Hybrid Business Process Simulation: Updating Detailed Process Simulation Models Using High-level Simulations*

Mahsa Pourbafrani and Wil M. P. van der Aalst

Chair of Process and Data Science, RWTH Aachen University, Germany
{mahsa.bafrani,wvdaalst}@pads.rwth-aachen.de

Abstract. Process mining techniques transfer historical data of organizations into knowledge for the purpose of process improvement. Most of the existing process mining techniques are "backward-looking" and provide insights w.r.t. historical event data. Foreseeing the future of processes and capturing the effect of changes without applying them to the real processes are of high importance. Current simulation techniques that benefit from process mining insights are either at detailed levels, e.g., *Discrete Event Simulation* (DES), or at aggregated levels, e.g., *System Dynamics* (SD). System dynamics represents processes at a higher degree of aggregation and account for the influence of external factors on the process. In this paper, we propose an approach for simulating business processes that combines both types of data-driven simulation techniques to generate holistic simulation models of processes. These techniques replicate processes at various levels and for different purposes, yet they both present the same process. SD models are used for strategical what-if analysis, whereas DES models are used for operational what-if analysis. It is critical to consider the effect of strategical decisions on detailed processes. We introduce a framework integrating these two simulation models, as well as a proof of concept to demonstrate the approach in practice.

Keywords: Process mining · Discrete event simulation · Hybrid process simulation · Scenario-based predictions · System dynamics

1 Introduction

After bringing transparency into processes, the process mining mission is to find data-supported ways to improve the processes in different aspects, e.g., performance metrics. In [2], the capability of process mining techniques to design realistic simulation models is discussed. Process mining supports designing the simulation models by capturing all the aspects of the process in detail. However, some influential factors remain undiscovered. These undiscovered factors eventually affect the simulation results. For instance, the efficiency of resources or

*Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy-EXC-2023 Internet of Production - 390621612. We also thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

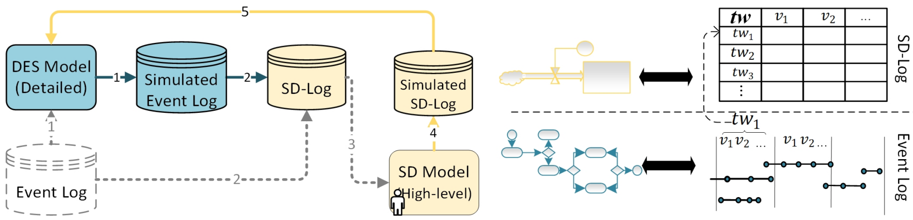

Fig. 1: General idea of designing comprehensive simulation models for business processes. We update the detailed simulations (DES) by the aggregated simulation (SD) results to take the high-level/strategical effects into account. The dotted arrows represent the optional steps in the approach, e.g., SD models can be given as inputs or can be derived from SD-Logs (left). The different aggregated levels of data used in or generated by DES and SD simulations (right).

the effect of workload on the speed of resources is not taken into account when a discrete event simulation model for a process is designed.

The presented approach in [11] is based on using *process mining* and *System Dynamics* (SD) to tackle these types of problems. System dynamics techniques model a system and its boundary, i.e., environmental variables which influence the system and capture these influences over steps of time. The advantage of this approach is that the process variables are designed based on event logs at higher levels, i.e., not a matter of single instances. For instance, the average waiting time of customers in the process per day has a more significant influence on the number of allocated resources per day than a long waiting time of a single customer. It should be noted that high-level simulation techniques such

Table 1: The general comparison of Discrete Event Simulation (DES) and System Dynamics (SD) techniques in process mining.

	DES in PM	**SD in PM**
Goal	– Detailed simulation of processes – Mimicking processes – Operational	– High-level simulation of processes – Policy and decision-making – Strategical
Usage	Operational	Strategical
Data	Detailed event logs	– Coarse-grained process log: – aggregated process variables over time
Simulation step	Events	Time steps, e.g., day
Weakness	Not capturing external factors	Evaluation of results

as SD ignore the provided detail, which improves the accuracy of the simulation results to some extent. On the contrary, detailed simulations in process mining, i.e., *Discrete Event Simulation* (DES), lack the high-level effects of process variables on each other as well as quality-based variables, e.g., the effect of tiredness of resources on the execution time of cases. The overview of the comparison of DES and SD techniques in process mining simulation is shown in Table 1. Note that standard event logs and aggregated process variables (higher level logs) are referred to as detailed event logs and coarse-grained process logs, respectively.

Figure 1 (right) depicts the data granularity level in process mining for both designing and re-generating simulation results, i.e., events in DESs are transformed into aggregated process variables such as v_1 and v_2 at each time step tw in SDs.

The raised concerns for simulation in process mining as mentioned specifically for discrete event simulation in [1] show that only DES models in process mining for simulating processes are not sufficient. A comprehensive business process simulation should be able to exploit the detailed process steps, i.e., workflow and resources for every single case, and the strategical perspective, and external factors at the same time. DES and SD are at different levels and for different purposes, yet, represent the same process with two views. Therefore, exploiting these two techniques in process mining makes designing a comprehensive simulation of a business process possible. The direction of interaction between these two techniques is based on business processes and scenarios [5]. The direction can be one model updating the second model or both models updating each other, resulting in bi-directional interaction.

In this paper, we propose a framework to generate comprehensive simulation models for business processes. The framework aims to combine the advantages of both simulation techniques as shown in Fig. 1. Using an event log of a process that also can be generated by the process DES model, we extract possible aggregated process variables, e.g., average arrival rate per day. The event log can be achieved by a DES model of the business process using approaches such as [16]. SD models are designed based on the generated coarse-grained process logs, i.e., SD-Logs, out of event logs [13]. The preprocessing step of our framework is generating an event log, the corresponding SD-Log, and two models at different levels. Using the provided input from the preprocessing step, we design a method to define and discover the overlapping variables between two process simulation models since they are at different levels. The transformation phase to update the DES model using the updated variable from the SD model is the critical step of our approach, in which we use the designed detail simulation models in process mining and a list of possible variables in an SD-Log. For instance, the efficiency of resources, in reality, is not 100%, therefore, using an SD model, we can incorporate the effect of tiredness, workload per day on the resources' efficiency, or the effect of their expertise and then update the resource service time in the DES model. With the continuation of DES execution with the updated variables, a new event log is generated, which includes the simulated effect from the SD model.

The remainder of this paper is organized as follows. In Section 2, we present related work. In Section 3, we introduce background concepts and notation. In Section 4, we present the proposed approach, which we evaluate in Section 5. Section 7 concludes this work.

2 Related Work

Employing *Discrete Event Simulation* (DES) in process mining is a common approach to simulate business processes. The provided insights by process mining help the traditional business process simulation techniques to generate more accurate results based on the history of the business processes. In [16], all the aspects of a process from control-flow, organizational, performance status, and decision points are discovered from process mining techniques and considered in designing the simulation model. *Colored Petri Net* (CPN) models help in capturing both the activity-flow of processes as well as other aspects. CPN Tools [14] offers a platform for designing and simulating the CPN models. In [15], an approach for the automatic generation of CPN models for running on the CPN Tools is presented. Other approaches in the area of business process simulations exploit the process model based on the BPMN notation and improve the quality of the models using the provided information on the corresponding event logs [4].

Conducting high-level simulation models for processes is proposed in [11]. The feature extraction from event logs with a given period of time, e.g., one day, is performed before designing a model and the exploited modeling technique is System Dynamics (SD). The variables are captured with their relations and the generated model is used for simulating the process on the given time window.

These two approaches have weaknesses that can be covered by each other [6]. Using only aggregated level modeling, the details of process instances are neglected. At the same time, in detailed simulation techniques, external factors, as well as aggregated influences are ignored. In different simulation areas, the combination of DES and SD are exploited [7]. To connect two simulation models, three directions are specified [5]: (1) DES results update SD model, (2) SD results update the DES before simulation, and (3) both update each other in different phases. According to [3], the first sort of interaction is more common. It is usually more important to capture the influence of high-level decisions on detailed systems. In [18], the combination is used to perform simulation for a case study in the healthcare area where the number of newly infected patients is predicted using SD and inserted into the DES model of the serving patients in the hospital.

In process mining, both simulation techniques are proposed to be used individually on processes based on event logs. In our approach, we propose to exploit high-level simulation results for strategical scenarios and consider detailed simulation modeling of processes. These two types of modeling are supported by event data and the generated models are valid due to the existence of the previous executions of processes, event logs. To the best of our knowledge, this is the first time that both high-level and detailed simulation techniques for business processes are taken into account together.

3 Preliminaries

In this section, we define the concepts and notations used in process mining and system dynamics simulation including coarse-grained process logs, i.e., aggre-

Table 2: A part of an event log for a sample process inside a hospital. Each row represents a unique event indicating a specific case ID, activity, resources, and timestamps.

Case ID	Activity	Start timestamp	Complete timestamp	Resource
154	registration	01.01.2021 11:45:00	01.01.2021 11:57:00	resource 1
155	admission to the ward	01.01.2021 11:57:10	01.01.2021 12:40:52	resource 1
154	registration	01.01.2021 11:47:17	01.01.2021 12:05:01	resource 2
156	registration	01.01.2021 12:51:23	01.01.2021 13:02:47	resource 1
⋮	⋮	⋮	⋮	⋮

gated process variables over time. Since these coarse-grained logs are utilized for SD simulation, they are referred to as SD-Logs.

3.1 Process Mining

The stored event data of processes in the form of event logs are used for process mining techniques. The form of logs which we use in our approach is defined in Definition 1.

Definition 1 (Event Log). *Let $\mathcal{C}$ be the universe of cases, $\mathcal{A}$ be the universe of activities, $\mathcal{R}$ be the universe of resources and $\mathcal{T}$ be the universe of timestamps. We call $\xi=\mathcal{C} \times \mathcal{A} \times \mathcal{R} \times \mathcal{T} \times \mathcal{T}$ the universe of events. The event e is a tuple $e=(c, a, r, t_s, t_c)$, where $c\in\mathcal{C}$ is the case identifier, $a\in\mathcal{A}$ is the corresponding activity for the event e, $r\in\mathcal{R}$ is the resource, $t_s\in\mathcal{T}$ is the start time, and $t_c\in\mathcal{T}$ is the complete time of the event e, where $t_s \leq t_c$. We assume that events are unique and an event log L is a set of events, i.e., $L\subseteq\xi$.*

We also define projection functions, $\pi_{\mathcal{C}}: \xi \rightarrow \mathcal{C}$, $\pi_{\mathcal{A}}: \xi \rightarrow \mathcal{A}$, $\pi_{\mathcal{R}}: \xi \rightarrow \mathcal{R}$, $\pi_{\mathcal{T}_S}: \xi \rightarrow \mathcal{T}$ and $\pi_{\mathcal{T}_C}: \xi \rightarrow \mathcal{T}$ for attributes of events. A sequence of events w.r.t. timestamp with the same case identifier represents a process instance (trace). Consider Table 2 where the first row is the event $e=(c, a, r, t_s, t_c)$ for the patients with case ID "154" as c, the activity "admission to the ward" as a which was started at timestamp "01.01.2021 11:45:00" as t_s by resource "resource 1" as r and was completed at timestamp "01.01.2021 11:57:00" as t_c.

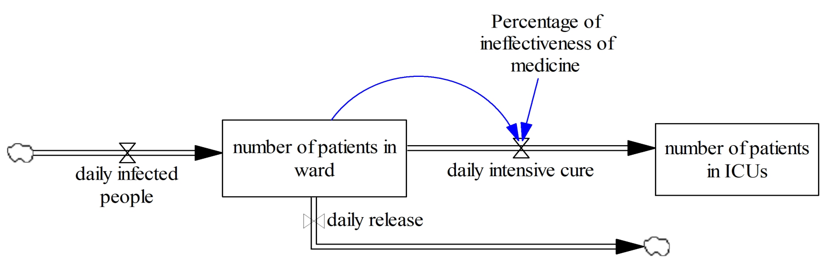

Fig. 2: A simple example stock-flow diagram.

Table 3: A part of an SD-Log for the sample hospital event log. The columns represent the process variables and the rows represent the steps.

Time Window Daily	Arrival rate	Finish rate (release rate)	Avg service time (time in the hospital)	Avg waiting time in process
1	180	180	0.3590	0.6099
2	147	140	0.4156	0.5409
3	160	162	0.4011	0.5961
4	116	119	0.4455	0.4908
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

3.2 System Dynamics

System dynamics techniques model complex systems and their environment at a higher level of aggregation over time [17]. The *causal-loop diagram* and the *stock-flow diagram* are the essential diagrams in system dynamics by which the relations between all the influential factors in/outside a system in both conceptual and mathematical ways are represented, respectively [17]. Figure 2 shows a simple stock-flow diagram where the *number of patients in the ward* in each day is calculated as follows: $number\ of\ patients\ in\ the\ ward = number\ of\ patients\ alreay\ in\ the\ ward + today\ infected\ people - todaty\ intensive\ cure - today\ release.$ The values of stock-flow elements get updated in each step based on the current/previous values of the other elements that influence them.

SD-Logs In order to design system dynamics models for processes, event logs should be transformed into the SD-Logs. SD-Logs are required for generating simulation models as well as populating them using the values of variables with the purpose of validation [9].

Definition 2 (SD-Log). *Let $L \subseteq \xi$ be an event log, let $\mathcal{V}$ be a set of process variables, and let $\delta \in \mathbb{N}$ be the selected time window. An SD-Log of L, given δ, $sd_{L,\delta}$ is $sd_{L,\delta} \in \{1, \ldots, k\} \times \mathcal{V} \to \mathbb{R}$, s.t., $sd_{L,\delta}(i, v)$ represents the value of process variable $v \in \mathcal{V}$ in the i^{th}-time window ($1 \leq i \leq k$) where $k = \lceil \frac{(p_c(L) - p_s(L))}{\delta} \rceil$.*

If L and δ are clear from the context, we exclude them from the generated SD-Logs and write *sd* instead. Definition 2 also indicates the format of generated outcomes as an SD model simulation in which the values of the variables in the simulation are generated. Table 3 is a part of a sample SD-Log which shows the generated SD-Log with $\delta = 1$ *day* that includes different process variables, e.g., in the first time window (day) 180 cases arrived at the process.

4 Approach

Figure 3 represents the framework including three main components: DES, SD simulation, and developing and updating interface variables. Components (1) and (2) each have two steps: discovering/designing simulation models and executing the discovered simulation models. Given an event log, and the discovered DES simulation model, the process can be simulated (1). The generated event log is inserted into the SD-Log generator and the output is used to populate the SD model (2). Having both models populated with the data and ready to run, it is time to design the connection to update DES based on SD results (3). We use DES models in the form of *Colored Petri Net* (CPN) models. In the second component, event logs are transformed automatically into multiple variables describing the process (SD-Logs) over a specific period of time, e.g., per day, as introduced in [9].

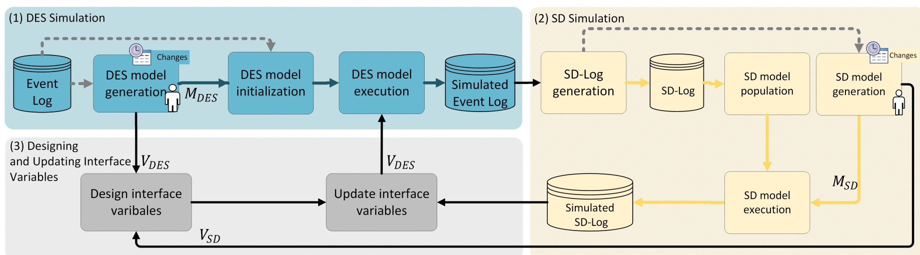

Fig. 3: The framework starts with the design and simulation of DES models (M_{DES}) and generates an event log (1). The event log is transformed into SD-Logs for generating/populating the SD model (M_{SD}) (2). Possible interaction interfaces between two models are discovered (3), e.g., simulation parameters in both models ($V_{DES} \cap V_{SD}$). Then, the detailed simulation model parameters for execution (V_{DES}) get updated by the results of the high-level simulation model. Dashed lines indicate the optional steps in designing DES and SD models.

To systematically address the connection between two models, we consider designing detailed simulation models based on the process mining insights. Furthermore, we define and extract a collection of possible variables for designing high-level simulation models from an event log [9]. The next step is to use the provided framework to update the interface variables, i.e., variables that exist in both detailed and high-level models, see Fig. 3 (part 3). Consider that the target scenario is to measure the influence of advertising investment on the acquisition rate of new customers (cases) in the process in two months. The DES model used to generate the event log is designed to simulate a specific number of cases per day. The corresponding SD model is developed based on the event log and the relevant scenario, and the new arrival rate value is predicted, e.g., as a result of viral marketing or effects of billboard sites. When the new DES model is run with the new arrival rate, the event log is updated. The updated event log clearly reveals whether the process is capable of handling the additional cases in terms of resources.

4.1 DES Simulation

Simulation parameters such as arrival rate and average service duration of various activities for regenerating the process should be initialized before running a DES model. As a result, the first component considers the DES model discovery and execution steps separately.

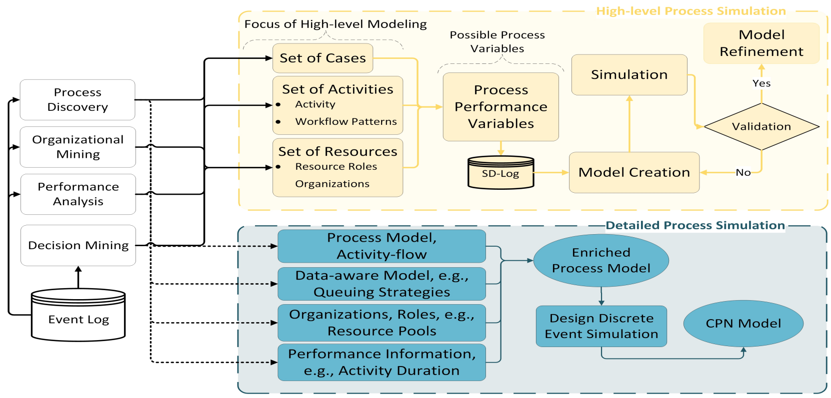

Fig. 4: Using process mining to generate simulation models of processes at different levels. Detailed process simulation models (bottom) include process activities, their performance metrics, resources, and all the possible design choices for handling the cases in the simulation, e.g., queue strategies. Our proposed framework for using process mining and system dynamics together in order to design valid models to support the scenario-based prediction of business processes at higher levels (up). Discovery and conformance techniques in process mining provide insights into the processes in different aspects, e.g., a set of activities, which are the potential design choices for aggregated simulation models.

4.1.1 Designing DES Model The process of designing a detailed simulation model of a process using process mining is started with discovering a process model and enriching that with other aspects. For instance, for the simulation environment, often the arrival process is sampled from a *negative-exponential* distribution. To capture possible executable aspects of processes, we design process simulations based on the process mining insights as shown in Fig. 4 (bottom). In this work, we consider the designing process of the DES simulation model starting from an event log or designed based on the highlighted parameters in Fig. 4 by domain knowledge.

4.1.2 Executing DES Model While simulating a process, all the mentioned aspects in the process simulation model can be changed for simulating different scenarios. Parameters such as the arrival rate function and performance parameters of activities in the process such as duration, number of resources can be changed as well as changing the serving queuing strategy and the flow of activities for applying different scenarios on the process. The designed DES simulation using process mining enables us to discover the change points in the process which can be updated by high-level simulation models of the process. In the DES execution step, the parameters require to be initialized. We refer to the set of simulation parameters in a DES model as V_{DES}.

Definition 3 (DES Simulate). *Let ξ be the universe of events, $n \in \mathbb{N}$, and $\mathcal{M}_{DES}$ be the universe of discrete event simulation models. $Sim_{DES} : \mathcal{M}_{DES} \times \mathbb{N} \to 2^{\xi}$. Given simulation model m_{DES}, its set of initial values of paramters V_{DES}, the specified period of time n and the start time of the simulation, $Sim_{DES}(m_{DES}^{V_{DES}}, n) = L \subseteq \xi$ simulates the process.*

Table 4: List of possible process variables generated from coarse-grained event logs [9]. The variables can be generated at different levels, e.g., the whole process or single activity level. The table shows the possibility of applying different aggregation functions (AF) on top of the performance indicators (IN) for different aspects (AS). The valid combinations provide process features, which along with the selected design choices form process variables [9].

Validator	IN												
	Value	Count					Service time			Waiting time			Time in process
AS / AF	Numerical variable	Categorical variable	Numerical variable	Case	Resource	Activity	Case	Resource	Activity	Case	Resource	Activity	Case
Sum	True	False	True	False	False	False	True	True	True	True	True	True	True
Average	True	False	True	False	False	False	True	True	True	True	True	True	True
Median	True	False	True	False	False	False	True	True	True	True	True	True	True
Null	False	True	False	True	True	True	False	False	False	False	False	False	False

Function Sim_{DES} illustrates the simulation process for a given DES model of a process at an abstract level. Note that, in the approach, the KPIs are measured over simulated event logs in a specific period of time. Therefore, in Sim_{DES}, we consider the simulation duration, e.g., one day, to be a given input by the user.

4.2 SD Simulation

The second component aims to deliver data-driven SD simulations of processes in order to integrate detailed and high-level simulation. To accomplish this, we use event logs of processes to extract a number of performance parameters from the current state of the process and provide an interactive platform for modeling the performance metrics as system dynamics models. The models that are built can address *what-if* queries.

4.2.1 Designing SD Model The advantage of the introduced approach in [11] for generating high-level simulation models is that the variables, i.e., simulation elements, are directly generated based on real values and can be validated. The relations that form the models are also supported by the detected behavior in event logs as shown in Fig. 4 (top). To define aggregated process variables over steps of time on the specified part of the process for simulation, the performance indicators (IN), process aspects (AS), and aggregation functions (AF) are required. The list of possible process variables given the three criteria for the selected focus of simulation is determined using the valid combinations in Table 4. For instance, the average (AF) number (IN) of resources (AS) is a process variable that can be measured over steps of time, e.g., daily. The provided list will eventually be used to facilitate the integration phase of two simulation models for determining interface variables for updates. We refer to the generated values of the possible process variables ($\mathcal{V}$) over time steps as SD-Logs (Definition 2).

Definition 4 (SD-Log Generation). *Let* $L \subseteq \xi$ *be an event log,* $\delta \in \mathbb{N}$ *be the selected time window, and* $\mathcal{L}_{SD}$ *be the universe of SD-Logs defined in Definition 2.* $sdGen : 2^{\xi} \times \mathbb{N} \rightarrow \mathcal{L}_{SD}$*, such that, for the given* L *and* δ*,* $sdGen(L, \delta) = sd_{L,\delta}$ *generates the corresponding* $sd_{L,\delta} \in \mathcal{L}_{SD}$*.*

In Definition 4, we define a function to generate SD-Logs based on event logs. Given event log L and time window δ, function *sdGen* generates the corresponding SD-Log $sd_{L,\delta}$. The size of the time window used to generate the SD-Logs is critical. In [12], multiple time series models are trained, and the SD-Logs are generated using the one with the smallest error.

4.2.2 Executing SD Model The SD models are designed with the help of extracted SD-Logs along with users and high-level target scenarios. For a process, SD simulation is performed for the given time step (δ) using function Sim_{SD}. The generated SD-Log ($sd_{L,\delta}$) of the process and the designed SD model (m_{SD}) are the inputs. The set of simulation variables in an SD model is referred to as V_{SD}. The future values of variables are produced in the form of SD-Logs as a result of SD simulations.

The set of SD simulation variables (V_{SD}) can include all or some of the process variables ($\mathcal{V}$) in the generated SD-Log from an event log, as well as a set of external variables (V_{EX}), i.e., $V_{SD} \subseteq \mathcal{V} \cup V_{EX}$. The SD models should be populated with the values and equations to be executable and generate simulation results in the form of SD-Logs. Consider the model in Fig. 2, where variable *daily infected people* should be populated, e.g., from the SD-Log, in order to create future values of variable the *number of patients in the ward* over time. It should be noted that the *number of patients in the ward* in the model is derived using an equation and does not need to be directly supplied. For variable $v \in V_{SD} \cap \mathcal{V}$, there are multiple possibilities of initializing and populating the SD model. For instance, for every simulation step, the value of v is taken from the corresponding row in the SD-Log, or generated by a random generator function. The random generator function is based on the distribution of values of variable v over time.

Definition 5 (SD Simulate). *Let $\mathcal{L}_{SD}$ be the universe of SD-Logs, $\mathcal{M}_{SD}$ be the universe of system dynamics models, and $j \in \mathbb{N}$ be the number of time steps. $Sim_{SD} : \mathcal{L}_{SD} \times \mathcal{M}_{SD} \times \mathbb{N} \rightarrow \mathcal{L}_{SD}$. For instance, $Sim_{SD}(sd, m_{SD}, j){=}sd^{'}{\in}\mathcal{L}_{SD}$ simulates the given $m_{SD} \in \mathcal{M}_{SD}$ over j time steps using the provided values in the sd and the simulation result is represented as an SD-Log (sd').*

The defined *sdGen* and Sim_{SD} enable the main steps in simulating the process models at higher levels for the focused parts of the processes as the targets of high-level simulation. These functions are used later in our framework for integrating high-level simulations and detailed simulations for business processes.

4.3 Designing and Updating the Interface Variables

The activity flow of a process, duration of each activity, batching, or queuing strategies can be updated based on high-level decisions derived from simulation models. The provided list of the changeable parameters in the detailed simulation models and the presented process variables in SD models are the baseline of finding the interfaces between these two types of models for interactions. Exploiting the simulation parameters in the detailed simulation models, we discover

Table 5: The sample mapping table for finding the interface variables in SD-Logs and DES parameters, which enables interaction of the two models of processes possible. The table is generalized, e.g., type of cases (categorical and numerical attributes), organizations and type of resources, and activities follow the same mapping table.

<table>
<tr><td colspan="2">DES Process insights</td><td></td><td colspan="2">SD Process Variables</td></tr>
<tr><td colspan="2">Simulation Parameters
Execution Configuration</td><td></td><td colspan="2">Simulation Aspects</td></tr>
<tr><td>Case</td><td>Number of cases
(Case intervals)</td><td>=</td><td>Number of cases</td><td>Case</td></tr>
<tr><td>Activity</td><td>Processing time</td><td>=</td><td>Service Time</td><td>Activity</td></tr>
<tr><td rowspan="2">Resource</td><td>Processing time</td><td>=</td><td>Service Time</td><td rowspan="2">Resource</td></tr>
<tr><td>Number of resources</td><td>=</td><td>Number of resources</td></tr>
</table>

the ones which can be changed or get influenced by external factors or high-level decisions. In DES models of processes, all the shown process mining insights in Fig. 4 are considered to be simulation parameters that can be changed in order to perform different simulation scenarios of the processes. The changes of the DES simulation parameters can be driven from the high-level simulation model of the process, e.g., the flow of activities, the policy of handling the queues in activities, or resources based on the designed SD models. However, our goal is to automate the interaction between the two simulation models. Therefore, we focus on the parameters that can be found directly in the SD-Logs and not rely on the design choices of SD models.

Table 5 shows the overview of a sample interface variables that can be found in both the DES model and the SD model of a process. These parameters and aspects enable the automatic updating of their values in one of the models based on the other. They can get extended w.r.t. designed models and used parameters. To eliminate the development details of the interaction from SD results to the DES model, the process is considered as a general method. The method looks for all the existing variables in the SD simulation results (V_{SD}) which are in the form of SD-Logs. Afterward, it updates the values of the corresponding simulation parameters in the DES model (V_{DES}) with the last values in the simulated SD-Log, i.e., the predicted values of variables in the last simulated steps. For instance, for the variable average service time of resources from *sells department*, i.e., v, a new DES model is generated in which the value of v is replaced by the last values of v in the simulated SD-Log. As a result, if the time window is one day and the SD model is run for 30 days, the value of v is taken from $sd(30, v)$.

Algorithm 1 presents the interaction between two simulation models as described. The algorithm starts with simulating the DES model for a specific period of time with the real values of the simulation parameters from process mining insights. Considering δ as the time window, the simulation duration is derived from $k * \delta$ where k is the number of steps (window of time) for simulation, e.g., $k = 20$ and δ=1 *day*, the simulation duration is 20 days. The simulated event log with the same time window δ is used to generate an SD-Log, which is used to populate the SD model. Note that δ for generating SD-Logs can be different. After the SD model refinement, i.e., adding external factors, the values of interface variables in the DES model are updated by their new values as SD

Algorithm 1: General algorithm of updating process mining detailed simulation based on the changed in the gateway variables.

Input: Detailed process simulation model $m_{DES}^{V_{DES}}$ initialized withe a set of parameters V_{DES}, High-level simulation model m_{SD} and its variables V_{SD}, time window δ, and k the number of time steps
Output: updated event log using the SD results $L^{'}$
$L{=}Sim_{DES}(m_{DES}^{V_{DES}}, k*\delta)$;
$sd{=}sdGen(L,\delta)$;
$sd^{'}{=}Sim_{SD}(m_{SD}, sd, 1*\delta)$;
foreach v *in* $V_{DES} \cap V_{SD}$ **do**
    Set value of v in m_{DES} to be $sd^{'}(k,v))$;
end
Return updated m_{DES} as m'_{DES};
$L^{'}{=}Sim_{DES}(m_{DES}^{'V_{DES}}, k*\delta)$;
return $L^{'}$;

model simulating results. Running the DES model, a new event log is generated in which the KPIs can be measured and the effects of high-level changes can be tracked.

5 Proof of Concept

We designed a scenario to demonstrate the need for hybrid simulation of processes and how to address that using the proposed data-driven approach. A process is designed using CPN Tools in the form of a Colored Petri Net. We simulate the process and generate simulated event logs. The corresponding SD-Logs are extracted from the simulated event logs, and finally, both CPN models and SD models are considered as inputs of the approach. The SD models are used to update the CPN models for the next simulation step. In each step of simulating the process, specific process KPIs are calculated. These KPIs represent the effect of high-level simulation models on the detailed process.

5.1 Implementation

As a proof of concept, the platform for running and updating a DES model of a process (CPN model) based on the results of simulating the corresponding SD model is developed. The platform is in the form of a Jupyter notebook, which includes the instructions for re-running the experiments and performing additional analysis. The designed CPN model, the SML[1] file, the SD-Log, the SD-model, and the python platform are publicly accessible.[2] The supplied tool in [10] enables producing a ready-to-execute CPN model in the CPN Tools from

[1] Standard Machine Language
[2] https://github.com/mbafrani/PMSD/tree/master/HybridSiminPmSharedMaterialforReview

an event log[3] for the purpose of defining different processes and scenarios. The automatically generated models are able to generate event logs following various changes, e.g., incorporating the effects of high-level simulation models as presented in this paper. In addition, given an event log, the presented tool in [8] supports the data-driven SD model generation.

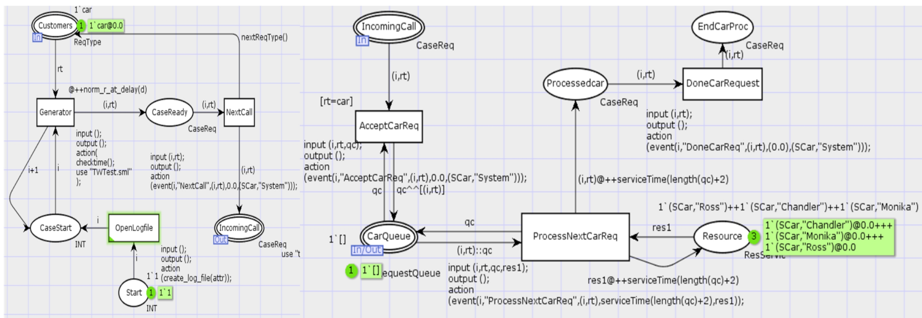

Fig. 5: The designed CPN simulation model using CPN Tools for handling the requests in one of the departments in the sample company.

5.2 Designed Process and Scenario

We created a process within a process mining firm that offers customer support by handling two types of requests in two departments, namely new client inquiries and current customer support. Working days are from Monday to Friday and the working hours are from 9:00 am to 5:00 pm (including 1-hour lunchtime). The process is modeled using the CPN Tools, as partly shown in Fig. 5. On average, every 5 minute, one new request is received by the company. The request arrival is modeled as a negative exponential distribution. The number of requests in the queue is limited to 20 and more upcoming requests will be automatically rejected. The service time spent by the resources in each department for executing requests is derived from a normal distribution. We designed the process model, such that resources perform the process of the request faster if the number of requests in the line is higher. This effect, i.e., the queue length on the processing time of requests, is modeled as an exponential nonlinear relation between the number of people in the queue and the service time.

In the current scenario, the company is looking to increase the number of handled requests, decrease the rejected requests, and have a more realistic simulation of their process. In the detailed simulation model, the resources considered working with full efficiency, e.g., %100 of the available time during the day. It is also considered that all the resources have the same level of expertise, e.g., the same speed in handling requests. One of the potential actions is to increase

[3] https://cpn-model-process-discovery-1.herokuapp.com/generate-cpn-model/

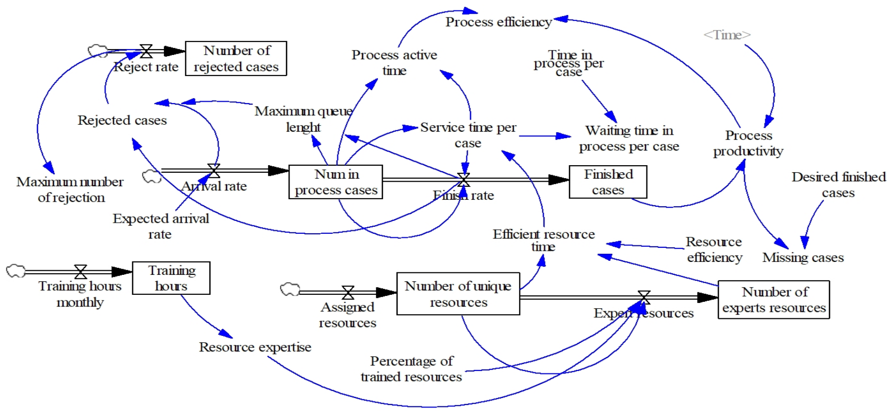

Fig. 6: The designed stock-flow diagram based on the generated event log using the CPN model in Fig. 5. The model extended by capturing the effect of training over three months on the resource efficiency, which shows the actual service time considering their efficiency per day.

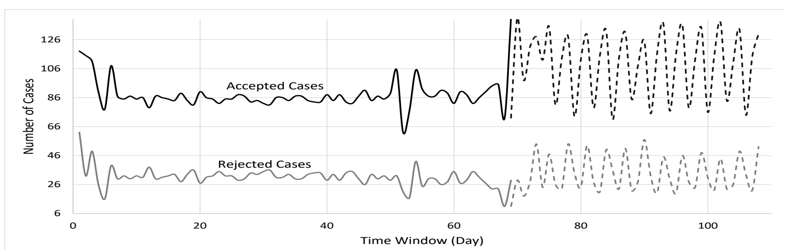

Fig. 7: A part of the process KPIs in 4 months. The number of successfully handled requests, rejected request and average service time of handling the requests by resources for the base CPN model and the updated ones with SD model results, i.e., the dashed lines.

the expertise level of resources by means of training. To capture the effects of training and resource efficiency on the process KPIs, a higher-level simulation model is required. This model should have aggregated time since training requires time to eventually appear in the service time of handling cases, e.g., not all the resources get trained at once.

5.3 Hybrid Simulation of the Sample Process

Running the detailed simulation model for 30 days, we calculate a set of important KPIs, such as the number of addressed requests and rejected requests. Using the generated event log of the CPN model, we generate SD-Log for daily process variables and generate the SD model using the tool presented in [8]. As shown in Fig. 6, the SD model includes the external factors of efficiency and training into account. In this scenario, after a specific amount of training, e.g., 300 hours, the resources become more experts, and they are able to handle more requests in less time, e.g., %20 faster on average. The training hours per month and the percentage of resources that receive the training affect the resources' efficiency, the number of finished and rejected cases over time. To capture these effects on the DES model of the company, we run the SD model for a couple of months, e.g., 4 months. The result of changing the efficiency of resources on the available time and number of handled requests is derived from running the SD model. The SML function, in the CPN model, checks for new updates on the average service time, i.e., the interface variable which is common in two simulation models, before execution. For instance, Equation 1 presents the SML function that reads the values of variables and their function from "CurrentValues.sml" and check if the new updates are available in the updated "UpdatedValues.sml" for the execution. The average service time gets overwritten by the execution of SD models in the corresponding SD-Logs.

$$\begin{array}{l} fun\ checktime()\ = \\ if\ OS.FileSys.compare(OS.FileSys.fileId("CurrentValues.sml"), \\ OS.FileSys.fileId("UpdatedValues.sml"))\ =\ EQUAL \\ then\ use\ "CurrentValues.sml" \\ else\ use\ "UpdatedValues.sml"; \end{array} \tag{1}$$

Using our framework, we update the average duration of handling requests by resources with the new value of service time from the SD model. Variables such as training and expertise of the resources are not easy to be captured and included in the discrete event simulation, i.e., the aggregated timing simulation is required to reflect the effect of changes such as training as well as defining the effect quantitatively is not a straight-forward step. System dynamics modeling enables us to handle such effects in detailed simulation models of processes. As illustrated in Fig. 7, the impact of training on average service time and, eventually, the number of handled cases are obvious after about 2 months. The impact of efficiency and training is considered, resulting in more accurate KPIs.

6 Discussion

The primary goal of this paper is to demonstrate the importance of comprehensive data-driven process simulation modeling at various levels and their interaction for businesses. Furthermore, the potential of creating various simulation models from event logs and applying results of higher-level and what-if analyses on the specific operational process is demonstrated by means of a sample scenario. It should be noted that providing a general framework for automatically combining and executing both DES and SD models at the same time is a challenging task. Moreover, even if the bases of models are built automatically using process mining insights, human domain knowledge is still crucial. We do not focus on eliminating the role of the user in the modeling phase, as it is an essential component of any effective simulation model design, specifically in strategical simulation models, e.g., SD. There are further considerations such as defining and locating interface variables. This issue can also be mitigated using the simulation parameters in DES and the provided set of process variables in

SD-Logs. There are a few steps that will be needed in the future to make the technique more effective. For instance, (1) predefined scenarios that are simple and restricted to the variables in the extracted SD-Logs, or (2) substituting the DES Engine with a simple yet powerful engine that requires less expertise of modeling, e.g., SML programming in this work, will improve the framework. The assessment is a simple version of a real-world scenario with a synthetically constructed process, intended primarily to highlight the necessity and practicality of integrating two simulation-driven processes based on event data.

7 Conclusion

Simulating business processes enables organizations to examine the consequences of changes on their processes without implementing them directly. However, most simulation models are unable to capture reality. Although forward-thinking process mining techniques such as discrete event simulations attempt to address the accuracy issue of simulation models by leveraging process mining insight, there are a few unexpected issues such as the effects of external factors in the process or the role of quality-based variables. In this research, we suggested a strategy that takes advantage of both discrete event simulation approaches in process mining and high-level simulation techniques such as system dynamics simulation to mimic processes at a detailed level while applying high-level decisions. We simulate the processes with both models, and the interplay of the models results in simulation models including detailed and high-level aspects. Using common scenarios in businesses, we demonstrated the use of our technique, including the evaluation of simulation findings.

References

1. van der Aalst, W.M.P.: Business process simulation survival guide. In: Handbook on Business Process Management 1, Introduction, Methods, and Information Systems, 2nd Ed., pp. 337–370. Springer (2015)
2. van der Aalst, W.M.P.: Process mining and simulation: a match made in heaven! In: Computer Simulation Conference. pp. 1–12. ACM Press (2018)
3. Brailsford, S.C., Desai, S.M., Viana, J.: Towards the holy grail: combining system dynamics and discrete-event simulation in healthcare. In: Proceedings of the 2010 winter simulation conference. pp. 2293–2303. IEEE (2010)
4. Camargo, M., Dumas, M., González, O.: Automated discovery of business process simulation models from event logs. Decis. Support Syst. **134**, 113284 (2020)
5. Jovanoski, B., Minovski, R., Voessner, S., Lichtenegger, G.: Combining system dynamics and discrete event simulations - overview of hybrid simulation models. Journal of Applied Engineering Science **10**, 135–142 (01 2012)
6. Morecroft, J., Robinson, S., et al.: Explaining puzzling dynamics: comparing the use of system dynamics and discrete-event simulation. In: Proceedings of the 23rd International Conference of the System Dynamics Society. pp. 17–21. System Dynamics Society Boston, MA (2005)

7. Morgan, J., Belton, V., Howick, S.: Lessons from mixing OR methods in practice: using DES and SD to explore a radiotherapy treatment planning process. Health Systems **5** (06 2016)
8. Pourbafrani, M., van der Aalst, W.M.P.: PMSD: data-driven simulation using system dynamics and process mining. In: Demonstration and Resources Track at BPM 2020 co-located with the 18th International Conference on Business Process Management (BPM 2020). pp. 77–81 (2020), `http://ceur-ws.org/Vol-2673/paperDR03.pdf`
9. Pourbafrani, M., van der Aalst, W.M.P.: Extracting process features from event logs to learn coarse-grained simulation models. In: Advanced Information Systems Engineering - 33rd International Conference, CAiSE 2021, Proceedings. pp. 125–140. Springer (2021). https://doi.org/10.1007/978-3-030-79382-1_8
10. Pourbafrani, M., Balyan, S., Ahmed, M., Chugh, S., van der Aalst, W.M.P.: GenCPN: Automatic generation of CPN models for processes. In: Proceedings of Demonstration Track at ICPM 2021 co-located with 3rd International Conference on Process Mining (2021)
11. Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M.P.: Scenario-based prediction of business processes using system dynamics. In: OTM 2019 Conferences, 2019. pp. 422–439 (2019). https://doi.org/10.1007/978-3-030-33246-4_27
12. Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M.P.: Semi-automated time-granularity detection for data-driven simulation using process mining and system dynamics. In: Conceptual Modeling - 39th International Conference, ER 2020, Proceedings. pp. 77–91. Springer (2020). https://doi.org/10.1007/978-3-030-62522-1_6
13. Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M.P.: Supporting automatic system dynamics model generation for simulation in the context of process mining. In: BIS 2020, Proceedings. pp. 249–263 (2020). https://doi.org/10.1007/978-3-030-53337-3_19
14. Ratzer, A.V., Wells, L., Lassen, H.M., Laursen, M., Qvortrup, J.F., Stissing, M.S., Westergaard, M., Christensen, S., Jensen, K.: CPN tools for editing, simulating, and analysing coloured Petri nets. In: Applications and Theory of Petri Nets 2003, 24th International Conference, ICATPN 2003, Proceedings. pp. 450–462 (2003)
15. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering colored Petri nets from event logs. STTT **10**(1), 57–74 (2008)
16. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering simulation models. Inf. Syst. **34**(3), 305–327 (2009)
17. Sterman, J.D.: Business dynamics: systems thinking and modeling for a complex world. McGraw-Hill (2000)
18. Viana, J., Brailsford, S., Harindra, V., Harper, P.: Combining discrete-event simulation and system dynamics in a healthcare setting: A composite model for chlamydia infection. European Journal of Operational Research **237**(1), 196 – 206 (2014)