

INCREMENTAL TRANSFORMATION OF BPSIM-ENRICHED BPMN MODELS INTO DEVS

Mariane El Kassis
François Troussel
Gregory Zacharewicz
Nicolas Daclin

Laboratory for the Science of Risks
IMT-Mines Ales
6 Avenue de Clavières
Alès, 30100, FRANCE

ABSTRACT

This paper introduces a novel methodology for business process simulation, focusing on the incremental transformation of Business Process Modeling and Notation (BPMN) models enriched with Business Process Simulation Interchange Standard (BPSIM) elements into the Discrete Event System Specification (DEVS) formalism. The proposed method enhances the precision and consistency of simulations by systematically converting BPMN components and BPSIM characteristics into DEVS representations, using adaptable rules and templates. A major contribution of this work is the introduction of the Interaction Intermediate Model (I2M), a model that provides a visually lucid representation with significant semantics, effectively encapsulating BPMN and BPSIM simulation aspects. The resulting DEVS model ensures accurate, reliable, and interoperable simulations. We provide a thorough analysis of this methodology, emphasize its advantages, and validate its efficiency through a case study. This method, applicable across various sectors, equips stakeholders with the ability to make informed decisions based on accurate simulation insights, effectively bridging the gap between conceptual modeling and simulation methodologies.

1 INTRODUCTION

Business Processes (BPs) are a sequence of interrelated actions performed by different organizational components to achieve objectives and build workflow (OMG 2013). They form the cornerstone of any organization, providing the structure and guidance necessary for efficient and effective task execution. By improving and optimizing BPs, business process management helps companies achieve their goals and provide value to stakeholders.

Continuous development and optimization of these processes are vital in today's rapidly evolving and unpredictable business landscape. Modeling and simulation (M&S) have emerged as potent instruments for examining and enhancing business processes. Simulations enable testing diverse scenarios and strategies without disrupting the actual process, fostering informed decision-making. The application of M&S to business processes has gained traction in the context of digital transformation and Industry 4.0 (Steiner et al. 2019), and across industries such as aeronautics, manufacturing, healthcare, banking, and logistics (Jahangirian et al. 2010).

In this paper, we present an innovative methodology for simulating business processes using a two-stage transformation process. Specifically, we propose to use the BPMN 2.0 (Business Process Modeling and Notation) model enriched with BPSIM 2.0 (Business Process Simulation Interchange Standard) to capture

essential simulation elements, and transform it into the DEVS (Discrete Event System Specification) model. We demonstrate the effectiveness of our methodology by presenting detailed examples.

The remainder of this paper is organized as follows: Section 1 provides an introduction and the motivation for our research. Section 2 presents an overview of the current state of business process simulation, including the standardization efforts in this field and the integration challenges of business process modeling and simulation. In Section 3, we introduce our novel methodology for business process simulation, which includes a two-stage transformation process and the Interaction Intermediate Model (I2M). Section 4 illustrates the practical application of our approach with a detailed case study. Finally, in Section 5, we conclude the paper by discussing prospective future research directions and summarizing the significant contributions of our research.

2 STATE OF THE ART

2.1 Background

Business process simulation plays an indispensable role in understanding complex system dynamics and identifying areas that require improvement. A crucial aspect of this process is conceptual modeling, which translates real-world complexities or systems into abstract, comprehensible representations, preparing them for thorough exploration through simulation studies (Robinson 2011). In particular, Discrete Event Simulation (DES) applies conceptual modeling as a comprehensive and intelligible tool to represent discrete events.

The standardization of business processes plays a crucial role in enhancing their understandability and manageability. Various business process management tools aim to achieve this objective by harmonizing graphical notation and software (Mens and Gorp 2006). However, these technologies often have limitations in their simulation capabilities. Many free and open-source business process management software only offer partial support for model simulation (Pufahl et al. 2018), and the integration of modeling and simulation elements remains deficient. This gap requires model extensions to incorporate essential simulation elements such as scenario parameters, resources, and time. Consequently, effective utilization of these tools becomes limited to individuals with profound simulation expertise.

After simulation, analysts can scrutinize the results, identify areas of interest, and strategize for process improvement. Despite the potential advantages of this process, the current state of simulation technologies presents challenges in seamlessly integrating business process modeling and simulation. This scenario emphasizes the urgent need for further exploration and advancements in this field.

2.2 BPMN and Its Extension

The Business Process Model and Notation (BPMN) provides a comprehensive platform for specifying business processes (OMG 2013). However, in its standard format, certain simulation-specific details of a business domain might not be captured entirely. This observation has led to the development of various simulation-oriented extensions that aim to enhance the BPMN's modeling capabilities. These extensions cater to different facets such as temporal dynamics (Gagne and Trudel 2009), interoperability (Mallek et al. 2012), resource performance (Lodhi et al. 2011; Bocciarelli and D'Ambrogio 2011), as well as constraints on resource allocation (Awad et al. 2009) and time (Cheikhrouhou et al. 2013; Pichler et al. 2017).

In particular, the Workflow Management Coalition (WFMC) has introduced the Business Process Simulation (BPSIM) specification as a standard extension. BPSIM brings a wide array of parameters to the table, which serve as additional elements for enriching a BPMN model before the simulation process.

This study utilizes BPSIM framework to extend BPMN, thereby enabling the models to incorporate additional attributes pertinent to simulation, such as time parameters, cost parameters, resource parameters, and priority parameters. The integration of BPMN and BPSIM thus equips the models with a more detailed representation of the business processes, enhancing their applicability in identifying bottlenecks, optimizing resources, and facilitating more informed decision-making processes.

2.3 BPSIM

BPSIM 2.0 specification, a comprehensive open-source framework, serves as a bridge to integrate BPSIM semantics and properties into BPMN models. This framework is the foundation of the methodology presented in this study. Our objective is to enrich BPMN elements with pertinent parameters and logic sourced from BPSIM, thereby enhancing the conceptual model's intricacy and expressiveness.

BPSIM parameters are chiefly classified into five categories (WFMC 2016):

- **Time Parameters:** These delineate time intervals as perceived from an external viewpoint. These parameters can influence the scheduling and execution of BPMN events and tasks, and are essential for representing temporal dynamics in the process model.
- **Control Parameters:** These specify the resources utilized by a business process element. Control parameters are crucial for modeling resource constraints and allocation policies, which can significantly impact process performance and outcomes.
- **Cost Parameters:** These encapsulate all costs affiliated with an activity, whether they involve human or non-human resources. This allows for the comprehensive economic analysis of business processes.
- **Property Parameters:** These designate simulation values for data instances utilized by the business process, thereby enabling the modeling of data-driven behaviors and dependencies.
- **Priority Parameters:** These regulate the priority of the associated BPMN element, influencing the order of execution when resources are scarce or tasks are competing for execution.

The BPSIM 2.0 specification by the Workflow Management Coalition (WFMC 2016) explicitly outlines which BPSIM parameter can be associated with each BPMN element. Consequently, BPSIM enhances BPMN by introducing a variety of capabilities such as probabilistic distribution, concurrent simulations, task-specific resource allocation, probabilistic duration, conditional results probabilities, resource priorities, and calendar availability, among others.

2.4 BPMN Transformation

The simulation of a business process model typically involves a two-step process: the enrichment of the model with pertinent simulation parameters, followed by its transformation into a low-level simulation model (Wagner 2014).

Model-Driven Engineering (MDE) is an approach to software engineering that emphasizes the use of models at various levels of abstraction, thereby shifting the development focus from coding to modeling. MDE hinges on two primary principles: model usage and model transformation. The concept of model transformation includes elements such as mappings and transformation rules, while the model usage principle is centered around concepts like meta-models and conformity (Mens and Gorp 2006).

The subject of BPMN simulation has been an area of active research, with several researchers exploring the transformation of BPMN models into low-level simulation models.

Among these investigations, several researchers have applied DEVS (Discrete Event System Specification) to the simulation of BPMN models (Cetinkaya et al. 2011; Bocciarelli and D'Ambrogio 2011; Bazoun et al. 2014; Boukelkoul and Maamri 2015). DEVS is a well-structured formalism for discrete event modeling and simulation, offering a hierarchical and modular approach, which enhances the comprehensibility and maintainability of models (Zeigler et al. 2011).

On the other hand, Petri nets have also been extensively used for the simulation of BPMN models (Dijkman et al. 2007; Ou-Yang and Lin 2008). Petri nets are a popular mathematical modeling tool that is particularly useful for the description and analysis of systems that are characterized by concurrency, synchronization, and resource sharing.

In the state of the art, both DEVS and Petri Nets have been used for the simulation of BPMN models. DEVS is appreciated for its expressiveness, modularity, support for hierarchical modeling, and simplicity,

and is widely supported by various simulation tools. On the other hand, Petri Nets, while also a powerful tool, lack certain features such as modularity and hierarchical modeling.

3 METHODOLOGY

In this paper, we present an innovative methodology for simulating business processes, aiming to bridge the gap in simulation expertise and empower BPMN end-users to make informed decisions based on accurate representations of their processes. Our methodology leverages BPMN and BPSIM, industry-accepted standards for modeling and simulating business processes. BPMN provides a universally understood language that effectively models our processes, while BPSIM's specific features facilitate meaningful simulation. For our simulation model, we have chosen DEVS due to its hierarchical and modular nature, making it suitable for modeling complex systems.

The advantages of DEVS lie in its ability to capture system dynamics effectively. Additionally, DEVS is widely implemented across various simulation tools, enhancing its practicality for simulating business processes. In the practical implementation of DEVS, we have selected DEVS-Suite as our simulator, influenced by its prior application in model-driven engineering research (Sarjoughian and Mahmoodi Markid 2012). DEVS-Suite's suitability aligns with our methodology, ensuring a seamless integration of DEVS-based simulation.

To address the oversight of simulation parameters and interaction in previous works, we propose the Intermediate Interaction Model (I2M). This separate modeling approach enables explicit modeling of interactions and ensures comprehensive transformation rules. Our methodology prioritizes incrementality, flexibility, and accuracy, allowing modifications to transformation rules at any stage while accurately representing crucial aspects in simulation. The I2M serves as a comprehensive solution that empowers users to bridge the gap in simulation expertise, enabling them to understand and review simulation results and make informed decisions based on accurate representations of their business processes.

3.1 Intermediate Interaction Model (I2M)

In our endeavor to convert a BPMN-based business process enriched with BPSIM into DEVS, we conducted a comprehensive exploration of potential combinations. Our objective was to identify and implement necessary transformations while preserving vital functionalities. However, the extensive range of allowed combinations between BPMN and BPSIM posed a significant challenge. To address this, we developed the Intermediate Interaction Model (I2M), which consists of two types of elements: I2M simple elements and I2M composite elements. The I2M model serves as a bridge, simplifying the transformation process and allowing for an intuitive representation of simulation functionalities.

The I2M simple elements are derived from an in-depth analysis of BPMN and BPSIM, encapsulating essential simulation functionalities and presented in a BPMN-like manner. For instance, time-related elements in BPMN and BPSIM are represented as I2M timers, visually resembling BPMN timers but with enhanced semantics adapted to our needs. Notably, while I2M elements bear visual similarities to BPMN components, their functionality is adapted, and they can interact in ways that BPMN elements cannot. For example, an I2M timer can be connected to another element with a message flow, an operation not possible with BPMN timers. To visually distinguish I2M elements from BPMN components, we have given them a blue border. This demarcation underscores the distinctive role and operation of I2M elements within our methodology, emphasizing their unique semantics while maintaining a familiar visual form.

However, certain elements in BPMN and BPSIM encompassed multiple simulation functionalities of interest. To accommodate these complexities, we introduced I2M composite elements. An I2M composite element represents an assembly composed of interconnected I2M simple elements. These connections are governed by a pre-established Generic Interaction Model (GIM), configured by an I2M engineer. This composition embodies a higher level of abstraction, facilitating the simulation of complex systems by orchestrating simpler elements in a structured and organized fashion. This differentiation between a

list of intermediate interaction elements and the interconnected equivalent model enabled us to create a streamlined transformation to DEVS. Our approach followed an incremental methodology that empowered users to define the interconnections between the simple elements based on their simulation requirements. By adopting this approach, we successfully shifted the complexity away from the implementation in DEVS, simplifying the overall transformation process.

By introducing the I2M model, we have effectively separated functionality from interaction, greatly simplifying the transformation to DEVS. This separation means that the complexity of the process is now primarily focused on the coupling of these elements, rather than on the implementation of complex DEVS models themselves. Consequently, the translation into DEVS is facilitated, as we are primarily dealing with simple I2M elements whose interaction patterns dictate the structure of the DEVS models. Thus, the intricacy is now centered on how these elements are interconnected, which is a far more manageable challenge.

Our methodology revolves around the use of an Intermediate Interaction Model as a bridge between BPMN+BPSIM and DEVS. The I2M serves as a transitional space, capturing every element from the BPMN+BPSIM model and providing an equivalent representation that mirrors the structure of BPMN while centralizing interactions. A key aspect of our methodology is the Generic Interaction Model (GIM), which comprehensively illustrates potential interactions between I2M elements. Incremental development and adaptability are emphasized, enabling enhancement of the method without extensive code modifications.

3.2 I2M Simple and Composite

The metamodel for the I2M is a critical component of our methodology, providing a structured representation of the I2M. It bears a resemblance to the BPMN model, both being visual representations with meaningful semantics. However, the I2M goes a step further by effectively illustrating BPSIM simulation aspects, making the simulation process and its impact on the overall model easier to comprehend. Our metamodel consists of two main elements: the simple I2M elements and the composite elements, which may include simple and composite I2M elements.

Simple I2M Elements: These elements represent individual functionalities, which could correspond to a BPMN or BPSIM element and which are important for the simulation. The simple I2M elements include `i2m:Start()`, `i2m:Stop()`, `i2m:Timer()`, `i2m:Send()`, `i2m:Receive()`, `i2m:Container()`, `i2m:MessageFlow()`, `i2m:SequenceFlow()`, `i2m:Pool()`, `i2m:Join()`, `I2M:Task()`. To reflect resource allocation, we have also defined `i2m:Broker()`, `i2m:Allocate()`, and `i2m:UnAllocate()`. To express stochastic behavior, we have created the `i2m:RandomNumberGenerator()`.

Composite I2M Elements: A composite I2M element is a set of simple I2M elements interconnected by preconfigured rules. This set of I2M elements is housed within an `i2m:Container()` element. The final transformation of these composite elements will follow a simplification rule based on the I2M elements contained in the set according to the GIM associated with these composite elements.

3.3 Transformation

Inspired by Model Driven Architecture (MDA), we propose in Figure 1 a transformation methodology facilitated by an Intermediate Interaction Model (I2M).

The first step in our approach involves a model-to-model transformation where we transform BPMN+BPSIM model into the I2M model. This transformation is executed using the Atlas Transformation Language (ATL), a tool that allows us to describe complex transformations in a maintainable, readable, and declarative way (Jouault et al. 2008).

I2M plays the role of a pivot in our methodology, serving as a bridge model that captures all essential interaction aspects. It acts as a common intermediary, enabling a smooth transformation process from BPMN+BPSIM to DEVS. Once the BPMN+BPSIM model is transformed into the I2M, the subsequent transformation step involves the conversion of the I2M into a DEVS model which is carried out using

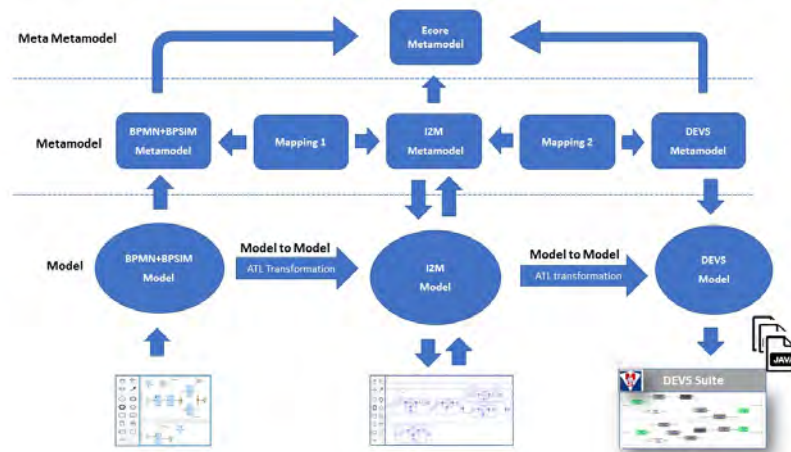


Figure 1: Dual transformation architecture.

Model to Model transformation. Finally, DEVS-Suite code is generated using Model to Text (JAVA Code) transformation similar to the one performed in Robati et al. (2015). The generated code can then be executed in the DEVS-Suite simulator (Sarjoughian and Mahmoodi Markid 2012). This final step completes the transformation process, allowing for the study of the behavior and performance of the business process expressed in BPMN+BPSIM. By adhering to this methodology, we provide an efficient and flexible approach to transforming BPMN models, enriched with BPSIM parameters, into a simulation-ready format. This approach is advantageous not only for its accuracy in capturing the dynamic nature of business processes but also for its adaptability, allowing for incremental changes and updates without the need for extensive code revisions.

To implement our two-phase transformation process, we require the following:

- A metamodel for BPMN (OMG 2013) and a metamodel for BPSIM BPSIM (WFMC 2016)
- A metamodel for I2M, a contribution of the present paper
- A metamodel for DEVS, EMF-DEVS metamodel designed for EMF development and used along with the DEVS-Suite simulator (Sarjoughian and Mahmoodi Markid 2012)
- Mapping rules for the transformation from BPMN+BPSIM to I2M
- Mapping rules for the transformation from I2M to DEVS

These components ensure that the transformation process is systematic, consistent, and accurate, resulting in DEVS models that faithfully represent the original BPMN+BPSIM models, ready for effective simulation.

3.3.1 BPMN + BPSIM to I2M

In this section, we elucidate the transformation process from BPMN+BPSIM elements into I2M counterparts. This transformation serves as a cornerstone in our approach to facilitating the seamless transition from business process models into simulation-ready constructs. The mapping presented herein is not arbitrary but is grounded in the functional equivalence of BPMN+BPSIM elements to I2M primitives. This equivalency ensures that the behavior and semantics of the original business process model are preserved in the transformed model, thereby enhancing the validity and credibility of any ensuing simulation studies.

Table 1 illustrates some of these mappings and provides a concise explanation for each. This mapping, while not exhaustive, covers a wide range of typical BPMN+BPSIM constructs, offering a substantial foundation for the transformation of more complex business processes.

Table 1: Selected Mapping of BPMN+BPSIM Elements to I2M Simple Elements.

| BPMN + BPSIM | I2M | Explanation |
|----------------------|--|---|
| Start event | i2m:Start() | Triggers the start of a process |
| End event | i2m:Stop() | Signifies the end of a process |
| Timed event/Delay | i2m:Timer() | Captures time constraints or durations |
| Sending events | i2m:Send() | Captures sending interaction of a message i2m:Send(message) or a signal i2m:Send(signal) |
| Receiving events | i2m:Receive() | Captures receiving interaction of a message i2m:Receive(message) or a signal i2m:Receive(signal) |
| Sequence flow | i2m:SequenceFlow() | Represents the order of activities |
| Message flow | i2m:MessageFlow() | Represents the exchange of messages |
| Sub-process | i2m:Container() | Represents a logical grouping or encapsulation of activities used to house a set of interconnected i2m elements |
| Parallel Join | i2m:Join() | Represents the synchronization of concurrent activities |
| Parallel Fork | i2m:Fork() | Represents the splitting of control flow into concurrent activities |
| Simple task | i2m:SimpleTask() | Represents a unit of work that is not further decomposed |
| Resource allocation | i2m:Broker(), i2m:Allocate(), i2m:Unallocate() | Models resource allocation and deallocation |
| Stochastic behaviors | i2m:RandomNumberGenerator() | Captures variability and randomness |

Table 2 presents an assortment of mappings that involve composite I2M elements. These mappings do not explicitly denote the interconnections between the corresponding I2M elements but rather provide an illustrative catalog of the component elements contained within each composite element.

Table 2: Transformation of BPMN+BPSIM into Composite I2M elements.

| BPMN+BPSIM Element Type | Specific Element | Composite I2M Element |
|-------------------------|---------------------|---|
| Start Events | Timed start event | i2m:Start() → i2m:Timer() |
| | Message start event | i2m:Start() → i2m:Send(message) |
| | Signal start event | i2m:Start() → i2m:Send(signal) |
| End Events | Message end event | i2m:Send(message) → i2m:Stop() |
| | Signal end event | i2m:Send(signal) → i2m:Stop() |
| Tasks | Send Task | i2m:Container{i2m:Start(), i2m:SimpleTask(), i2m:Send(message), i2m:Stop() } |
| | Receive Task | i2m:Container{i2m:Start(), i2m:SimpleTask(), i2m:Receive(message), i2m:Stop() } |
| Resources | Resource allocation | i2m:Container{i2m:Broker(), i2m:Start(), i2m:Allocate(), i2m:Unallocate(), i2m:Stop() } |

Certain I2M elements, notably i2m:Broker() and i2m:RandomNumberGenerator(), are unique per process flow. Their uniqueness plays a critical role in ensuring the reproducibility of simulations, which is a cornerstone of credible simulation studies.

The i2m:Broker() guarantees consistency in resource allocation across multiple simulations, while i2m:RandomNumberGenerator() ensures the generation of reproducible stochastic behaviors. This approach upholds the reliability of our simulation results, enhancing their value in facilitating informed decision-making in business process management.

The composite elements listed in Table 2 are either directly linked with an i2m:SequenceFlow() or contained in an i2m:Container(). These elements are interconnected based on a preconfigured GIM for that

particular type, such as a task. By simplifying the GIM using mapping rules, we determine the relevant interconnections for the transformed composite, ensuring it accurately represents the relationships and dependencies specific to the task’s BPMN+BPSIM functionalities.

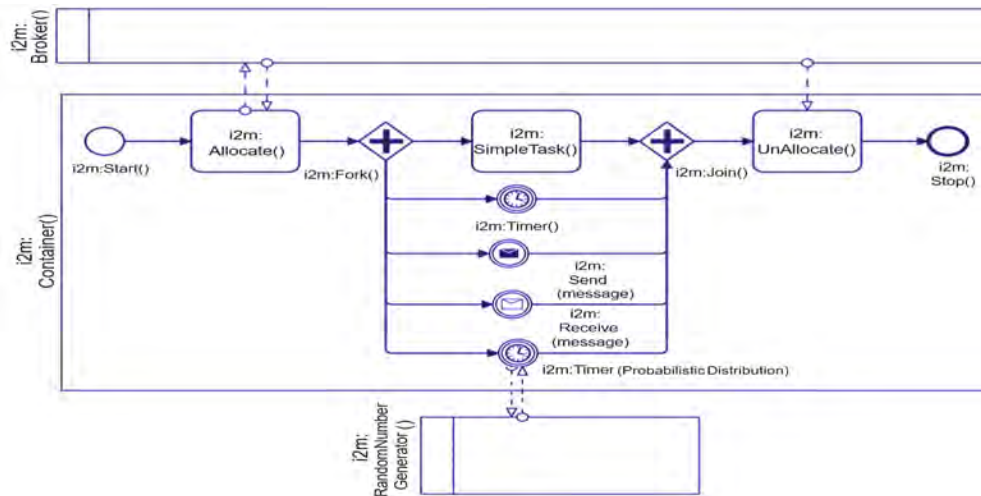


Figure 2: GIM (Generic Interaction Model) for Tasks.

3.3.2 I2M to DEVS

In the process of our transformation from I2M to DEVS, we meticulously identify corresponding atomic and coupled DEVS components for each component in the i2m model. For instance, an i2m:Allocate() aligns with a DEVS Allocate element, and an i2m:SimpleTask() maps seamlessly to a DEVS SimpleTask element, amongst other direct correspondences.

As a consequence, the final DEVS model we generate is a precise reflection of the initial I2M model. This model can then be simulated using DEVS Suite, providing users with valuable insights into the behavior and performance of the represented business process.

To illustrate our methodology, we examine a specific BPMN Receive Task that is enriched with resources and time parameters from BPSIM. The interaction intermediate model (I2M) representation derived from these functionalities is:

```
i2m:Container{i2m:Start(), i2m:Timer(), i2m:Receive(),
i2m:Stop(), i2m:Allocate(), i2m:Unallocate(), i2m:Broker() }
```

As a part of our methodology, we simplify this interconnected I2M model. Accordingly, we initiate this process with the GIM of a task, depicted in Figure 2. Here, the objective is to selectively eliminate any redundant elements and connections that are not required for the specific integration of the Receive Task with resources and time.

The outcome of this simplification process is a more streamlined I2M model, represented in Figure 3. For clarity, we have marked the elements to be removed in red.

Transitioning from the I2M model to the DEVS framework is notably seamless. We’ve already developed all basic I2M elements within the DEVS Suite; the final transformation phase concentrates on their coupling to create the comprehensive DEVS model.

Each I2M element undergoes a one-to-one transformation into a DEVS representation, preserving original behavior and semantics without loss of information. This process is simplified due to the DEVS elements being purpose-built to reflect the functionality of their I2M counterparts.

However, the transformation of the Receive Task posed a challenge due to its three-branched join element in the I2M model. As our DEVS Suite includes a two-branched join element, we devised a unique

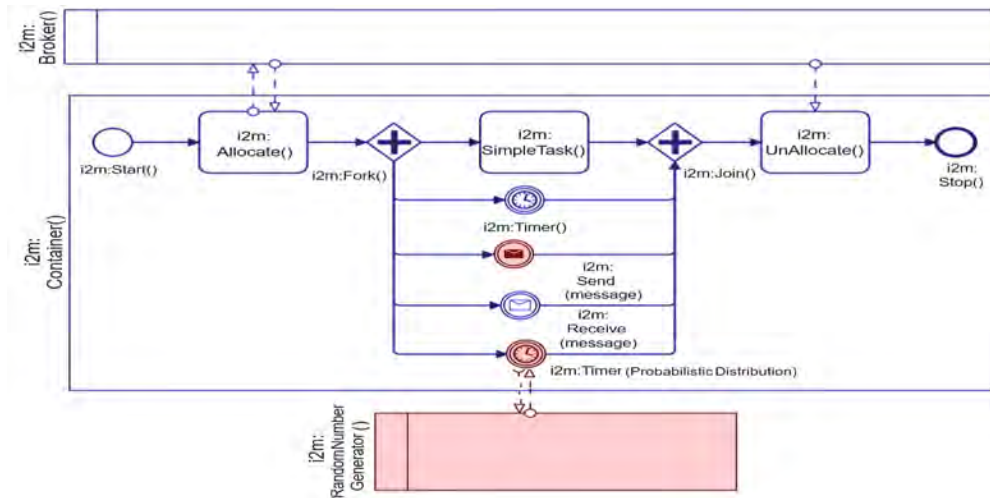


Figure 3: Simplified I2M Composite of BPMN+BPSIM Receive Task.

coupling strategy using two DEVS join elements, namely join1 and join2, to maintain fidelity. The outcome is a faithful DEVS model, as shown in Figure 4, accurately capturing the original Receive Task behavior, further enriched by BPSIM’s resources and time parameters.

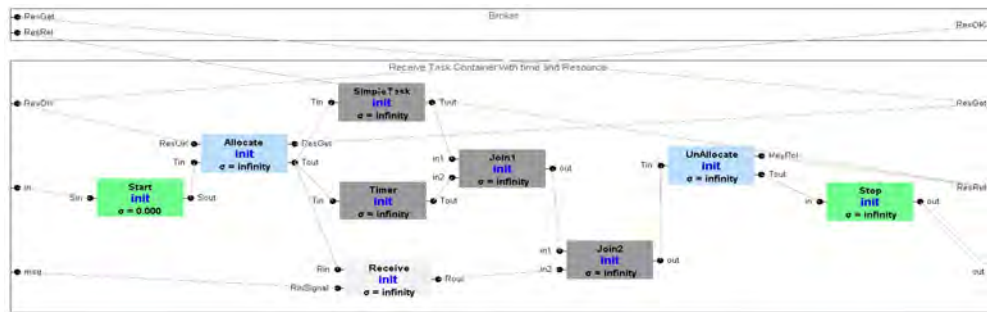


Figure 4: DEVS representation of a receive task Container with Resource and time.

4 APPLICATION CASE

In this section of the paper, we present a practical demonstration of a BPMN+BPSIM process to emphasize the effectiveness of our proposed transformation approach. We’ve selected an example that includes tasks using shared resources and allocated specific durations, with resource assignments and temporal parameters outlined in BPSIM.

Figure 5 displays a graphical representation of the process. It depicts the consecutive execution of Tasks 1 and 2, followed by the concurrent execution of Tasks 3 and 4, highlighting the intertwined resource dependencies and timeframes associated with each task.

Initiating the transformation journey, we convert this BPMN+BPSIM process into its equivalent I2M representation. Subsequently, we move on to the final transformation phase, where we turn it into a DEVS model. This practical example confirms the utility of our methodology in accurately capturing resource assignments, temporal parameters, and task interdependencies, enabling the construction of a comprehensive and reliable DEVS simulation model.

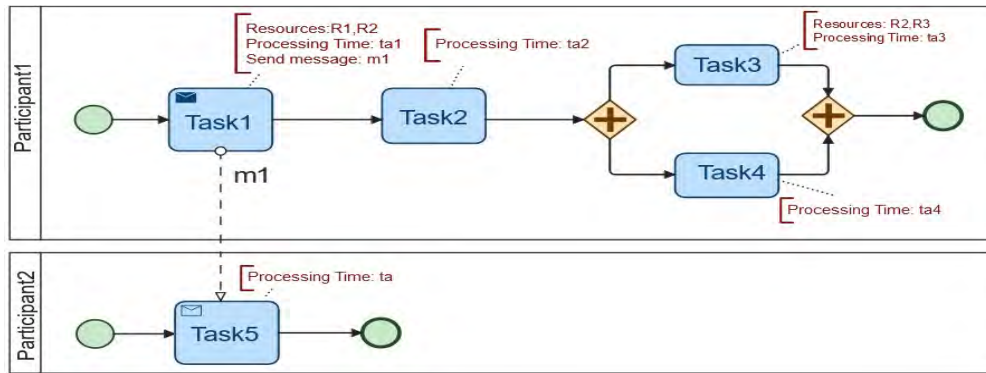


Figure 5: Application case in BPMN + BPSIM.

Figure 6 unveils the completely transformed I2M model. This model, developed in accordance with the GIM model of a Task as referenced in Figure 2 and guided by the mappings delineated in Table 2, allows for an in-depth exploration of the transformed elements and their interrelationships.

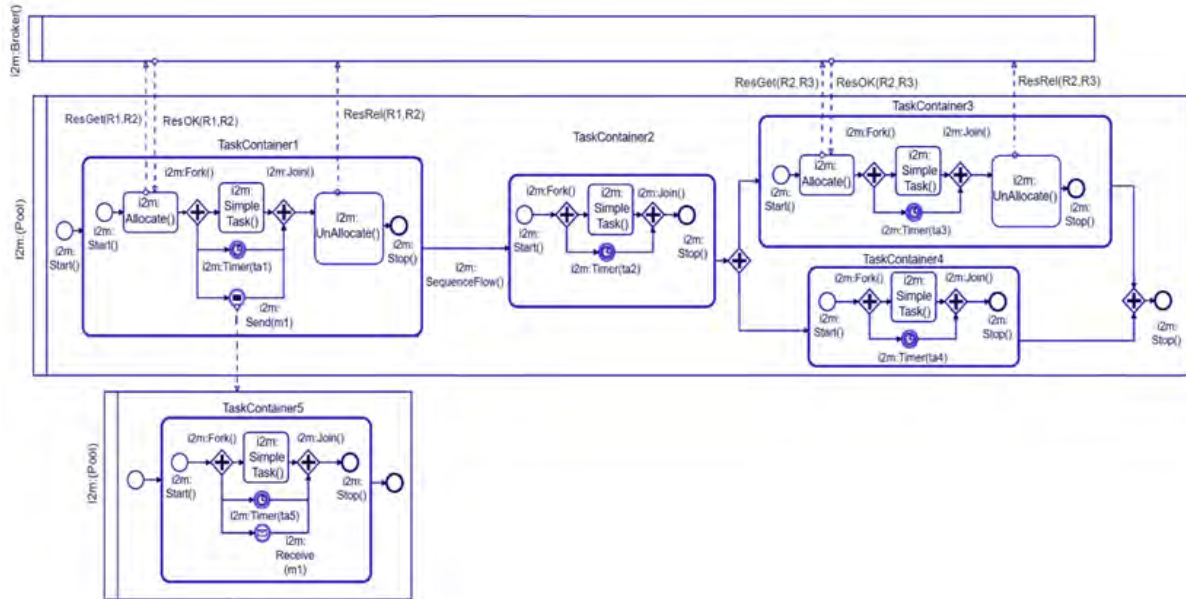


Figure 6: I2M resulting model.

Progressing with the transformation, the I2M model is translated into a DEVS model, integrating a high degree of detail and accuracy. This model is distinguished by its unique structural hierarchy, reflecting the intricate choreography between different tasks and resources within the process.

As displayed in Figure 7, the DEVS model is meticulously organized and interconnected, forming a sophisticated network of coupled DEVS elements. This coupling accentuates the intrinsic complexity of the process while also offering a detailed blueprint for task execution and management.

This dual-phase transformation, from BPMN+BPSim to I2M, and subsequently, to DEVS, delivers an exhaustive and detailed perspective of the process, fostering a deeper grasp of its architecture and dynamics. This profound comprehension sets the stage for future enhancements and refinements, driving the process toward heightened efficiency and effectiveness.

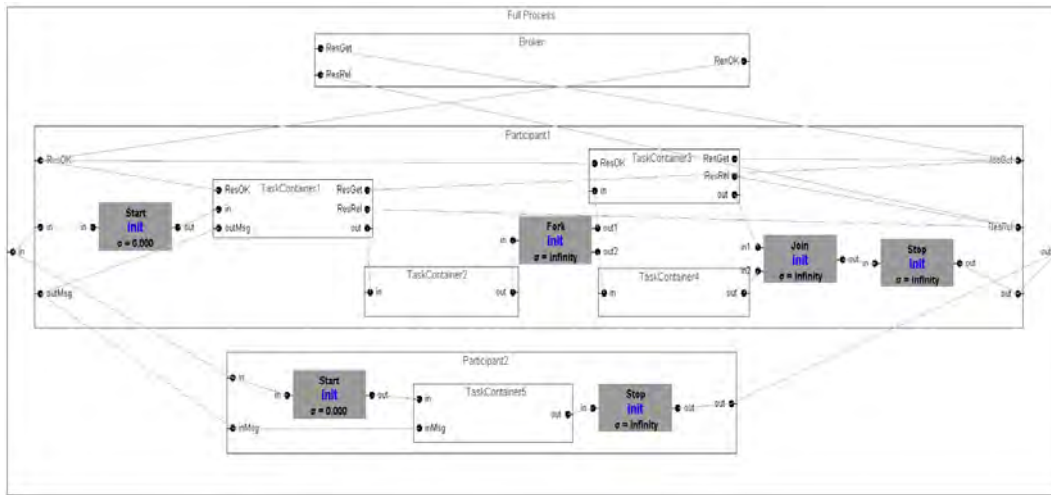


Figure 7: Final DEVS target model.

5 CONCLUSION

This paper presents a pioneering methodology for simulating business processes, introducing the concept of the Intermediate Interaction Model (I2M) as a crucial component. Anchored in industry standards and guided by Model-Driven Architecture principles, our methodology simplifies the transformation process and bridges the gap between business process modeling and discrete event simulations.

The I2M model, introduced as a significant contribution in this study, plays a pivotal role by translating BPMN+BPSIM elements into equivalent DEVS components, effectively linking the two domains. This innovative approach addresses a long-standing challenge in the field, enhancing the interplay between business process modeling and simulation.

Our methodology involves the meticulous mapping of BPMN and BPSIM components onto the I2M model, followed by their transformation into atomic DEVS models. By leveraging the control flow logic of the process, these atomic models are coupled to produce the final DEVS coupled model. This dual-stage transformation process, facilitated by the I2M model, effectively encapsulates the complexity while ensuring simplicity, modularity, flexibility, and enhanced simulation capabilities.

Moving forward, our future research will focus on advancing the capabilities of the I2M model and developing additional transformation rules to further refine and expand the potential of our methodology.

In conclusion, this study highlights our significant contribution in introducing the I2M within the proposed methodology. By simplifying the transformation process, our methodology establishes a solid foundation for the development of open-source, community-driven tools in the field. It also encourages further research and advancements in the integration of business process modeling and simulation, making a valuable contribution to the broader academic and industrial communities.

REFERENCES

- Awad, A., A. Grosskopf, A. Meyer, and M. Weske. 2009. "Enabling Resource Assignment Constraints in BPMN". Technical report, BPT in Hasso Plattner Institute, Potsdam.
- Bazoun, H., Y. Bouanan, G. Zacharewicz, H. Boyer, and Y. Ducq. 2014. "Business Process Simulation: Transformation of BPMN 2.0 to DEVS Models". In *SCS/ACM/IEEE Symposium on Theory of Modeling and Simulation Part of SpringSim 2014*. Tampa, United States.
- Bocciarelli, P., and A. D'Ambrogio. 2011. "A BPMN Extension for Modeling Non-Functional Properties of Business Processes". In *Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium*, 160–168.
- Boukelkoul, S., and R. Maamri. 2015. "Optimal model transformation of BPMN to DEVS". In *AICCSA*, 1–8.

- Cetinkaya, D., A. Verbraeck, and M. Seck. 2011. "MDD4MS: A Model Driven Development Framework for Modeling and Simulation". In *Proceedings of the 2011 Summer Computer Simulation Conference*, 113–121.
- Cheikhrouhou, S., S. Kallel, N. Guermouche, and M. Jmaiel. 2013. "Toward a Time-Centric Modeling of Business Processes in BPMN 2.0". In *Proceedings of International Conference on Information Integration and Web-Based Applications & Services, IIWAS '13*, 154–163. New York, NY, USA: Association for Computing Machinery.
- Dijkman, R. M., M. Dumas, and C. Ouyang. 2007. "Formal Semantics and Analysis of BPMN Process Models Using Petri Nets". *Queensland University of Technology, Tech. Rep.*:1–30.
- Gagne, D., and A. Trudel. 2009. "Time-BPMN". In *2009 IEEE Conference on Commerce and Enterprise Computing*, 361–367.
- Jahangirian, M., T. Eldabi, A. Naseer, L. K. Stergioulas, and T. Young. 2010. "Simulation in Manufacturing and Business: A Review". *European Journal of Operational Research* 203(1):1–13.
- Jouault, Allilaire, Bézivin, and Kurtev. 2008. "ATL: A Model Transformation Tool". *Science of Computer Programming* 72(1-2):31–39.
- Lodhi, A., V. Köppen, and G. Saake. 2011, 01. "An Extension of BPMN Meta-model for Evaluation of Business Processes". *J. Riga Technical University* 43:27–34.
- Mallek, S., N. Daclin, and V. Chapurlat. 2012. "The Application of Interoperability Requirement Specification and Verification to Collaborative Processes in Industry". *Computers in Industry* 63(7):643–658.
- Mens, T., and P. V. Gorp. 2006. "A Taxonomy of Model Transformation". *Electronic Notes in Theoretical Computer Science* 152:125–142.
- OMG 2013. *Business Process Model and Notation (BPMN) 2.0.2*.
- Ou-Yang, C., and Y. Lin. 2008. "BPMN-based Business Process Model Feasibility Analysis: A Petri Net Approach". *International Journal of Production Research* 46(14):3763–3781.
- Pichler, H., J. Eder, and M. Ciglic. 2017. "Modelling Processes with Time-Dependent Control Structures". In *Conceptual Modeling*, edited by H. C. Mayr, G. Guizzardi, H. Ma, and O. Pastor, 50–58. Cham: Springer International Publishing.
- Pufahl, L., T. Y. Wong, and M. Weske. 2018. "Design of an Extensible BPMN Process Simulator". In *Business Process Management Workshops*, edited by E. Teniente and M. Weidlich, 782–795: Springer International Publishing.
- Robati, T., A. El Kouhen, A. Gherbi, and J. Mullins. 2015. "Simulation-Based Verification of Avionic Systems Deployed on IMA Architectures". In *ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MoDELS'15)*. Ottawa, Canada.
- Robinson, S. 2011. *Conceptual Modeling for Discrete-Event Simulation*. Boca Raton: CRC Press.
- Sarjoughian, H., and A. Mahmoodi Markid. 2012, 03. "EMF-DEVS modeling". Volume 44.
- Steiner, F. et al. 2019. "Industry 4.0 and Business Process Management". *Tehnički Glasnik* 13(4):349–355.
- Wagner, G. 2014. "Tutorial: Information and Process Modeling for Simulation". In *Proceedings of the Winter Simulation Conference 2014*, edited by A. Tolk, L. Yilmaz, S. Y. Diallo, and I. O. Ryzhov, 103–117. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- WFMC 2016. "Business Process Simulation Specification (BPSIM)". <https://www.bpsim.org/specifications/2.0/WFMC-BPSWG-2016-01.pdf>. Accessed 19th March.
- Zeigler, B., Praehofer, and Kim. 2011. *Theory of Modeling and Simulation*. second edition ed. Elsevier, Academic Press.

AUTHOR BIOGRAPHIES

MARIANE EL KASSIS is a Ph.D. candidate at IMT Mines Alès, France, specializing in business process simulation and distributed simulations. She is currently focused on creating a web-based platform for M&S business processes, aiming to enhance the understanding and optimization of operational workflows. Her email address is mariane.el-kassis@mines-ales.fr.

FRANÇOIS TROUSSET is an Associate Professor at IMT Mines Alès. He received his Ph.D. from the University of Besançon in 1992, on modeling software process and performing model verification. He worked as a system and network engineer for about 20 years before returning to research activities. His research interests focuses on Artificial Intelligence, Decision Making, Modeling and Simulation. His email address is francois.troussel@mines-ales.fr.

GREGORY ZACHAREWICZ is a Full Professor at IMT Mines Alès. He received his Ph.D. from Aix-Marseille University in 2006, and his Habilitation to lead Research from the University of Bordeaux in 2014. His research focuses more generally on distributed DES M&S, DEVS, ERP, BPMN and workflow. His email address is gregory.zacharewicz@mines-ales.fr.

NICOLAS DACLIN is an Associate Professor at IMT Mines Alès, His research interests focuses on process-centric ecosystems for process management, orchestration, and M&S. He received his Ph.D. from the University of Bordeaux in 2007 and his Habilitation to lead Research from the University of Montpellier II in 2017. His email address is nicolas.daclin@mines-ales.fr.