Model Driven Service Engineering Architecture (MDSEA) [Bazoun et al. 2014], a distinction can be made between static and dynamic service system modeling [Cardoso et al. 2012].A business process is a series of activities that produces a product or service for a customer. Business Process Modeling (BPM)[Cardosoet al. 2012] results in a representation of an organization's business processes to be analyzed and improved [Weske 2007].Business process's models provide a suitable static view, but frequently missing thetemporal dimensionto expressoutput performance such as an expected cost or a desired duration. In detail, the impact of correct or incorrect behaviorof complex models over time is not clearly visible using static view. This issue can be solved by running a business processsimulation for analyzing and understanding the business process model according to its dynamic.

In this research, we suggest the use of Model Driven Development (MDD) approaches in order to benefit from the conceptual models in the further steps of the simulation study through automated model transformations [2]. The model driven technologies are introduced to the simulation field in the last decade. To the best of our knowledge, Vangheluwe and de Lara [3] introduced the metamodeling and model transformation research into DEVS based modeling and simulation in 2002. Since then, the simulation field has made significant progress in automating the code generation from simulation model specification [4, 5].

2 Literature Review

2.1. Business Process Modeling

Business Process Modeling (BPM) is the activity of defining a graphical representation of either the current or the future processes of an organization [9]. BPM is generally performed by business analysts in order to analyze and improve process efficiency and quality. A business process model is a visual representation of the sequential flow and control logic of a set of related activities or actions.
Different modeling techniques have been used to develop business process models [10]. UML, IDEF and BPMN are the most common business process modeling techniques [11]. In this work, BPMN is selected since it is an industry-wide standard for creating business process models and has

ISSN: 2321-8169
183 – 186

recently been updated. BPMN follows the tradition of flowcharting notations for readability and flexibility. In addition, the BPMN execution semantics is fully formalized. There are five basic categories of elements in BPMN [12]. These are: flow objects, data elements, connecting objects, swimlanes and artifacts.

2.2. Discrete Event Simulation

Discrete event simulation is an effective tool for analyzing and designing complex systems. In a discrete event system, state of the system changes at discrete points in time [13]. DEVS is a well known mathematical formalism based on system theoretic principles. Any system with discrete event behavior can be represented with the DEVS formalism and an equivalent DEVS representation can be found for other formalisms [14].

Classic DEVS specification defines the structure for the basic DEVS formalism. Models that are expressed in the basic formalism are called as *Atomic* models. An atomic DEVS model is defined with the following information: the set of input values, the set of output values, the set of states, the internal transition function, the external transition function, the output function and the time advance function.

3. Transformation from BPMN to DEVS

In the context of BPMN to DEVS transformation, authors in [Cetinkaya et al. 2012]and [Mittal et al. 2012] presented a Model Driven Development (MDD) framework for modeling and simulation (MDD4MS). In the frame of this framework they defined a model to model transformation from BPMN as a conceptual modeling languageto DEVS as a simulation model specification.BPMN and DEVS Meta-models were presented. In addition, a set of transformation rules were defined in order to transform BPMN models into DEVS models. According to these rules, some BPMN concepts (Pool, Lane, SubProcess) were mapped to DEVS coupled component, while Task, Event (Start, End, and Intermediate), and Gateway were mapped to DEVS atomic component.

Comparing the BPMN metamodel defined with the latest version of BPMN 2.0 metamodel[OMG 2011] we can conclude that several concepts are missing and thus were not transformed into their corresponding DEVS concept. Authors didn’t mention the different types ofBPMNTasks (UserTask, ManualTask, ServiceTask…) and BPMN Intermediate Events (Message, Signal…) that can be mapped differently when transformed into DEVS concepts. The distinction would be in the figure of states forming every DEVS infinitesimal Model.Based on these remarks, the work presented in this paper takes into consideration these points in an attempt to benefit from previous work and propose new mapping and transformation rules.

DEVS Simulators

Electing a target DEVS tool for model transformation requires a literature review of current DEVS Simulation tools. The DEVS group standardization maintains on its websitethe updated list of most used DEVS tools known by the DEVS community [Wainer 2013]. In [Hamri and Zacharewicz 2012], the authors have given a brief description and comparison of popular tools.

ADEVS was the first DEVS tool developed in C++ by the Arizona University. It consists in an ad-hoc simulator. DEVS abstractclasses should be extended by users to define atomic and coupled models, and then the simulation canbe launched. The drawback resides in the fact that users needprogramming skills to code the models.

DEVSJAVA is a Java framework in which the kernel simulator is ADEVS. It ropes also modeling and simulation of DEVS with variable structures. However, at atomic level, the user should put into practice the equivalent DEVS actions in Java (in our opinion the user has not enough skills toprogram his atomic models).

CD++Builder is a DEVS modeling and simulation environment that integrates interesting features and facilities for the user. It allows modeling and simulation of other DEVS formalisms (cell-DEVS, Quantized-DEVS, etc). It provides a DEVS graphical editor to model coupled and atomic models, and to encapsulate them through components for further reuse.

4. MODEL TRANSFORMATION FROM BPMN 2.0 TO DEVS MODELS

This section introduces the main transformation principles from BPMN model to DEVS model, including the transformation architecture, DEVS metamodel, the mapping of BPMN concepts to DEVS concepts, and the implementation using a transformation language.

Transformation Architecture

The metamodel approach [OMG 2003] is one of the most used transformation techniques. Figure 1 presents the metamodel approach adapted to the context of model transformation from BPMN2.0 model to DEVS model.Three different levels are identified: model, metamodel, and meta-metamodel. The BPMN model is the source model to be transformed, while the DEVS model is the target model resulting from the ATL transformation. BPMN and DEVS models conform to the BPMN 2.0 and DEVS metamodels respectively.

ISSN: 2321-8169
183 – 186

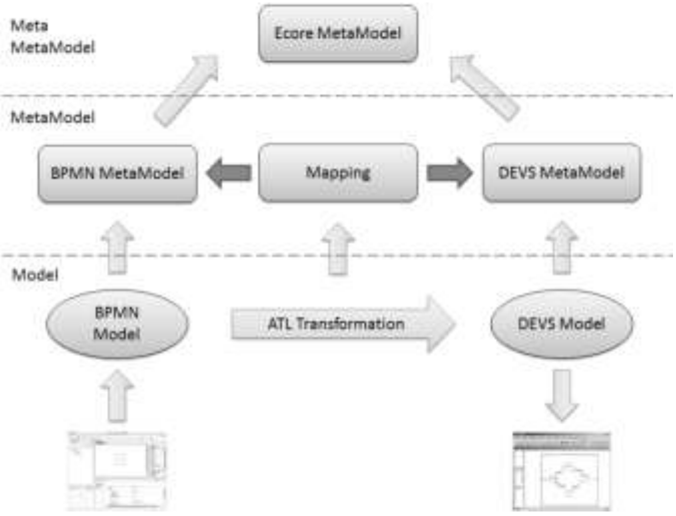

Figure 1.Transformation architecture

BPMN and DEVS MetaModels

Source and target metamodels should be well identified to proceed with the transformation (Figure 1). BPMN 2.0 metamodel specified in [OMG 2011] is the source metamodel. There is no endorsed metamodel for the target DEVS metamodel, but several researches were held for the purpose of building a DEVS metamodel but a synthesis work is proposed in [Garredu et al. 2012]. The transformation from BPMN to DEVS models has required gathering previous works for setting a DEVS metamodel, as a result the authors proposed a simplified DEVS metamodel. It is used as a target metamodel which conforms to the DEVS specification [Zeigler et al. 2000].**Error! Reference source not found.** It is used as a target metamodel which conforms to the DEVS presents the DEVS metamodel defined in Eclipse Ecore format.

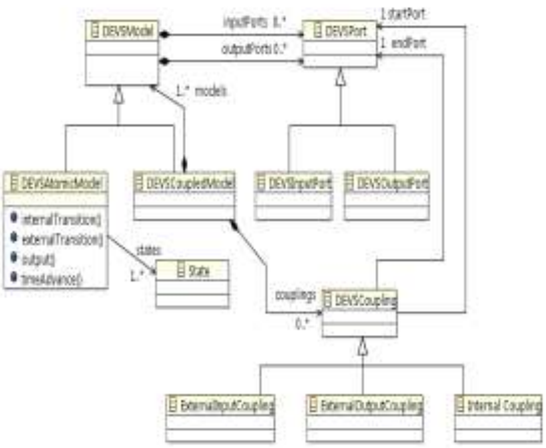

Figure 2.Simplified DEVS metamodel

5 Conclusions

We presented a procedural DEVS metamodel, a BPMN metamodel and a model transformation from BPMN to DEVS. Until now, the structure of the generated models, efficiency of the transformation and compatibility with DEVS look very promising. However, the repositioning of the modeling elements is not as we desired.

During our prototyping process, the model to code transformation from DEVS models is defined as well. We generate Java code for both coupled and atomic models. In other words, we generate executable simulation models from BPMN and DEVS models. Now, we are working on the improvement of the model transformations and the efficient integration of all steps.

Business process modeling and simulation in the frame of the Model Driven Service Architecture (MDSEA) project. As a result, it presented a transformation of BPMN models into DEVS models based on previous researches done in this domain. The approachhas now proposed anexhaustive mapping, based on existing works plus additional concept mapping from BPMN concepts to DEVS concepts.

REFERENCES

[1] Business Process Simulation: Transformation of BPMN 2.0 to DEVS Models

[2] Model Transformation from BPMN to DEVS in the MDD4MS Framework Deniz Cetinkaya ,Alexander Verbraeck , Mamadou D. Seck

[3] [ATL 2013]"ATL/User Guide – The ATL Language" http://wiki.eclipse.org/ATL/(accessed 10 November 2013).

[4] [Bazoun 2013]: Bazoun, H., Zacharewicz. G., Ducq. Y., Boye, H. "Transformation of Extended Actigram Star to BPMN 2.0 and Simulation Model in the frame of Model Driven Service Engineering Architecture". TMS, (2013).

[5] [Bazoun 2014]: Bazoun, H., Zacharewicz. G., Ducq. Y., Boye, H. "SLMToolBox: An implementation of MDSEA for Servitisation and Enterprise Interoperability". Paper accepted in I-ESA (2014) 7th international conference.

[6] [Boye 2014]: boyé, H., Bazoun, H., Belkhelladi, K. "SLMToolBox: ATool Set For Service Engineering". Paper accepted in MODELSWARD 2014 2nd international conf on Model-Driven Engineering and Software Development

[7] [Cardoso 2012]: Cardoso, J., Pedrinaci, C., Leidig, T., Rupino, P., De Leenheer P. "Open semantic service net-works." Paper presented at: The international Symposium on Service Science (ISSS); (2012).

[8] [FP7 2011]: FP7 – FoF-ICT-2011.7.3 – "Manufacturing SErvice Ecosystem Project- Annex 1 description of work" – July 29th 2011. http://interop-vlab.eu/

[9] [Garredu 2012]: Garredu, S., Vittori, E., Santucci, J-F., Bisgambiglia, P-A."A Meta-Model for DEVS Designed following Model Driven Engineering specifications." SIMULTECH,page 152-157.SciTePress, (2012).

[10] [Hardis 2013]: Hardis is a software company with specialist expertise in management computing http://www.hardis.fr/ eng/ jsp/site/Portal.jsp(accessed 18 October 2013).

[11] [Hamri 2012]: Hamri, M. and Zacharewicz, G. “Automatic generation of object-oriented code from DEVS graphical specifications.”. In WSC'12. Article 409 , 12 pages, 2012.

[12] [Cetinkaya 2012]: Çetinkaya, D., Verbraeck, A., Seck, M.D. “Model Transformation from BPMN to DEVS in the MDD4MS Framework”, TMS-DEVS, (2012): 304-309

[13] [Mittal 2012]: Mittal, S., and Risco Martin, J.L. Netcentric System of Systems Engineering with DEVS Unified Process. 610-613, 2012. CRC Press.

[14] [McNeill 2010] Ken McNeill “How to extend the Eclipse Ecore metamodel.”http://www.ibm.com/developerworks/library/os-eclipse-emfmetamodel/index.html

[15] [OMG 2011]: OMG, “Business Process Model and Notation (BPMN) version 2.0” document num: formal/2011-01-03.

[16] [OMG 2003] : OMG, “MDA Guide Version 1.0.” document number: omg/2003-05-01.

[17] [Thoben 2001]: Thoben, K.-D., Jagdev, H., Eschenbcher, J. “Extended Products: evolving traditional product concepts” In the 7th International Conference on Concurrent Enterprising: Bremen, Germany, June 2001.

[18] [Wainer 2013]: DEVS TOOLS, hosted by G. Wainer at Carlton University, November 2013, http://www.sce.carleton.ca/faculty/wainer/standard/tools.htm

[19] Robinson, S. 2006. “Conceptual Modeling for Simulation: Issues and Research Requirements”, In Proceedings of the WSC’06, pp.792-800.

[20] Cetinkaya, D.; A. Verbraeck; M.D. Seck. 2011. “MDD4MS: A Model Driven Development Framework for Modeling and Simulation”. In Proceedings of the 2011 Summer Computer Simulation Conference, Den Haag, The Netherlands.

[21] Vangheluwe, H.; J. de Lara. 2002. “Meta-models are Models Too”, In Proceedings of the WSC’02, 597-605.

[22] Guiffard, E.; D. Kadi; J.-P. Mochet; R. Mauget. 2006. “CAPSULE: Application of the MDA Methodology to the Simulation Domain”. In Proceedings of the EURO SIW’06.

[23] Risco-Martín, J.L.; J.M. de la Cruz; S. Mittal; B.P. Zeigler. 2009. “eUDEVS: Executable UML with DEVS Theory of Modeling and Simulation”, Simulation, 85: 750-777.

[24] Cetinkaya, D.; A. Verbraeck. 2011. “Metamodeling and Model Transformations in Modeling and Simulation (invited paper)”, In Proceedings of the WSC’11, AZ, USA.

[25] Wainer, G.A.; K. Al-Zoubi; O. Dalle; R. Hill; S. Mittal; Risco-Martín, J.L.; H. Sarjoughian; L. Touraille; M.K. Traoré; B.P. Zeigler. 2010. “Standardizing DEVS Model Representation”, Chapter 17 in Discrete-Event Modeling and Simulation: Theory and Applications. G.A. Wainer; P. Mosterman, Taylor and Francis.

[26] Traoré, M. K. 2009. “A Graphical Notation for DEVS”. In Proceedings of the SpringSim’09. SCS, CA, USA.

[27] Schedlbauer, M. 2010. The Art of Business Process Modeling: The Business Analyst's Guide to Process Modeling with UML & BPMN. Cathris Group.

[28] Recker, J.; M. Rosemann; M. Indulska; P. Green. 2009. “Business Process Modeling- A Comparative Analysis”. Journal of the Association for Inf. Systems, 10:4, 333-363.

[29] List, B.; B. Korherr. 2006. “An Evaluation of Conceptual Business Process Modelling Languages”. In Proceedings of the 2006 ACM Symposium on Applied Computing (SAC '06). ACM, NY, USA, pp. 1532-1539.

[30] BPMN. 2011. “Business Process Model and Notation (BPMN) Version 2.0”. Object Management Group, Inc.

[31] Fishman, G. S. 2001. Discrete-Event Simulation: Modeling, Programming, and Analysis. Springer.

[32] Zeigler, B.; H. Praehofer; T. Kim. 2000. Theory of Modeling and Simulation. Academic Press.

[33] Stahl, T.; M. Völter; J. Bettin; B. von Stockfleth. 2006. Model-Driven Software Development: Technology, Engineering, Management. John Wiley.

[34] MDA. 2000. “Model Driven Architecture”. Object Management Group, Inc.

[35] ATL. 2005. “Specification of the ATL (Atlas Transformation Language) Virtual Machine Version 0.1”. ATLAS INRIA and LINA Research Group.