A Discrete-Event Simulation Metamodel for Obtaining Simulation Models from Business Process Models

M. Teresa García, M.A. Barcelona, M. Ruiz, L. García-Borgoñón, and I. Ramos

Abstract Organizations need to be agile and fl exible to meet the continuous changes. Business Process Management (BPM) is harnessing the continuous changes suffered by organizations in the value chain and, therefore, in their processes. Simulation models offer the ability to experience different decisions and analyze their results in systems where the cost or risk of actual experimentation are prohibitive. BPMN models are not directly executable nor is it possible to simulate their behavior in various input parameters. This paper proposes the application of model-driven engineering (MDE) to integrate the defi nition of business processes with Discrete- Event Simulation (DES) as a tool to support decision-making. We propose a platform independent DES metamodel and a set of rules, to automatically generate the simulation model from BPMN 2.0 based business process in accordance with previous metamodel.

Keywords Model Driven Engineering • Business Process • Discrete-Event Simulation • Decision-Making

M.T. García (✉) • M. Ruiz
University of Cádiz, C/Chile,S/N , 11003 Cádiz, Spain
e-mail: mayte.garcia@uca.es; mercedes.ruiz@uca.es

M.A. Barcelona • L. García-Borgoñón
AragonInstituteofT echnology, c/MaríadeLuna7 , 50018 Zaragoza, Spain
e-mail: mabarcelona@ita.es; laurag@ita.es

I. Ramos
University of Seville, Av. Reina Mercedes S/N, 41012 Sevilla, Spain
e-mail: iramos@us.es

25.1 Introduction

Due to the changing and competitive environment facing organizations today, it is necessary to keep alive the concept of continuous improvement towards excellence.

Organizations need to be agile and flexible to meet the continuous changes that are subjected (e.g. change in customer needs, in applicable law or technology available at the organization's staff) be effective and efficient if they want to build their business in the medium and long term. The business and technical layers must be able to combine their efforts.

In the business context, Business Process Management (BPM) is harnessing the continuous changes suffered by organizations in the value chain and, therefore, in their processes. BPM techniques enable analysts to handle all aspects of these processes. However, today, there remains a significant gap between the analysts and software engineers who must develop applications that support (all or part) the business process.

Simulation models offer the ability to experience different decisions and analyze their results in systems where the cost or risk of actual experimentation are prohibitive. At company level, the simulation of processes associated with business can be very helpful for making strategic decisions, tactical and operational.

The business process modeling is the representation of the activities of the business processes of an organization to be analyzed and improved [13]. Currently, BPMN 2.0 [10] is used to model business processes, which is an evolved version of BPMN 1.0 [9] which is considered the de facto standard for this. However, BPMN models are not directly executable nor is it possible to simulate their behavior in various input parameters. Thus, for these models resulting from greater utility, particularly during the decision-making process, it would be interesting to transform them into others who can simulate and analyze.

Today there are several tools for modeling business processes, but they do not allow their simulation in the broad sense of the word. Similarly, it is possible to find numerous simulation tools, although not specifically designed to simulate business processes.

This paper proposes the application of model-driven engineering (MDE) to integrate the definition of business processes with Discrete-Event Simulation (DES) as a tool to support decision-making. We propose a platform independent DES metamodel and a set of rules, to automatically generate the simulation model from BPMN 2.0 based business process in accordance with previous metamodel.

The rest of the paper is organized as follows. First, we present the state of the art and related work in Sect. 25.2. Then Sect. 25.3 exposes the problem statement. Basic concepts related to DES are introduced in Sect. 25.4, which precedes the metamodel showed in Sect. 25.5. A case study is presented in Sect. 25.6. Finally, Sect. 25.7 shows the conclusions and directions for future research.

25.2 Related Work

In this section we present related work on the subject. The term Software Process Simulation Modeling (SPSM) comes from the first ProSim workshop in 1998 in which Kellner, Madachy and Raffo (KMR) discussed the "why, what and how" of process simulation [7]. They identified the questions and issues that simulation could be used to address, the scope and variables that could be usefully simulated and the modeling approaches that could be employed.

Ten years later Zhang et al. [14] performed a systematic literature review of software process simulation papers fom the ProSim series[1] and they analized the facts, trends and future research directions. They identified as major trends: (1) system dynamics (SD) and discrete-event simulation form the main stream of SPSM paradigms; (2) new simulation paradigms continue to be introduced; (3) continuous modeling gradually lost its dominant position in SPSM research in comparison with discrete approached during the decade; (4) most of newly introduced paradigms enhanced the research capability at the micro-process level; (5) in recent years, micro- processes have been attracting more simulation researchs. They identified the following future research directions: (1) more recent modern software development processes need to be further investigated (e.g. agile, open-source and global development; (2) more new simulation paradigms need to be experimented and introduced; (3) more attempts are needed to effectively tackle the uncertainty of software process in practice; (4) hybrid simulation models should address more than SD and DES in vertical integration; (5) process simulation models should become more reusable, which makes them easier to build.

Raffo et al. [12] described in detail how Process Simulation Modeling supports each of the various CMMI Process Areas from level 2 through level 5. They pointed out some of the tangible benefits that PSIM can provide: (1) selection of the best possible development process, quality assurance strategy, or set of tools for a given project, situation or circumstance; (2) improved project planning through the use of an objective and quantitative basis for decision making; (3) enhanced project execution and control through PSIM's ability to quickly evaluate alternative responses to unplanned events; (4) a deeper understanding of the many factors that influence success for complex software development projects; (5) enhanced ability to communicate process choices and alternatives; (6) an objective way for project managers to answer difficult questions such as "Will adding resources early in the project really reduce overall project cost?".

van der Aalst et al. [1] focused on the behavior of resources and propose a new way of characterizing their availability. Experiments show that it is possible to capture human behavior in the business processes of a much better way. By incorporating best resource characterization tools, business process simulation can finally fulfill his promise slope.

---

[1] Proceedings of International Workshop on Software Process Simulation Modeling (ProSim) and International Conference on Software Process (ICSP), and the special journal issues.

Guizzardi et al. [5] present a Discrete-Event Simulation Ontology that defines the basic concepts that any general purpose DES language should support in order to evaluate DES languages. Then they make a study of how they adapt the basic elements of BPMN models to represent Agent Based Simulation to analyze the following properties: consistency, completeness, clarity and conciseness [6].

Onggo et al. [11] propose the use of BPMN to represent a conceptual model of Agent-Based Simulation. BPMN is widely used by organizations and can serve for business users adopt Agent Based Simulation to design this visual modeling tool called ABMN Designer (Agent-Based Model and Notation Designer).

Cetinkaya et al. [3] proposed a Model Driven Development work environment for Modeling and Simulation (MDD4MS). This environment includes: (1) the life cycle of Modeling and Simulation; (2) the definitions for conceptual metamodel for BPMN what you have selected; (3) the stages of modeling, specification, for which they have chosen DEVS (Discrete-Event System Specification), and implementation; (4) the transformation of metamodels; (5) and a tool architecture for the overall process.

25.3 Problem Description

In this section we describe the context of the problem we want to solve. As discussed in Sect. 25.1, today is very important for organizations to have business processes that are flexible and can be adapted to the dynamic conditions of a global market increasingly competitive. The modeling and simulation of business processes have become an essential tool for scenario analysis and the results would anticipate changes before they occur to support decision-making.

Currently, those responsible for modeling business processes need to have experts in simulation to obtain the corresponding models and carry out the tests because, they do not know simulation tools. This implies that the BP modeler must define any process change, the variables to be adjusted and the elements for analyzing, then the simulation team creates and executes the simulation model. After that, the BP modeler is responsible for interpreting the results to decide how to improve the BP defined. This process is shown in Fig. 25.1a.

This gap between the BP modeler and simulation experts may create inconsistencies between what the BP modeler wants to analyze and how it is interpreted by the technical team, it makes more difficult to create new scenarios for analysis and interpretation of simulation results can be subjective regarding the BP domain.

Ideally, BP modeler could also perform simulation. It is essential to develop tools that enable BP designers: (1) automatically obtain simulation models from BPMN models of their business processes, (2) complete such models with initial values of the simulation parameters and (3) generate the input file for execution on a particular simulation tool.

To this end, we propose a MDE approach [4] to address this problem by following the flow defined in Fig. 25.1b, which we explain in more detail below:

- First, the BP designer must create a model of your business process. This model must conform to the BPMN 2.0 metamodel.
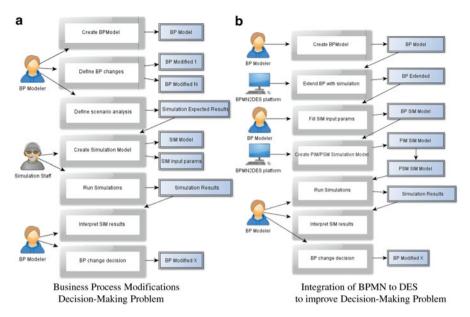
Fig. 25.1 Business process modifications decision-making problem and BPMN to DES integration

- Then the model developed in the previous step will become an extended BPMN model. This model contains all the information of the original model and a set of parameters, initially empty, necessary for process simulation. To automate this task we implement both the Extended BPMN metamodel and the corresponding M2M transformation.
- At this point, the designer must set (manually) the specific values for the parameters of the model generated in the previous step for those who want to carry out the simulation.
- Once the extended BPMN model was completed, it will be transformed into a simulation model independent of the concrete platform on which finally will perform the simulation. To automate this task we implement both the metamodel and the corresponding PIM Simulation M2M transformation.
- Then, using M2T transformations (to be defined once for each considered simulation platform), we will obtain the input files that can be executed in the corresponding simulation tools, for example, Vensim[2] or AnyLogic.[3]
- The simulation results will be interpreted by the BP Modeler to take the appropriate decisions.

In summary, the MDE proposed approach contains a number of metamodels and transformations between them, as shown in Fig. 25.2.

[2] http://www.vensim.com
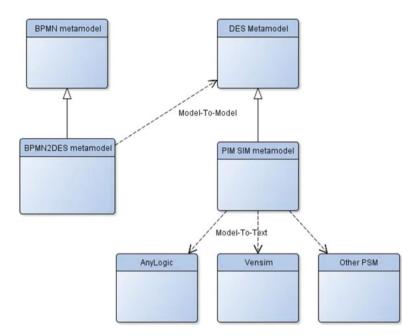[3] http://www.xjtek.com

Fig. 25.2 MDE approach from BPMN to DES

25.4 Discrete-Event Simulation

A *model* is an abstraction or a simplified representation of a complex system that can be real or conceptual. A model is designed to show certain characteristics of the system that you want to study, predict, modify or control. Therefore, a model includes some aspects, but not all, of the system to be analyzed.

A *simulation model* is a computational model that has the characteristics cited above and which represents a dynamic system. Simulation models offer as main advantage the possibility of experiencing different decisions and analyze their results in systems where the cost or risk of actual experimentation are prohibitive. Furthermore, the simulation allows the analysis of systems of such high complexity that are impossible to represent by static models.

The common goal of simulation models is to provide mechanisms for experimentation, prediction of behavior, resolution of questions of the type *What would it happen if…?* and learning of the system represented, among others [3].

A *Discrete-Event Simulation Model* is one in which the model state changes occur at discrete points in time, which are the events [8]. Measuring the time of the simulation is given in units of time, appropriate for the system in question.

The main components involved in DES are [2]:

- *Entities*: they are the objects that move, change state and interact with others through the system. Some are permanent, they are found always in the system, and temporary staying for a limited time.

- *Attributes*: they identify entities, they include a number indicating the time of the next event.
- *Variables*: they reflect the characteristics of the system as a whole. They are not associated with entities, although they may be modified by them. they can be predefined by the simulation software or be set by the user.
- *Resources*: they are a special type of entity used by other entities to perform an action. An entity requests a resource. When assigned uses and releases it later.
- *Queues*: they are the place where entities expect when they can move, for example, because the resources needed are not available.
- *Activities*: they are functions that resources perform over entities. Any activity can have a defined duration, empirical or stochastic, but can also be defined with a fictional length. In each often be quoted an entity with one or more resources. The tasks performed in each process would be our system activities.
- *Events*: they are facts that occur in a given time, and which rise to changes in the state of a system entity. They may be endogenous, if given by conditions of the model, or exogenous, if the causes are external. Keep in mind that during an activity can not be given any event.

25.5 DES Simulation Metamodel

In order to develop the proposed MDE approach, as we have shown in Fig. 25.2, it is necessary to have a Platform Independent DES metamodel. In this paper we propose the metamodel presented in Fig. 25.3.

A model consists of *Parameters*, *Variables*, *Events*, *Institutions*, *Activities* and *Transitions*. On one hand *Entities* are related to *Events* that can cause a change of state in the same as a trigger condition, and the *Activities* that will act on them performing a particular action (i.e. create, add to the queue or destroy) expressed through the action link attribute. The *Event* may occur when given a certain condition and activation value, which are its attributes.

On the other hand, *Activities* that act on the *Entities* can be of three types: *initial*, *final* and *intermediate*. The *initial* is what starts the process simulation using an activation function and a value, the *final* contains an attribute that collects information in order to end the simulation and the *intermediate* activities are those that allow the process to move forward and keep your life in that attribute. *Transition* class allows passage from one *Activity* to another, and subclass represents the alternative decision between two *Activities*, depending on the evaluation of a condition. Any *Transition* has an activity of *origin* and a *destination*.

Figure 25.4a shows the graphical representation of the elements which form a pattern of discrete-event simulation for modeling according to the defined metamodel. Once we have defined the metamodel, it is necessary to establish a model-to-text transformation scheme in order to obtain the files that serve as input to a simulation tool. Figure 25.4b shows the relationship between our DES metamodel elements and basic elements of AnyLogicTM Enterprise Library tool for discrete-event simulation.
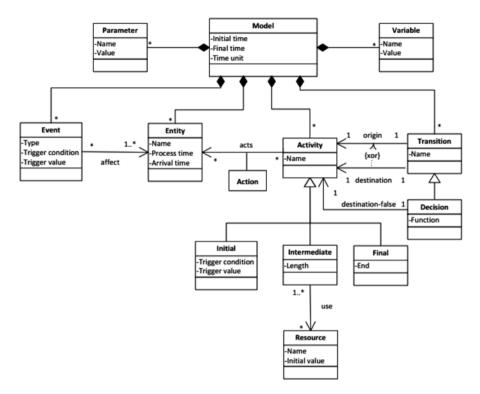
Fig. 25.3 DES metamodel. (**a**) DES elements model graphic representation. (**b**) Metamodel transformation from DES elements to AnyLogic

a DES elements model graphic representation

b Metamodel transformation from DES elements to AnyLogic™

Fig. 25.4 DES model graphic representation and M2T transformation to AnyLogic

25.6 Case Study

To validate our proposal, we propose a simulation model for customer service process in a bank branch. Customers are coming to the branch and, on arrival, they decide if they want to perform the operation that requires the intervention of an employee of the branch or can do it themselves in the Automated Teller Machine (ATM). If they decide to use the ATM, there may already be another client using it, so they must wait in line. Conversely, if required attention window, head to one of the windows that are free or, if all are busy, wait in line for their turn.

Having a simulation model of a system like this allows management of the branch test different decisions regarding the allocation of staff to the windows depending on the demand at different times of the working day and analyze the effects of these decisions toward implementing a solution that optimizes the results.

The input parameters for the simulation of this process are set to the following values:

- Arrival rate of people in the office: 1.7 persons/minute.
- Average shooting time cashier: Triangular (0.5, 1, 1.5).
- Probability of using ATM: 50%.
- Number of windows: 4.

The discrete-event simulation model for this example, compliant with our DES metamodel, is shown in Fig. 25.5.

Following the model transformation scheme that we have presented in Fig. 25.4b, we obtain the simulation model specifically for AnyLogicTM shown in Fig. 25.6.

The simulation of DES model with the initial parameters presented above, would give the results as shown in Fig. 25.7. In view of these results, the business process modeler can make decisions, to do that, just run different simulations, with the only effort to make the business process and change, through the interface, the initial parameter values.
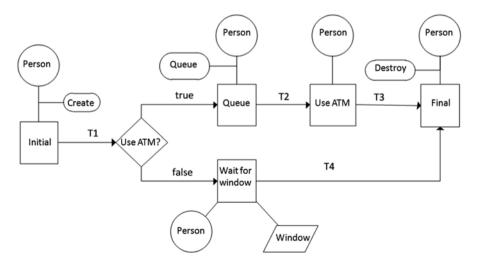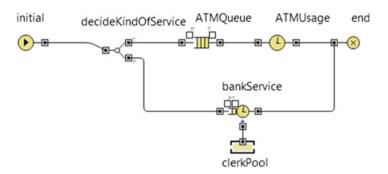

Fig. 25.5 DES simulation model

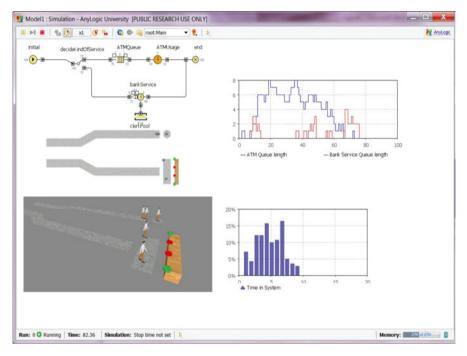**Fig. 25.6** AnyLogicTM simulation model

Fig. 25.7 Simulation results

25.7 Conclusions and Future Work

Simulation models offer the opportunity to experience different decisions and analyze their results in systems where the cost or risk of actual experimentation are prohibitive. They also favor the study and decreased risks and advise management on the strategic, tactical and operational, therefore, they can be considered as a very important tool in business to facilitate decision making at the time of define and model business processes.

In this paper we proposed a metamodel in order to, from a BPMN 2.0 based description of business processes, automatically generate the simulation environment that can be executed using any of the existing simulation platforms. Our proposal focuses on reducing the gap between the team that defines the processes and technical staff that supports the simulation infrastructure.

The case study allowed us to conceptually validate the proposal, so that our future research will focus on: (1) generating a user interface that facilitates the designer configuring the simulation parameters; (2) making use of MDE by transformation rules for model-to-text (M2T) automatically generating the specific language model for various simulation platforms (3) developing the diverse metamodels and transformations model-to-model (M2M) which must provide an environment that integrates BPMN with DES following a MDE approach (4) approving the proposal with empirical studies and case studies in the industry.

References

1. van der Aalst W, Nakatumba J, Rozinat A, Russell N (2008) Business process simulation: how to get it right. BPM Center Report BPM-08-07, BPMcenter.org
2. Banks J, Carson JS, Nelson BL, Nicol DM et al (2005) Discrete-event system simulation
3. Cetinkaya D, Verbraeck A, Seck MD (2011) Mdd4ms: a model driven development framework for modeling and simulation. In: Proceedings of the 2011 summer computer simulation conference, Society for Modeling & Simulation International, pp 113–121
4. Garcia MT, Ruiz MVCC (2012) Como usar mde para obtener un modelo de simulacion a partir de un modelo de proceso de negocio. Actas de las Jornadas de Ingenieria del Software y Bases de Datos 2012 (JISBD 2012)
5. Guizzardi, G, Wagner G (2010) Towards an ontological foundation of discrete event simulation. In: Simulation conference (WSC), Proceedings of the 2010 Winter, IEEE, pp 652–664
6. Guizzardi G, Wagner G (2011) Can BPMN be used for making simulation models? In: Barjis J, Eldabi T, Gupta A (eds) Enterprise and organizational modeling and simulation. Springer, New York, pp 100–115
7. Kellner MI, Madachy RJ, Raffo DM (1999) Software process simulation modeling: why? what? how? J Syst Software 46(2):91–105
8. Law A, Kelton W (2000) Simulation modelling and analysis. 3a edicion. McGraw Hill. Boston 2000
9. Model BP (2004) Notation (BPMN) version 1.0. BPMI Notation Working Group. http://www.bpmn.org. Accessed April 2013
10. Model BP (2011) Notation (BPMN) version 2.0. OMG specification, Object Management Group. http://www.omg.org/spec/BPMN/2.0/PDF. Accessed April 2013
11. Onggo BS, Karpat O (2011) Agent-based conceptual model representation using BPMN. In: Proceedings of the Winter simulation conference, pp. 671–682. Winter Simulation Conference
12. Raffo DM, Wakeland W (2008) Moving up the CMMI capability and maturity levels using simulation
13. Weske M (2012) Business process management. Springer, Berlin
14. Zhang H, Kitchenham B, Pfahl D (2008) Software process simulation modeling: facts, trends and directions. In: Software engineering conference, 2008. APSEC'08. 15th Asia-Pacific, IEEE, pp 59–66