Proceedings of the 21st International Conference on Modeling & AppliedSimulation (MAS), 021
19th International Multidisciplinary Modeling & Simulation Multiconference

2724-0037 © 2022 The Authors.
doi: doi.org/10.46354/i3m.2022.mas.021

Business Process Web-based Platform for Multi modeling and Distributed Simulation

Mariane El Kassis[1,*], François Trousset[1], Nicolas Daclin[1] and Gregory Zacharewicz[1]

[1]IMT-Mines Ales, LSR, 6 Avenue de Clavières, 30100 Alès, France

[*]Corresponding author. Email address: mariane.el-kassis@mines-ales.fr

Abstract

Businesses and organizations constantly strive to improve their business processes (BPs). One of the critical components of creating effective BPs is the stakeholders' participation in BP management and improvement initiatives through modeling and simulation methodologies, which aim to minimize conflicts, promote innovation, boost ownership, and promote inclusive decision-making. To bridge the gap between the worlds of BP modeling and simulation, this study highlights the current development of concepts, methodologies, and tools. This article will provide an overview of recent research and suggest a method for extending Business Process Modeling and Notation (BPMN) models with Business Process Simulation Interchange Standard (BPSIM) and converting them into simulation models for Discrete Event System Specification (DEVS) based on a metamodel. Additionally, an architecture is proposed to tackle the interoperability need based on co-simulation and distributed simulations.

Keywords: Business Process Modeling, Model transformation, Distributed Simulation, BPMN, BPSIM, DEVS, FMI, HLA

1. Introduction

Business processes play a vital role in the organization's operation and are widely implemented and used in many businesses. It is utilized on three different levels: the descriptive level, the executable level, and the analysis level. The descriptive level is used to share and capitalize the knowledge, the executable level, to orchestrate and automate business operations, and the analytical level, to improve performance. For this reason, knowledge of the BPMN language is becoming in demand in the corporate sector (Pasha and Pasha, 2013). BPMN workflows are typically created using independent applications by business users or developers, enabling stakeholders to translate their knowledge into conceptual models apart from development, implementation, and environments. On the other hand, business process simulation is crucial to improving an organization's performance. Most business process improvement projects still do not rely on simulation despite being an essential business process management component. The need for technical abilities, typically not acquired by business stakeholders, may be one of the causes. Modeling and Simulation (M&S) techniques advocate using conceptual models for creating simulation models, which necessitates involving business stakeholders in discrete event simulation improvement initiatives. We anticipate extending BPMN workflows by using discrete-event simulation components from BPSIM. The extended model will then be converted into a DEVS model to run the simulation. The traditional simulation approach is no longer sufficient for complex models requiring resource-intensive processing and interaction with multiple systems. Hence, the need to define transformation rules for splitting up the simulation model into multiple sub-models that can be executed and distributed across many processors or computers, based on the most suitable modeling tools, in an integrated manner.

The paper is structured as follows: after a brief introduction in section 1, state of the art is presented in section 2. Methodology and application are detailed in sections 3 and 4. Section 5 concludes the paper and opens outlines.

2. State of the art

2.1. Background

Business processes need to be standardized to be better understood and maintained (Mens and Gorp, 2006). Several standards have been created for this specific objective. This paper studies BPMN, an OMG standard that offers a conventional notation for cooperation between business users, analysts, and implementers. With over 20 years of use and development, BPMN is currently mature and widely used to express business processes (Pasha and Pasha, 2013). Several business process management tools that combine graphical notation and software provide the ability to comprehend process models and orchestrate them (Pufahl et al., 2018). The simulation capabilities of these technologies are still very constrained. Business stakeholders can create business processes using a few available free and open-source BPMN applications. These tools might enable model simulation; however, integrating the modeling and simulation aspects is still insufficient. For instance, the model to be simulated needs to be extended to include external simulation components (scenario parameters, resources, duration, etc.). Due to the continued separation of these two worlds, only people familiar with simulation concepts can use simulation. However, BPSIM is a standardized specification created to simulate business models by enhancing the conceptual model with scenario factors that impact the way processes are conducted. BPSIM strives to improve specific BPMN elements by providing details on relevant parameters and logic important to the simulation model. Hence the need for a BPMN+BPSIM editor that allows business users to create a business process model, to enrich it with simulation parameters and accordingly simulate it to get results of interest. This study aims to propose a method and a tool for multi-modeling and multi-simulations:

- Multi-Modeling by introducing a transformation from a BPMN+BPSIM model into a DEVS simulation model/ sub-models based on the Model-Driven Engineering (MDE) approach
- Multi-Simulation by allowing co-simulation and distributed simulations through Functional Mock-up Interface (FMI) and High-Level Architecture (HLA)

We, therefore, plan to develop a web-based platform that will empower business stakeholders and involve them in discrete event simulations and business process improvement initiatives. The proposed web platform is based on standards allowing multi-modeling and multi-simulation of business processes and bringing interoperability to several simulation components. This study examines the transformation and simulation of a BPMN model that has been extended using BPSIM into simulation models/sub-models that complies with the DEVS formalism to verify and simulate model properties. The simulation model is partitioned and executed in a co-simulation/distributed approach.

2.2. BPMN Extensions

Business process modeling notation (BPMN) is a method used to graphically express the value chains and business operations of a company (ISO/IEC 195103) (OMG, 2013). The model extension approach enriches the model with external notions originating from various domains in order to cater to specific purposes. According to OMG, extensibility is one of the key characteristics of the BPMN metamodel. Resources and key performance indicators are examples of features not included in BPMN that are visualized using BPMN extensions (Onggo and al., 2017). Works concentrated on resource definition (Stroppi et al., 2015) and resource allocation constraints (Awad et al., 2009), considering the different types of resources. Additionally, other works focused on extensions related to performance measurement (Friedenstab et al., 2012) and risk measurement (Marcinkowski and Kuciapski, 2012). In the context of discrete event simulation, resource allocation processes and resource failure are also modeled as BPMN extensions. The two types of BPMN extensions are conceptual enrichment and operational enrichment. Adding attributes like resource description, type, and capacity expands the concept of resources. The description of the enriched element's behavior constitutes operational enrichment. For instance, tasks were extended in (Mallek et al., 2010) to describe the interoperability behavior. The enriched model is still considered BPMN-Compliant even if extension elements and new characteristics are added to already-existing BPMN elements (OMG, 2013). BPMN extensions may be classified into three types:

- Descriptive: if an extension's primary goal is to describe a domain.
- Analytical: if the primary goal is to facilitate some sort of analysis of current BPMN models.
- Execution: If the extension is designed to assist with process execution

2.3. BPMN transformation and simulation

MDE is a strategy of software engineering that encourages the creation of models at multiple degrees of abstraction, moving the level of development from programs to models. Model use and model transformation are two of the fundamental tenets of MDE. While the model transformation principle encompasses the ideas of mappings and transformation rules, the model principle focuses on the concepts of meta-model and conformance (Mens and Gorp, 2006). Many scholars spent the last two decades focusing on the transformation of high-level operations into simulation models. The source and target models in

MDE must adhere to their respective metamodels. While the simulation model DEVS lacks a specific standardized metamodel, BPMN complies with the BPMN metamodel standardized by OMG (OMG, 2013). To provide a model-driven development framework for modeling and simulation (M&S), researchers presented the BPMN and DEVS metamodels along with a defined set of rules for transforming a conceptual source model (BPMN) into a simulation model (DEVS). BPMN elements differ from each other in terms of their internal behavior. Therefore, they may be considered black boxes, moreover, DEVS' behavior is transparent and considered a white box. A BPMN flow's components must be mapped to the appropriate DEVS representation before they can be simulated in DEVS. A mapping for several BPMN elements already exists. Nevertheless, only a small number of elements were studied and considered. Every study team created its own DEVS metamodel since there is currently no standard metamodel. The first mapping notions were introduced in (Cetinkaya et al., 2011) and they were later expanded to cover more areas, such as additional categories of BPMN tasks (Bazoun and al., 2014) and resource allocation and failure (D'Ambrogio and Zacharewicz, 2016). However, only a small number of BPMN principles are included in these efforts. They still lack several crucial components, such as message flows, interrupting events, and intermediate events. Numerous academics examined the conversion to alternative discrete event simulation models, like Petri nets, in addition to the transformation of BPMN to DEVS (Mutarraf and al., 2018). The notion of time is not properly specified in Petri nets; hence DEVS simulation of models is more practical than Petri net simulation (Zeigler et al., 2011). DEVS also offers a formal specification of the simulator, along with a broader framework for modeling and simulation systems (Zacharewicz and Hamri, 2007).

2.4. Simulation applications

It is essential to include simulation settings that best reflect reality to get the most realistic simulation results possible. The present crop of BP modeling tools on the market varies in their simulation capacities, which impacts the outcomes' precision. Three categories (Pufahl et al., 2018) can be used to group simulation tools:

- Tools for managing or modeling business processes that facilitate simulation (e.g., ADONIS, ARIS Toolset, Bizagi Modeler, L-SIM, Simul8, PragmaDEV)
- Tools for general-purpose simulation (e.g., Arena, AnyLogic)
- Stand-alone business process simulation tools (e.g., Bimp)

Bizagi, L-SIM, Simul8 and PragmaDEV use the BPSIM standard. The following criteria were used to assess the simulation capabilities of several BP simulation tools: Definition of the context, time commitment, control, available resources, costs, and priorities. Most business process simulation tools that use BPMN are proprietary and/or commercial. The result validation of such application is complex as we lack information on the simulation engine used in the backend. As a result, reproducing the experience is difficult to control, compelling the need for open-source, reusable, and flexible simulators. The industry and academia developed several DEVS tools. The design objectives and particular programming language implementations of these tools vary. The performance, formalisms, compliance, functionality, and accessibility of the DEVS simulators that are currently available were compared (Yentl and Hans, 2017) :

- Parallel DEVS models can be effectively simulated using ADEVS, built in C++.
- For programmers who are experienced with C++, ADEVS is advised, and it is thought to be performant at the expense of functionality.
- CD++, which was created in C++ and is specialized in Cell DEVS models, is advised for non-programmers.
- VLE was created in C++ and stayed at a more fundamental level.
- Python PDEVS was suggested for instructional reasons and provided features useful to DEVS beginners.
- DEVS-Suite well illustrates the semantics of Parallel DEVS models in Java. It was recommended for instructional uses.

2.5. Standards for distributed simulations

From the perspective of the M&S process, distributed simulation implies working with various related subsystems that are modeled and simulated in a distributed manner. For diverse fields of expertise, there may, in fact be several M&S tools that are created and implemented in different languages. Furthermore, some of these tools must only work with specific type of hardware. Middleware and processes for interoperability enable the synchronization of these many instruments and the management of data transfers among them. The usage of distributed simulation technologies is a paradigm for modeling dynamic, diverse, and spatially distributed systems. In addition to accelerating simulations, they function as strategic technologies for connecting simulation components of various types (Fujimoto, 2015). Several approaches are used in the M&S field to address the problems of simulation model interoperability and execution in distributed computing platforms. Two of the most notable projects in these areas are Functional Mock-up Interface (FMI) and High-Level Architecture (HLA). HLA is an IEEE standard for distributed computer simulation (IEEE, 2010). A distributed simulation is known as a "federation" according to the HLA standard. Several HLA simulation entities, known as Federates, make form a Federation. These Federates can communicate using the Run-Time Infrastructure (RTI). The RTI is the cornerstone of a Federation execution and provides several services to manage communications and information sharing among Federates. The HLA standard was employed in this work to reproduce our system in a distributed environment. FMI is a European standard created in 2011 by MODELISAR to enhance the design of systems and software incorporated in automobiles. This standard was created and developed for industrial applications, particularly for cyber-physical systems, to ease data flows. One of its goals is to make it easier for industrial partners to collaborate by giving them a common mechanism to exchange models while ensuring the safety of their industrial systems. Model exchange (ME) and cosimulation (CO) are two potential uses that the standard enables with two interfaces (Blochwitz, 2016). FMI's main objective is to facilitate model reuse across numerous modeling environments and tools during the system development phases. A Functional Mock-up Unit (FMU) is a simulation component that complies with FMI; it contains a model description file, user defined libraries, source codes, model icons, and documentation.

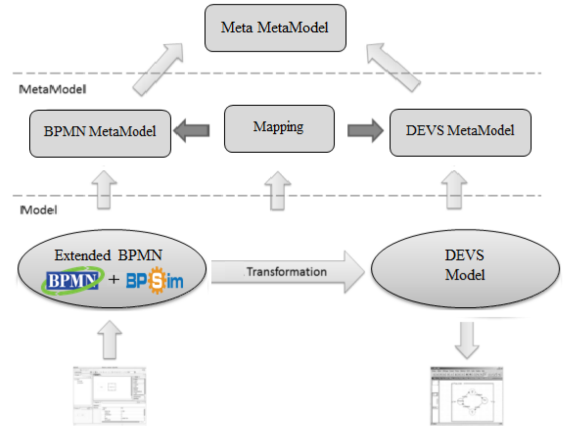


Figure 1. Extension and Transformation architecture

3. Methodology

BPMN is initially made to depict business processes without addressing simulation. With the addition of the required simulation parameters, our objective is to be able to simulate BPMN models. The Workflow Management Coalition (WFMC) has developed a standard called BPSIM that outlines how to parameterize business process models from several viewpoints to enable process simulation, analysis, and optimization (WFMC, 2016). We aim to incorporate BPSIM2.0 to the BPMN2.0 standard for the specification level. The extended BPMN (BPMN+BPSIM) model can then be simulated using a variety of BPSIM-capable simulation engines that are already available and that can read and interpret the semantics present in the extended model. At the simulation level, DEVS model is used because it takes the temporal dimension into account (Zeigler et al., 2011). Furthermore, the current BPSIM simulators are proprietary, making it challenging to customize them for certain situations. In this context, our goal is to convert BPMN expanded with BPSIM models into DEVS models that may be operated or simulated to detect problems and enhance the business process, as shown in Figure 1. Based on a specified set of mapping and transformation rules, the extended BPMN model (BPMN+BPSIM) is transformed to the corresponding DEVS model. The ultimate objective is to create a simulation tool that is standard-based, open, and extensible. Our approach is open providing users the ability to enhance the set of production rules to match needs and requirements. Using this approach, we propose a framework able to define transformation rules and produce distributed simulations in DEVS using HLA and FMI standards.

3.1. BPMN EXTENSION WITH BPSIM

The open-source BPSIM 2.0 specification document provides means to apply BPSIM semantics and properties to BPMN. One of the objectives of this study is to extend some BPMN elements with information on associated parameters and logic that is important for the simulation model generated from BPSIM, hence enriching the conceptual model. The BPSIM perspective is defined by the following five parameters (WFMC, 2016):

- Time Parameter: Describes time intervals as defined by an external observer
- Control Parameter: Identifies the resources employed by a business process element
- Cost Parameter: Determines all costs of an activity for human or non-human resources
- Property Parameter: Specifies simulation values for data instances used by the business process
- Priority Parameter: Controls the priority of the associated BP element

The BPSIM2.0 WFMC specification (WFMC, 2016) defines which BPSIM parameter may be associated with each BPMN element. BPSIM adds various capabilities to BPMN, including the following: probabilistic distribution, concurrent simulations, resource allocation for each task, probabilistic duration, probabilities for conditional results, resource priorities, calendars availability, etc. Given that BPSIM parameters will modify the behavior of the BPMN element we need to adapt the transformation rules by taking into consideration these BPSIM parameters. Examples of BPMN elements extended BPSIM parameters are shown in Figure 2 , 3 and 4.

3.2. Transformation from (BPMN + BPSIM) to DEVS

Once the enrichment of BPMN with BPSIM is realized, the resulted BPMN+BPSIM model will then be transformed into DEVS model. This transformation requires a metamodel for both the source and target as well as a language to define transformation rules. The BPMN 2.0 OMG specification (OMG, 2013) and BPSIM 2.0 WFMC specifications

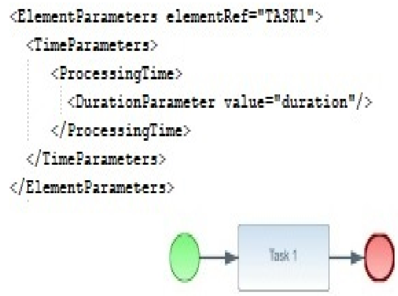


Figure 2. BPMN task extended with BPSIM duration

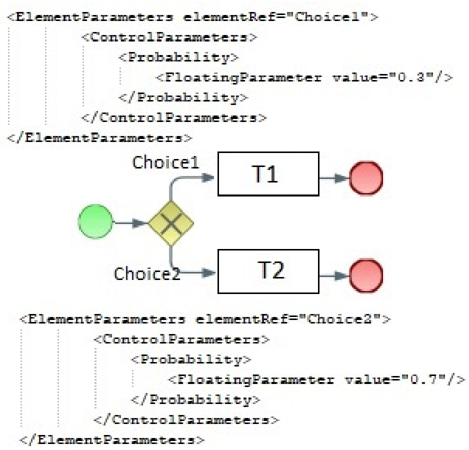


Figure 3. BPMN gateway with BPSIM probabilities on outgoing flows

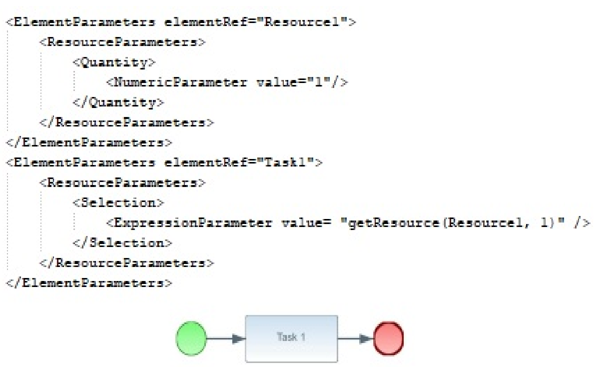


Figure 4. BPMN task using BPSIM Resource allocation

(WFMC, 2016) define separate metamodels for BPMN and BPSIM. We need to construct a new BPMN+BPSIM metamodel using pre-existing BPMN and BPSIM ones. Additionally, as there isn't yet a universal metamodel for DEVS, we'll need to either create one that is focused on BPMN+BPSIM element transformation or utilize an existing metamodel. In order to realize the transformation rules between the source and target model, specialized language needs to be defined. Several transformation languages were proposed such as XSLT (extensible stylesheet language transformation) and ATL (Atlas Transformation Language) (Cetinkaya et al., 2011) (Bazoun and al., 2014) have proposed transformation rules from BPMN to DEVS: These transformations do not cover the BPSIM perspective which aims to enrich the model and only selected BPMN elements were taken into consideration.

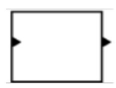
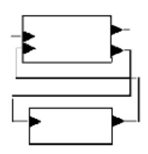
Based on MDE, our goal is to provide a library allowing transformation between source model (BPMN+BPSIM) into target model (DEVS). Therefore, we have started working on the mapping rules for existing BPMN to DEVS transformations by deducing their corresponding BPMN+BPSIM source model and its reflection on the DEVS formalism. Table 1 proposes examples of simple transformation rules for BPMN+BPSIM to DEVS. At first level, without distributed aspects and later we will introduce the distributed aspects. For the sake of simplification, Table 1 just gives information of what needs to be implemented in DEVS. Formalization of these DEVS transformation can be found in (Cetinkaya et al., 2011) and (Bazoun and al., 2014). Models that are expressed using DEVS' fundamental formalism (Zeigler et al., 2011) are referred to as atomic models. Coupled models refer to the composite models.The DEVS model is defined by its sets of input values (X), output values (Y), state variables (S), its internal (δint) *and external* (δext) *transition functions*, *output function*(λ) *and time advance function* (ta) :

$$AtomicDEVS = (X, Y, S, \delta int, \delta ext, \lambda, ta)(1)$$

Referring to the first line in Table 1, the transformation of a simple BPMN task enriched with a BPSIM duration (Figure 2) is mapped to a DEVS atomic model. Based on the transformation rules for BPMN task listed in (Bazoun and al., 2014), the corresponding DEVS model possesses two states an initial infinite passive state S0 and an active State S1. The time advance of S1 is equal to the duration that is specified in BPSIM TimeParameter semantics. The exclusive gateway fork is also mapped to an atomic DEVS model in (Cetinkaya et al., 2011) . The respective DEVS element takes by default equal probability for all outputs. An exclusive fork followed by BPSIM control parameters on the sequence flows as shown bin Figure 3 may be represented by a DEVS atomic model where the probabilities affect the condition of the DEVS model to move from passive to active state. BPMN tasks' resource allocation were represented as coupled DEVS models in (D'Ambrogio and Zacharewicz, 2016). The resource allocation is introduced under BPSIM ResourceParameter. Basic tasks are represented by Atomic DEVS model, moreover whenever a BPSIM resource parameter is added to the task as shown in Figure 4, it is considered as a resource allocation and therefore the resulting DEVS will be the association of both the DEVS atomic model of the task and atomic DEVS model of the resource.

Most of BPMN elements are still not exploited nor mapped to a DEVS representation. One of our objectives is

Table 1. Transformation rules to DEVS.

BPMN+BPSIM	DEVS model	DEVS Behavior
BPMN task + BPSIM duration (Figure 2)		BPSIM Time duration is reflected in DEVS time advance function
BPMN Exclusive gateway + BPSIM probability (Figure 3)		BPSIM controls reflected in the DEVS atomic model condition
BPMN task + BPSIM Resource parameter (Figure 4)		Atomic DEVS to be coupled with the DEVS model to the corresponding task

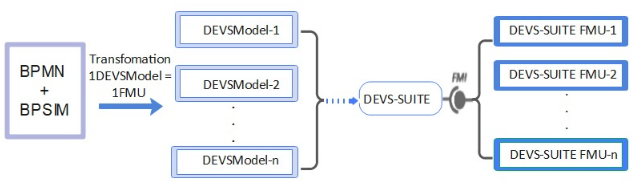


Figure 5. Transformation from BPMN+BPMSIM to FMUs

to create a library of DEVS mapping of all BPMN elements (starting by the most used ones). These mapping will then be expressed as transformation rules in our approach to allow automatic transformation of BPMN+BPSIM models into DEVS. Following this approach, we can also add transformation rules to automate the transformation of BPMN+BPSIM in order to operate a distributed simulation over several CPUs which will be discussed in the following sections.

3.3. Simulation

In the previous section, we have presented transformation from BPMN+BPSIM model into DEVS. Every BPMN that is extended with BPSIM is transformed into either an atomic DEVS or a series of atomic DEVS joined as Coupled DEVS element. Transformation techniques may be used to help in minimizing the processing load by distributing the simulation models into several blocks that could be simulated separately in a distributed way on different processors. One approach is to distribute the resources and separate them from the main flow. Figure 5 illustrates the conceptual approach.

The examples of transformation rules (Table 2) will result in creating a DEVS model containing coupled DEVS (called DEVSModel-i in Figure 5). We aim to convert each of these coupled DEVS into FMUs using an FMI interface. Using our conversion, the link between the generated coupled DEVS must be converted to ensure communication between DEVS-Suite FMUs spread over different CPUs. DEVS-Suite application is FMI compliant (Sarjoughian et al., 2021) and is used in our approach for the FMU development. The resulted FMUs may be simulated in a distributed manner using HLA which considers them as simulation components. Therefore, the FMUs are created as federates under same HLA federation. The HLA RTI is used as a master to FMI components, while the FMUs are the slaves and may exchange information with the master using subscribe and publish mechanism as illustrated in Figure 6. Simulation results are then visualized as heat maps allowing users to review Key performance indicators (KPIs) and Key Risk Indicators (KRI). The simulation information is then communicated to a display tool that will be used to display to the users the simulation results. The principles explained here will be highlighted in an example described in the following section 4.

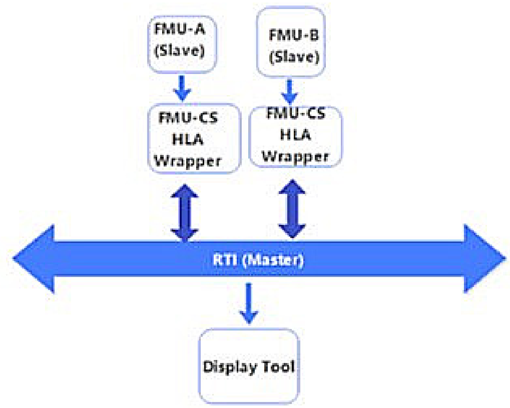


Figure 6. RTI/FMU interaction

Table 2. Transformation rules distributed simulations.

BPMN+BPSIM	DEVS
BPMN Swimlane	Coupled DEVS containing the conversion of all BPMN of the swimlane
BPSIM Resource used by only one BPMN task using no other resource (One on one relation)	Coupled DEVS containing the resource (linked to the coupled DEVS containing the task)
BPSIM Resource used by several BPMN tasks or by a task using other resources (n to m relation)	Coupled DEVS containing a broker and the resources (linked to the coupled DEVS containing the task)

4. Application

In this section, we will use Figure 7 is a BPMN example extracted from (D'Ambrogio and Zacharewicz, 2016) to examine the transformation of the BPMN to multiple DEVS models after adding BPSIM resource extensions.

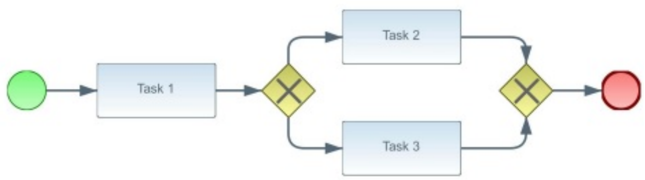


Figure 7. Example of BPMN model

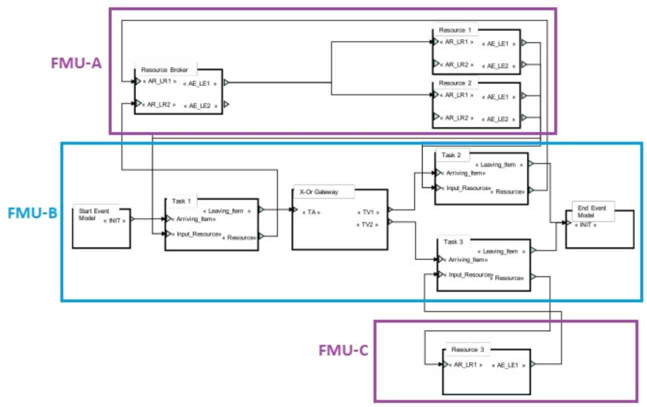


Figure 8. Suggested DEVS decomposition

4.1. Resource allocation example: Transformation of BPMN + PyBPMN annotations to DEVS

The BPMN process consists of a start and an end event, three tasks and two exclusive gateway XOR. The resource allocation used in this example was specified using PyBPMN annotations. Task1 and Task2 use alternative resources which are resource1 and resource2 while Task3 uses resource3.

In the DEVS representation proposed in (D'Ambrogio and Zacharewicz, 2016), a broker model is introduced to manage the alternative resources. Task1 and Task2 are coupled to the broker, Task3 is coupled directly with Resource3. For the sake of distributed simulation and after reviewing the DEVS representation resulting from the example illustrated in Figure 7, according to the rule described previously in Table 2, the resources and the main process will be contained in separate FMUs. Figure 8 illustrates the conceptual decomposition that may be adopted.

4.2. Proposition BPMN+BPSIM into multiple DEVS models

The example in Figure 7 is enriched with BPSIM standard instead of PyBPMN annotations used in (D'Ambrogio and Zacharewicz, 2016), as shown in Figure 9. The alternative resource concept is defined with bpsim:orResource tag where it is possible to specify the alternative resources. For instance, Task1 and Task2 may be enriched with ResourceParameters with orResource expression allowing the selection between alternative resources: bpsim:orResource(Resource1,Resource2). Task3 is also enriched with ResourceParameters.

Unlike Figure 8, the transformation of the BPMN+BPSIM model illustrated in Figure 9 will result in 3 DEVS models:

- DEVSModelA representing the equivalent DEVS model for the BPSIM resource1 and resource2. The alternative resource feature that is mentioned in orResource BPSIM tag will be transformed into a broker model of Resource1 and Resource2.
- DEVSModelB representing the equivalent DEVS model of the main process excluding the resources
- DEVSModelC representing Resource3

5. Conclusion

The literature on BPMN and business model simulation was reviewed in this research. It demonstrates the existence of simulation tools, but the majority of BPMN-compatible business processes simulation software is proprietary. Our goal is to provide a free, open-source web platform for the modeling and simulation of business processes utilizing BPMN for modeling and DEVS for simulation while making use of BPSIM standards as an extra element for achieving distributed simulations. BPMN will be extended with the BPSIM specification for simulation, and DEVS will be used for operational simulation. The three areas of concentration for our study are the definition of a BPMN + BPSIM metamodel, the definition of a general DEVS metamodel, and the expression of transformation rules from (BPMN + BPSIM) to DEVS taking into consideration the creation of multiple DEVS model that will be simulated in a distributed manner using FMU/FMI and HLA. The resulting program is meant to be open source, allowing society to contribute to future improvements, and is intended for any user (simulation experts and non-experts).

References

Awad, A., Grosskopf, A., Meyer, A., and Weske, M. (2009). Enabling Resource Assignment Constraints in BPMN. page 16.

Bazoun, H. and al. (2014). Business Process Simulation: Transformation of BPMN 2.0 to DEVS Models. In

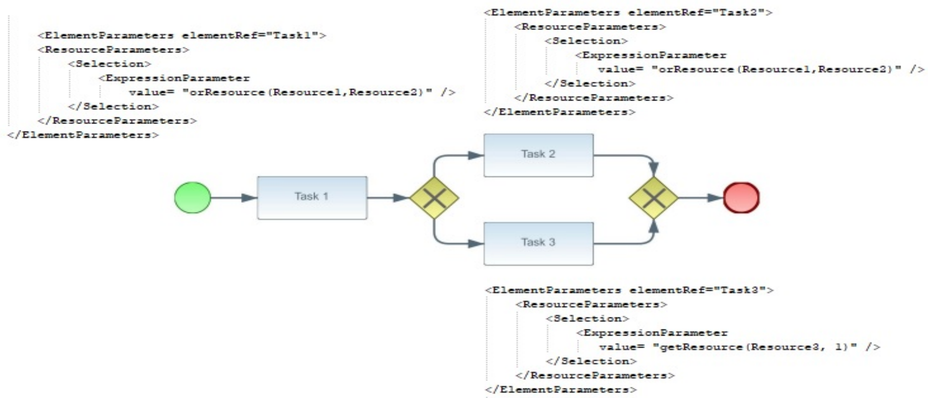


Figure 9. The BPMN example extended with BPSIM for resource allocation

SCS/ACM/IEEE Symposium on Theory of Modeling and Simulation part of SpringSim 2014, Tampa, United States.

Blochwitz, T. (2016). Functional Mock-up Interface for Model Exchange and Co-Simulation.

Cetinkaya, D., Verbraeck, A., and Seck, M. (2011). MDD4MS: A model driven development framework for modeling and simulation. In *Proceedings of the 2011 Summer Computer Simulation Conference*, pages 113–121.

D'Ambrogio, A. and Zacharewicz, G. (2016). Resource-based modeling and simulation of business processes. In *SCSC '16. Summer Computer Simulation Conference*, Montreal, Canada.

Friedenstab, J.-P., Janiesch, C., Matznerand, M., and Müller, O. (2012). Extending BPMN for Business Activity Monitoring. *Proceedings of the Annual Hawaii International Conference on System Sciences*, pages 4158–4167.

Fujimoto, R. (2015). Parallel and Distributed Simulation. In *Proceedings of the 2015 Winter Simulation Conference*, WSC '15, pages 45–59. IEEE Press. event-place: Huntington Beach, California.

IEEE (2010). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)– Framework and Rules. *IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000)*, pages 1–38.

Mallek, S., Daclin, N., and Chapurlat, V. (2010). Towards a conceptualisation of interoperability requirements. In *Enterprise interoperability iv*, pages 439–448. Springer.

Marcinkowski, B. and Kuciapski, M. (2012). A business process modeling notation extension for risk handling. In *IFIP International conference on computer information systems and industrial management*, pages 374–381. Springer.

Mens, T. and Gorp, P. V. (2006). A Taxonomy of Model Transformation. *Electronic Notes in Theoretical Computer Science*, 152:125–142.

Mutarraf, U. and al. (2018). Transformation of Business Process Model and Notation models onto Petri nets and their analysis. *Advances in Mechanical Engineering*, 10(12).

OMG (2013). *Business Process Model and Notation (BPMN) 2.0.2*.

Onggo, B. S. S. and al. (2017). A BPMN extension to support discrete-event simulation for healthcare applications: an explicit representation of queues, attributes and data-driven decision points. *Journal of the Operational Research Society*.

Pasha, M. A.-u.-r. and Pasha, S. (2013). Bloom's Taxonomy for Standardizing BPM Education of IT Under-Graduates Students. *Journal of Computer Applications*, pages 6–13.

Pufahl, L., Wong, T. Y., and Weske, M. (2018). Design of an Extensible BPMN Process Simulator. In Teniente, E. and Weidlich, M., editors, *Business Process Management Workshops*, pages 782–795. Springer International Publishing.

Sarjoughian, H., Zhang, C., and Lin, X. (2021). Control and Decision Communication Across Heterogeneous Model Types. *Center for Model-Based Cyber-Physical Product Development*, 26(15):15–15.

Stroppi, L. J. R., Chiotti, O., and Villarreal, P. D. (2015). Defining the resource perspective in the development of processes-aware information systems. *Information and Software Technology*, 59:86–108.

WFMC (2016). *Business Process Simulation Specification (BPSIM)*.

Yentl, V. T. and Hans, V. (2017). An evaluation of DEVS simulation tools.

Zacharewicz, G. and Hamri, M. E.-A. (2007). Flattening G-DEVS / HLA structure for Distributed Simulation of Workflows. In *AIS-CMS International modeling and simulation multiconference*, pages 11–16, Buenos Aires, Argentina.

Zeigler , B. P., Praehofer, H., and Kim, T. G. (2011). *Theory of Modeling and Simulation*. Elsevier, Academic Press, second edition edition.