Extracting Process Features from Event Logs to Learn Coarse-Grained Simulation Models

Mahsa Pourbafrani(✉) and Wil M. P. van der Aalst

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

Abstract. Most process mining techniques are backward-looking, i.e., event data are used to diagnose performance and compliance problems. The combination of process mining and simulation allows for forward-looking approaches to answer "What if?" questions. However, it is difficult to create fine-grained simulation models that describe the process at the level of individual events and cases in such a way that reality is captured well. Therefore, we propose to use coarse-grained simulation models (e.g., System Dynamics) that simulate processes at a higher abstraction level. Coarse-grained simulation provides two advantages: (1) it is easier to discover models that mimic reality, and (2) it is possible to explore alternative scenarios more easily (e.g., brainstorming on the effectiveness of process interventions). However, this is only possible by bridging the gap between low-level event data and the coarse-grained process data needed to create higher-level simulation models where one simulation step may correspond to a day or week. This paper provides a general approach and corresponding tool support to bridge this gap. We show that we can indeed learn System Dynamics models from standard event data.

Keywords: Process mining · Quantifying processes · Process variable extraction · Scenario-based simulation · System dynamics

1 Introduction

As a business owner, the ability to know the process behavior in different situations is a crucial requirement to improve the process and foresee the upcoming problems. Process mining is a set of data-driven techniques that paves the way to this aim and describes the processes from different aspects [1]. The next step in process mining is to answer the questions regarding the future of processes. Simulation and prediction techniques in process mining are introduced to address this goal [2]. It is possible to perform "what-if" analyses and apply different scenarios on the systems, using fine-grained simulation models that behave close to the real systems. Such models are difficult to create and it is hard to explore alternative scenarios. For example, workers who are involved in multiple processes

M. La Rosa et al. (Eds.): CAiSE 2021, LNCS 12751, pp. 125–140, 2021.
https://doi.org/10.1007/978-3-030-79382-1_8

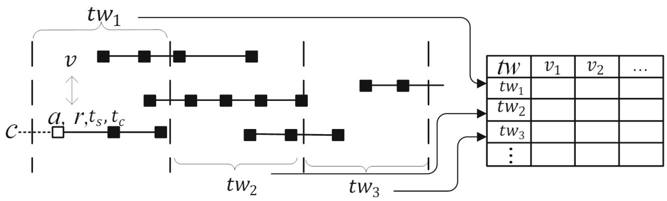

Fig. 1. Transforming fine-grained event logs into the quantitative variables to produce coarse-grained process logs. Time window (tw) indicates the time step, i.e., a specific period of time, and v represents the generated quantitative variable.

may seem under-utilized while being overloaded with work. Different aggregation levels of the states of a process are required for high-level decisions and investigating different scenarios. For instance, the average service time of cases per day plays a more important role than the service time of a single case in deciding on the number of resources to be allocated. As Fig. 1 shows, by looking at event data over a specific period of time, tw, different aspects of the process can be aggregated as process variables such as cases, time-related variables, resources, and activities. The aggregated state of the process and its behavior at that level directly affect every single instance in the process. The resulting coarse-grained process log has a value for each process variable per time window and is used to create a system dynamics model. System dynamics is an aggregated simulation technique that represents a system using the relationships between its variables [15]. System dynamics techniques are able to capture external factors, e.g., the effect of advertisements on the arrival rate of new customers, and simulate the general system without simulating low-level events, e.g., looking at the system at the aggregated level per day instead of taking every single event into consideration. Therefore, unlike traditional discrete event simulations, they are a good match to simulate processes at higher abstraction levels.

In [8], the idea of combining process mining techniques and system dynamics for the purpose of the scenario-based analyses was first presented. In this paper, we propose an approach to extract all the possible measurable aspects of a process systematically for creating coarse-grained process logs. As a result, we can generate default simulation models to be used by system dynamics techniques. The ultimate goal is to bridge the gap between the fine-grained event log and the coarse-grained process log. To do so, we extract *forward-looking* scenarios focusing on the performance aspect w.r.t. the existing attributes in the event log. These questions, i.e., scenarios, are the design choices that come from the process mining insights. For instance, the process shows a bottleneck in an organization, or a long waiting time for a specific part of the process, i.e., a set of activities. We map event logs into the part of the process which we want to focus on and analyze the filtered event logs. We split the filtered logs into the time steps, then we calculate measurable elements over each time step. The remainder of this paper is organized as follows. In Sect. 2, we present the related works. In Sect. 3, we introduce background concepts and notations. In Sect. 4, we present our main approach. We evaluate the approach in Sect. 5 by designing simulation models and Sect. 6 concludes this work.

2 Related Work

Several authors have explored approaches to use simulation in the context of process mining. In [13], the authors introduced an approach to design and generate discrete event simulation models from event logs in the form of Colored Petri Nets including many details such as resource pooling. In [5], the simulations are mainly focused on the activity-flow level presented by Petri nets. Other simulation techniques are based on BPMN models for simulating business processes. In [3], business process simulation including user interaction is proposed.

However, several challenges have not been addressed in the current simulation techniques. In many cases, simulation results are not accurate enough. This is due to the lack of sufficient historical information and not incorporating external factors. The simulation of business processes can be improved by exploiting the event logs and process mining techniques as proposed in [2]. Despite detailed simulation techniques such as discrete event simulation, system dynamics simulation techniques are able to capture a system at a higher level of aggregation as well as affecting the effect of external variables on the system [15]. Techniques such as system dynamics are able to capture external factors and influences. The combination of system dynamics and business processes is proposed in [4]. Authors in [12] mention the possibility of designing system dynamics models for the business processes. However, in the presented work the model generation and simulation are not supported by the data and it is based on the domain knowledge of the process.

The recently proposed approach in [8] introduces the idea of designing system dynamics models using process mining insights. The main goal is to capture the effects of the external variables in the simulation, e.g., the efficiency of users. However, only a proof of concept was provided to show the potential of the combination. Also, one of the applications of the approach, i.e., the production line, is shown in [11]. Furthermore, the extracted values for different variables are exploited to form the models [10]. Besides the hidden relationships between the variables, the granularity of the time step to extract the values highly affects the quality of the simulation results which is addressed in [9] by applying time-series analyses. In this paper, we propose a framework to define, generate and capture all the possible process variables and their quantitative values for answering "what-if" questions in the processes at different levels of aggregation. Our approach addresses designing, extracting, and calculating the required aggregate-simulation variables from event logs based on process mining insights.

3 Preliminaries

In this section, we define process mining and system dynamics concepts and the functions which are used in the proposed approach.

Process mining uses past executions of processes in the form of event logs. An event log captures events which include, case id, timestamps, activity, resource, and other possible attributes.

Table 1. Sample event log of a hospital. Each row is an event. For each unique patient (case) in the process, a specific activity at a specific time is performed by a specific resource.

Case ID	Activity	Age	Start timestamp	Complete timestamp	Resource
116	Registration	28	1/1/2020 10:29	1/1/2020 10:47	John
117	Registration	65	1/1/2020 10:29	1/1/2020 10:29	Sarah
116	First visit	35	1/1/2020 10:30	1/1/2020 10:50	Sam
118	Registration	78	1/1/2020 10:31	1/1/2020 10:49	Sarah
116	Examine	54	1/1/2020 10:31	1/1/2020 10:31	Carl
...	...	...	...	...	...

Definition 1 (Event Log). *An event 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. We call $\xi{=}\mathcal{C} \times \mathcal{A} \times \mathcal{R} \times \mathcal{T} \times \mathcal{T}$ the universe of events. 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. We assume that events are unique and an event log L is a set of events, i.e., $L{\subseteq}\xi$.*

For event log $L \subseteq \xi$, $p_s(L){=}\underset{e \in L}{min}\ \pi_{\mathcal{T}_S}(e)$ and $p_c(L){=}\underset{e \in L}{max}\ \pi_{\mathcal{T}_C}(e)$ return the minimum start timestamp and maximum complete timestamp in L.

A sequence of events with the same case identifier and ordered in time represents a process instance, i.e., a trace.

Definition 2 (Trace). *A trace $\sigma{\in}\xi^*$ is a finite sequence of events $\sigma{=}\langle e_1, ..., e_n\rangle$, where each $e_i{\in}\sigma$ happens at most once and for each $e_i, e_j{\in}\sigma, \pi_C(e_i){=}\pi_C(e_j) \wedge \pi_{\mathcal{T}_S}(e_i){\leq}\pi_{\mathcal{T}_S}(e_j)$, if $i < j$. For $\sigma \in \xi^*$, $\widetilde{\sigma}{=}\{e \in \sigma\}$ is the set of events in σ. We denote $\overline{L}$ as the set of all traces in the event log L.*

For instance, for a patient in an event log of a hospital in Table 1, the first event e represents that for the patient with case id *116* (c), the activity *registration* (a) was started at timestamp *10:29 01.01.2020* (t_s) by resource *John* (r) and was completed at timestamp *10:47 01.01.2020* (t_c). For the same patient, the sequence of events w.r.t. time is called a trace in the process, e.g., the sequence of activities is *registration, first visit, examine, second visit.*

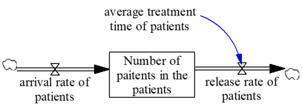
average treatment time of patients
Number of paitents in the patients
arrival rate of patients
release rate of patients

Fig. 2. The value of the stock *number of patients in the hospital* is calculated based on the *arrival rate of patients* and *finish rate of patients* flows (per time step). The value of *finish rate of patients* is affected by the *average treatment time of patients.*

System Dynamics. System dynamics techniques model dynamic systems and their interaction with their environment [16]. The stock-flow diagram is one of the main modeling notations in system dynamics. Systems are modeled w.r.t. three different elements, i.e., stocks, flows, and variables. Stocks are accumulative variables over time, flows manipulate the stock values and variables influence the values of flows and other variables over time. A simple stock-flow diagram for the hospital example is shown in Fig. 2. For instance, the *arrival rate* of the patients and the *release rate* of the patients as flows add/remove to/from the values of the *number of patients in the hospital* as a stock, also, *average treatment time* as a variable affects the release rate. Considering one day as the step of time w.r.t. Figure 2, on average 160 patients, enter the process in the hospital, i.e., the arrival rate, and on average the process takes 8 hours, i.e., average service time. Therefore, simulating the release rate and the number of patients in the hospital per day is possible. The number of patients in the t^{th} day of the simulation is equal to the initial number of patients in the hospital at the beginning of the simulation added by

$$\int_{0}^{t}(+arrival\ rate\ of\ patients - release\ rate\ of\ patients)\mathrm{d}t.$$

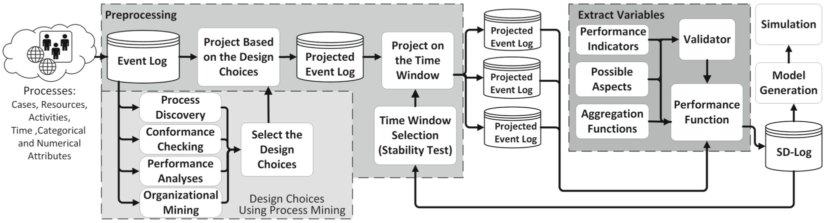

Fig. 3. The main framework to generate possible process variables which describe the process over the steps of time, i.e., SD-Logs. These results are used to form simulation models.

4 Approach

Our approach includes three main modules, as shown in Fig. 3, i.e., applying *Design Choices*, *Preprocessing* step, and *Extract Variables*. Using our approach process behavior is described at different levels. We generate process variables describing the process quantitatively. The transformed process log (SD-Log) provides a coarse-grained view of the observed behavior. These variables are used to generate high-level simulation models to answer what-if questions.

Questions and scenarios are based on the design choices, which are highlighted by the process mining insights. As shown in Fig. 3, process discovery [1], conformance checking [6], performance analysis, and organizational mining [14] results enable designing the simulation scenarios and models. These insights should be quantified in order to be put into action. Based on these results, the focus of the simulation models is either a set of activities, resources, or cases. Therefore, we use the projected event logs on the events including the specified aspects. The focus can be on the whole process, the organizational level, or a part of activity-flow in the process, e.g., workflow pattern structures. For instance, organizational mining shows low efficiency for one of the organizations in the process, therefore, simulation models w.r.t. this knowledge can be designed, e.g., does the resource allocation from the other organization improve the efficiency?

To describe the process over the steps of time, we aggregate the event logs at the time level, i.e., looking at the process in a specific period of time using *Preprocessing* module. The process event log is prepared w.r.t. the design choices from *Design Choice* and the selected time window, the next step is to extract the variables. The *Extract Variables* module defines and calculates possible variables over the steps of time. These variables are the main components of the simulation models for answering what-if questions.

Table 2. Possible design choices for generating simulation models using process mining techniques. *Discovery* and *Conformance Checking* techniques help in selecting a set of activities, resources, organizations, and cases based on the process event log L and the process model M.

PM Techniques	Insights					
	Set of cases	Set of activities		Set of resources		
		Activity	Workflow patterns	Resource	Roles	Organizations
Discovery (L)	+	+	+	+	+	+
Conformance checking (L, M)	+	+	+	+	+	+

4.1 Event Log Preparation

We break down the “forward-looking” analysis into the measurable elements which can be measured over time. We refer to these measurable elements of the scenarios/questions as process variables. These variables are either in the process or in the process environment which some are captured in the event log. To extract possible process variables over time steps, the first step is to form an event log based on the focus of the scenarios/questions and generating different event logs of the process for each time step.

Using the defined *Time Window Projection* and *Design Choices Projection* functions, different levels of what-if analyses are achievable, and the *Performance Function* generates the values of the performance variables.

Design Choices Using Process Mining. The insights provided by the process mining techniques indicate the focus of the modeling. Process discovery, conformance checking, performance analyses, and organizational analyses result in specific parts of a process to be simulated.

Table 2 presents possible insights from different process mining techniques. A set of cases, activities, and resources are possible targets of scenario-based analyses. For instance, for the given example, there is an XOR choice between two activities, *examine* and *radiology* in the process, and the involved activities can be a bottleneck based on the performance analysis and process discovery results, or conformance checking reveals a skipped path for a specific type of cases, e.g., *second visit* is not performed for young patients.

In order to apply the discovered design choices to the simulation model generations, the first step is to use them for process variable extraction. To do so, we define *Design Choice* projection which projects an event log based on the design choices in Table 2. The projected event log includes the corresponding events for the selected insights, e.g., a set of activities.

Definition 3 (Design Choice Projection). *Let ξ be the universe of events and $R{\subseteq}\mathcal{R}$, $C{\subseteq}\mathcal{C}$, and $A{\subseteq}\mathcal{A}$ be the selected sets of resources, cases, and activities, respectively. $DC{\subseteq}2^R{\times}2^C{\times}2^A$ is the universe of design choices. $\Pi_{(R,C,A)} : 2^{\xi} \nrightarrow 2^{\xi}$ is a function that projects a set of events on the given design choice $(R, C, A) \in DC$. For $L{\subseteq}\xi$, $\Pi_{(R,C,A)}(L){=}\{e{\in}L | \pi_{\mathcal{R}}(e) \in R \wedge \pi_{\mathcal{C}}(e) \in C \wedge \pi_{\mathcal{A}}(e) \in A\}$.*

For example, in our running example, the process performance analysis shows that the first activity for the patient, *registration*, is the bottleneck of the process. Projecting the event log of the hospital to that specific part structures the simulation model. Therefore, the projected event log only includes the events containing the *registration* activity.

Preprocessing. The design choices indicate which parts of the process should be considered for simulation modeling. The projection functions return an event log in which the events are only from the specified set of insights. Moreover, we define a time projection function to capture the provided event logs between two specific timestamps, e.g., indicated as tw in Fig. 1.

Definition 4 (Time Window Projection). *Let ξ be the universe of events and $\mathcal{T}$ be the universe of timestamps. For $t{\in}\mathcal{T}$ and $\delta{\in}\mathbb{N}$, given $L{\subseteq}\xi$, we define $Event_{t,t+\delta}(L){=}\{e{\in}L | t{\leq}\pi_{\mathcal{T}_S}(e){\leq}t + \delta\}$ and $CaseEvent_{t,t+\delta}(L){=}\{e{\in}L | \exists_{\sigma \in \overline{L}} e \in \sigma \wedge \exists_{e' \in \sigma} t \leq \pi_{\mathcal{T}_S}(e') \leq t + \delta\}$. The projection function $P_{t,t+\delta} : 2^{\xi} \nrightarrow 2^{\xi}$ returns a set of events, such that, $P_{t,t+\delta}(L){=}Event_{t,t+\delta}(L) \cup CaseEvent_{t,t+\delta}(L)$.*

An event log can be broken down into smaller ones per time period, e.g., instead of an event log of 10 days in an organization, 10 event logs for each day exist. Before extracting the variables on top of the projected event logs, it is important to consider the overlapping events in different time steps, i.e., between every t and $t{+}\delta$ (a window of time tw). To address this issue, for $k \in \mathbb{N}$ as the number of times steps using δ as time window, two functions, *Event* and

CaseEvent in Definition 4 are defined. Using $Event_{t,t+\delta}$, all the events started and finished in i^{th} step are captured. For instance, assume δ to be one day, an event started in one day and finished the next day is only considered in the step that it has started in, i.e., the first day. *CaseEvent* returns all the events related to the cases that one of their events happened at i^{th} time step.

4.2 Variable Extraction (SD-Log Generation)

For each of the provided event logs as a result of applying *Time Window Projection* in Definition 4, the process variables should be designed. To design the process variables for the given event logs, i.e., process describers over time steps, performance indicators should be determined.

Process performance indicators can be derived from the timestamp attributes t_s and t_c for the cases, activities, and resources at different levels, which all are considered as aspects. For instance, *service time* of a case, an activity or a resource, *waiting time* of a case, and *time in process* of a case are the possible performance indicators. The aggregation functions also can be applied on top of the performance indicators. These functions can be chosen between mathematical functions such as average, median, and sum. For instance, the average service time of cases in an event log, i.e., paints in the hospital, is calculated using *average* as the aggregation function, *case* as the aspect, and *service time* as the performance indicator. Note that for calculating the performance indicators related to the case aspect, *CaseEvent* makes it possible to capture the related events from the present cases in that time window.

Process Variable in Definition 5 defines process variables by assessing the validity of combining different possible process aspects, performance indicators, and aggregation functions. First, we define a set of possible combinations as shown in Table 3. Based on the design choices and different parameters, i.e., possible process features, process variables are designed. The process variables values are calculated by Definition 6.

Table 3. The validator table, which 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.

Validator	IN												
	Value	Count					Service time			Waiting time			Time in process
AF	AS												
	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
⊥	False	True	False	True	True	True	False	False	False	False	False	False	False

Definition 5 (Process Variable). *Let* $AF{=}\{average, median, sum, \perp\}$ *be the set of aggregation functions,* $IN = \{service\ time, waiting\ time, time\ in\ process, count, value\}$ *be the set of performance indicators and* $AS = \{case,\ resource,\ activity,\ numerical\ attributes,\ categorical\ attributes\}$ *be the set of process aspects. We denote* $\mathcal{F}{=}AF \times IN \times AS$ *as the set of process features (Table 3).* $\mathcal{V}{=}DC \times \mathcal{F}$ *is denoted as the set of process variables. For the given design choice* $(R, C, A) \in DC$ *and the process feature* $f \in \mathcal{F}$, $v{=}((R, C, A), f) \in \mathcal{V}$ *is a process variable.*

Table 3 shows the possibility of combining different parameters to generate valid process features, e.g., it is not possible to apply the average function ($af{=}average$) on the number ($in{=}count$) of activities ($as{=}activity$) in an event log. These possible features are used to form process variables using the design choices in Definition 3.

Definition 6 (Performance Function). *Let* ξ *be the universe of events and* $\mathcal{V}$ *be the set of process variables.* $\Phi{:}\mathcal{V} \times 2^{\xi} \rightarrow \mathbb{R}_{\geq 0}$ *generates the value of the process variable of an event log.*

We generate the set of sequential states of the process with *Time Window Projection* function in Definition 4 and define *Performance Variable* in Definition 5. The next step is to generate the values of the process variables by applying *Performance Function* on the projected event logs as defined in Definition 6.

For instance, let L be the event log of the running example, $f{=}(af, in, as)$ be a process feature where $af{=}average$, $in{=}time\ in\ process$, and $as{=}case$, based on Table 3, the combination is valid. For the design choice (R, C, A), consider $R{=}\{\pi_{\mathcal{R}}(e) | e \in L\}$, $C{=}\{\pi_{\mathcal{C}}(e) | e \in L\}$, and $A{=}\{\pi_{\mathcal{A}}(e) | e \in L\}$, i.e., the sets of all the resources, cases, and activities in L, respectively. Therefore, $v = ((R, C, A), f)$ is the average time that all cases (patients) spend in the hospital. $\Phi(v, L)$ represents the value of this process variable, i.e., $\Phi(v, L){=}\frac{\sum_{i=1}^{|\overline{L}|} p_c(\widetilde{\sigma_i}) - p_s(\widetilde{\sigma_i})}{|\overline{L}|}$.

The calculated values of variables form a coarse-grained process log, referred to as SD-Log, over time. The values define the process over time at a higher level of aggregation and can be used for designing the simulation models. Definition 7 defines an SD-Log and Algorithm 1 transforms an event log to an SD-Log.

Definition 7 (SD-Log). *Let* $L \subseteq \xi$ *be an event log,* $\mathcal{V}$ *be a set of process variables,* $\delta{\in}\mathbb{N}$ *be the selected time window, and* $k{=}\lceil \frac{(p_c(L) - p_s(L))}{\delta} \rceil$ *be the number of time steps in the event log w.r.t.* δ*. The SD-Log of a given L and* δ *is* $sd_{L,\delta}{:}\{1, ..., k\} {\times} \mathcal{V} \rightarrow \mathbb{R}_{\geq 0}$*, such that* $sd_{L,\delta}(i, v)$ *represents the value of performance function* $\Phi(L, v)$ *in the* i^{th}*-time window* $(1{\leq}i{\leq}k)$*.*

Table 4(a) shows a sample SD-Log with $\delta = 1$ *day* that includes different process variables for the sets of all the resources (R), cases (C), and activities (A) in the sample event log of the hospital, e.g., $f{=}(average, service\ time, case)$ and $((R, C, A), f)$ represents the process variable v, average service time for all the cases, i.e., patients, and $\Phi(v, L)$ calculates the value of v in each day in the sample log L. Also, in Table 4(a), *number of resources* in the hospital per day

Algorithm 1: Variable extraction algorithm w.r.t. the given design choices, which generates SD-Logs for scenario-based analysis.

```
Input: $event\ log\ L$, $set\ of\ process\ variables\ \mathcal{V}$, $time\ window\ \delta$,
       $design\ choice\ des$
Output: $SD\!-\!Log\ sd$
$L'{=}\Pi_{des}(L)$
$t_S{=}p_s(L')$(start time of the event log)
$t_C{=}p_c(L')$(complete time of the event log)
$k{=}\lceil\frac{(t_C-t_S)}{\delta}\rceil$
foreach $i \in [1,k]$ do
    $L''{=}P_{t_S,t_S+\delta}(L')$
    $t_S{=}t_S+\delta$
    foreach $v \in \mathcal{V}$ do
        add $\Phi(v,L'')$ to $sd(i,v)$
    end
end
return $sd$;
```

Table 4. A part of two sample SD-Logs of the running example with a time window of 1 day using different design choices. Each row shows a time step, here 1 day, cell-values represent the process variables' values and columns represent the process variables.

(a)					(b)			
Time Window (Daily)	Arrival rate of cases	Number of resources	Average service time	Average waiting time in process	Time Window Daily	Number of resources (*registration*)	Average service time (*registration*)	Average waiting time in process (registration)
1	180	6	0.359	0.609	1	2	0.425	0.237
2	147	6	0.415	0.540	2	1	0.120	0.483
3	160	6	0.401	0.596	3	1	0.806	0.506
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

represents the process variable $v = ((R, C, A), (\perp, count, resource)) \in \mathcal{V}$, where R, C, and A are the sets of all the resources, cases and activities in the log, $R{=}\{\pi_{\mathcal{R}}(e)|e \in L\}$, $C{=}\{\pi_{\mathcal{C}}(e)|e \in L\}$, and $A{=}\{\pi_{\mathcal{A}}(e)|e \in L\}$. For generating Table 4(b), $A{=}\{registration\}$ and R and C are the same as (a).

Based on the design choices, the whole process or specific parts of the process are selected to be modeled and the *Time Window Projection* function generates all the steps of the time for the given time window. The rest of the algorithm is calculating the values of the variables and forming the SD-Log.

Then investigating the relationships between process variables with each other will result in a system dynamics model [10]. The simulation model can be populated with the values of the process variables. Hence, we have a model on which different scenarios for a process can be played. For each question, the components of the question, i.e., process variables, can be the target of the question like the number of finished cases per day or the ones influencing the target of the question such as the number of resources available per day in this example.

5 Evaluation

Our goal is to design higher-level simulation models of processes using the proposed approach. With the models and the extracted SD-Logs which include process variables over time, we assess the validity of the designed models based on the simulated values. To do so, we start with presenting the possible valid models, i.e., system dynamics models for the processes. We use the event logs with common attributes to perform what-if analysis. A real event log, BPI Challenge 2012, is considered to evaluate the approach, i.e., designing the models and extracting the process variables of the process, SD-Log. The possible scenarios considered in the designed model for evaluation are presented in Sect. 5.1. We use one of the scenarios as an example to show the evaluation of the approach.

Extracting the corresponding SD-Log from the event log based on the defined process variables to populate and run the models is the next step. In the last step, the simulation results are compared to the real values inside the SD-Logs, e.g., the simulated number of cases per day in the process and the values in the SD-Log which are derived from the event log. Finally, we discuss the evaluation results, limitations, and possible improvements.

5.1 Designing Simulation Models

To design the simulation models, capturing the relationship between variables directly influences the validity of the models. Either the relations are known beforehand which can be proven by the data or it is an assumption that can be supported or rejected by values of variables over time. For instance, it is known that the number of cases in the process is directly affected by the arrival rate of the cases per hour and the process finish rate. Based on the process and the domain knowledge, the relationship between the number of resources and the arrival rate is expected to be seen, and the variables in the SD-Log can support or reject this assumption.

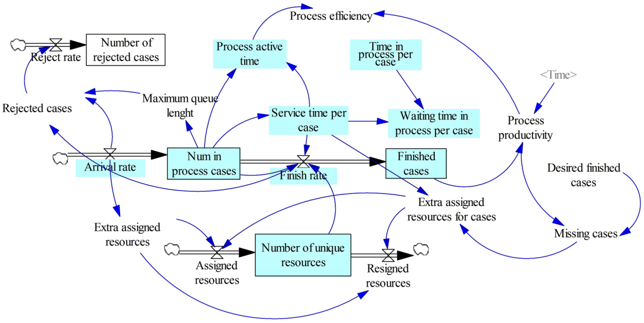

Fig. 4. The sample stock-flow diagram model for the business processes including multiple scenarios. The process variables directly extracted from the event logs are highlighted (blue). The model includes known and expected relationships inside a process at an aggregated level and can be customized for different levels, e.g., one organization. (Color figure online)

We design the basic model shown in Fig. 4 (highlighted elements), for the general process which is possible for validation since the variables can be extracted from the event logs' attributes. We extend the model with the possible external variables for possible business scenarios to answer more questions. This model can be used for different levels in the process, from an activity level to the general process based on the design choices. Common scenarios are inserted into the base model as follows:

- Process efficiency is the number of finished cases in the active time of the process per unit of time. Process efficiency gets affected by the number of cases in the process and the finish rate of the process.
- The effect of the arrival rate on adjusting the number of resources dynamically. An increase in the number of cases arriving in the process leads to an increase in the number of resources assigned to the process.
- Adjusting resources to achieve the desired number of finished cases per unit of time. In case that the finish rate is below the desired number per specified window of time, the resources can be increased or in the opposite situation, the unnecessary resources can be released.
- The effect of the desired capacity of the process on the number of rejected cases. The capacity of the process for handling the cases can be adjusted with the amount of possible rejected cases by the process.
- The effect of cases in the process per unit of time on the average service time of cases. The average service time can be decreased since the resources work faster under a specific amount of workload.

Figure 4 shows the designed model which can be applied in the process at different levels, e.g., one activity or one organization. It covers all the described scenarios and the performance variables presented in the can be validated.

Organizational/Process Blocks. The introduced models and scenarios can be applied to the organizations, activities, and resources in the processes.

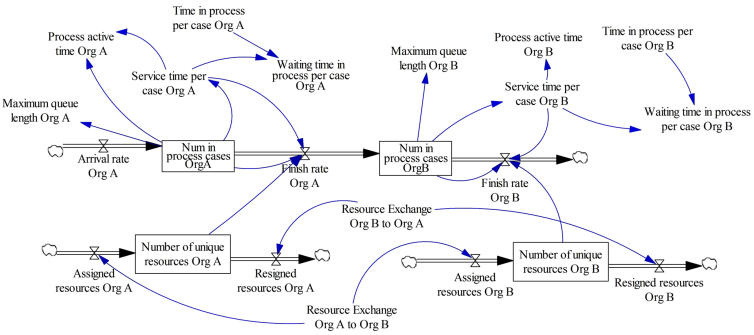
Time in process per case Org A
Process active time Org A
Service time per case Org A
Waiting time in process per case Org A
Maximum queue length Org A
Arrival rate Org A
Num in process cases OrgA
Finish rate Org A
Maximum queue length Org B
Process active time Org B
Time in process per case Org B
Service time per case Org B
Waiting time in process per case Org B
Num in process cases OrgB
Finish rate Org B
Resource Exchange Org B to Org A
Number of unique resources Org A
Assigned resources Org A
Resigned resources Org A
Assigned resources Org B
Number of unique resources Org B
Resigned resources Org B
Resource Exchange Org A to Org B

Fig. 5. The designed model for two organizations in the process which hand-over the tasks. Assessing possible scenarios such as how to share the resources between two organizations A and B for smaller queues is possible.

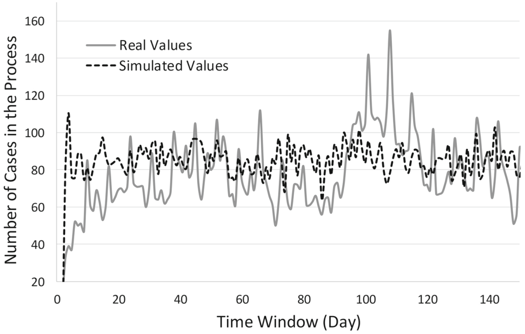

Fig. 6. The real and simulated values of the number of cases in the process only including the set of activities that were performed by the resources in BPI Challenge 2012.

Moreover, the extended models can be designed to capture the interaction between different parts of the process, e.g., two organizations which hand-over the work or the flow of cases, e.g., items in a production line or between different activities in the process. The most common scenario is that organizations sharing the flow of cases, therefore, organizations can exchange the resources and it can be modeled as shown in Fig. 5.

5.2 Evaluation Results

In this section, we assess the validity of the designed general models for the real processes using the provided tool [7] which is publicly available. As indicated in Sect. 5.1, models with variables outside the captured information in event logs are not possible to be validated completely. Therefore, we use the basic default models, highlighted part in Fig. 4 for this section to show the validity of the simulation models and their results. In the BPI Challenge 2012 event log, three different types of activities exist, i.e., performed by users, performed by the system, and performed by the resources. Performance analysis of the process reveals that the most time-consuming part of the process is the flow of tasks including the third type of activities, i.e., employees' tasks. The system-related tasks such as the submission of a request are instance tasks, i.e., the duration is zero, and not related to the efficiency and speed of the employees inside the organization. Therefore, we use the process which only includes the activities and tasks performed by the employees, i.e., their speed and efficiency affect the process. Using the *Design Choice* function, we created the projected event log only including the corresponding events to the third category of activities, i.e., design choice $=(R, C, A)$, where $R=$ the set of all resources in the log, $C=$ the set of all cases in the log, and $A=$ list of activities (employees' task).

Based on the time series analysis approach presented in [9] for time window selection, we chose a *one day* time window and focus on the general model for the simulation of the process in BPI Challenge 2012. We use the extracted variables

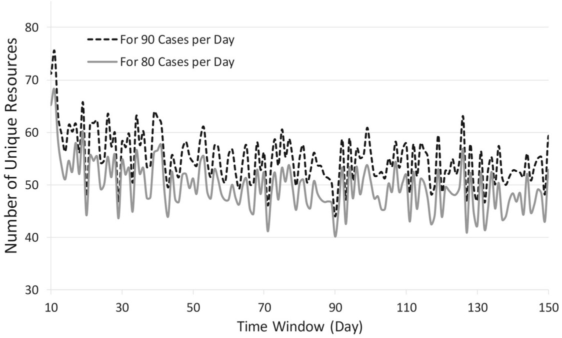

Fig. 7. The number of required unique resources using the dynamic assigning resources based on the desired number of finished cases per day. As shown, exploiting the extended model Fig. 4, on average 9% more resources per day is required to cover 90 finished cases instead of 80 cases.

to define the underlying equations inside the simulation models as proposed in [10]. The populated model with the equations and the values from SD-Log is simulated for 150 steps (days).

The results for the variable *number of cases in the process* in each time window are shown in Fig. 6. Calculating the average pair-wise error of the steps shows 24%. In order to form the stock-flow model, we used the functions that generate the random values for the variables such as the arrival rate using their discovered distribution. The validated model and the values of variables from the SD-Log can be used to exploit further scenarios for the extended model. For instance, in order to see the effect of an increase in the number of desired finished cases from 80 per day to 90 cases, using the model in Fig. 4, the simulation results show that on average 9% increase in the number of unique resources per day is required. Figure 7 represents the results for the two scenarios. The dynamic adjusting of the resources is done by captured relations among the variables, i.e., *assigned resources*, *number of missing cases*, and *average service time.*

Discussion. Using an event log, SD-Logs can be generated which are used to design and populate the corresponding system dynamics models. Inserting the effect of external factors increases the possibility of what-if analysis. However, by adding external factors from outside the event logs into the simulation models, the pair-wise evaluation is not possible, e.g., consider models including variables such as resources expertise, or their efficiency. Therefore, we start with generating models including the variables extracted from event logs and evaluate those, after that, we introduce the external factors to the models for further simulations and what-if analyses. Moreover, capturing the dynamic behavior of processes over windows of time is not always a straightforward task. For instance, in the event log of an emergency room, it is difficult to capture similar patterns in a daily manner for the process variables such as the arrival rate. Therefore, the evaluation of the results of the system dynamics simulation is not accurate enough, and it depends on the time window.

Generating SD-Logs and simulation models using bigger windows of time, e.g., one week instead of a daily manner, can increase the accuracy of the models w.r.t. the pair-wise comparisons of results. Given the above-mentioned concerns, applying the approach on the case studies with known influential external factors, e.g., the amount of money spent on the advertisement and the duration of the advertisement in a process, verifies the approach in practice. In principle, the quality of the captured data from event logs and the process domain knowledge affect the quality of the models.

6 Conclusion

In this paper, we presented an approach to capture the processes in a quantified manner over time. Describing the processes using process variables makes designing valid simulation models possible. We started from event logs and by exploiting process mining techniques the possible design choices are identified. All the possible process variables which represent the process over time w.r.t. different aspects are extracted. The provided functions imply how the design choices can be taken into action using the provided insights by process mining. These design choices are applied to the event logs. Moreover, performance functions are introduced regarding the existed aspects and levels in the event logs. The derived coarse-grained process logs, called SD-Logs, are created based on the performance functions for the generated variables over time and are used to form simulation models for "what-if" analyses. Furthermore, the general models are presented as guidelines for designing possible scenarios which can be customized based on the process variables and scenarios for different processes. We assessed the validity of the designed model using real event logs. The next step is to focus on the underlying equations between variables. These equations are used as a baseline of more accurate stock-flow diagrams in system dynamics modeling for simulation purposes.

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

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, 2nd edn. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
2. van der Aalst, W.M.P.: Process mining and simulation: a match made in heaven! In: Proceedings of the 50th Computer Simulation Conference, SummerSim 2018, pp. 4:1–4:12 (2018)
3. Camargo, M., Dumas, M., González, O.: Automated discovery of business process simulation models from event logs. Decis. Support Syst. **134**, 113284 (2020)

4. Duggan, J.: A comparison of Petri net and system dynamics approaches for modelling dynamic feedback systems. In: 24th International Conference of the Systems Dynamics Society (2006)
5. Khodyrev, I., Popova, S.: Discrete modeling and simulation of business processes using event logs. In: Proceedings of the International Conference on Computational Science, pp. 322–331 (2014)
6. Munoz-Gama, J.: Conformance Checking and Diagnosis in Process Mining - Comparing Observed and Modeled Processes. Lecture Notes in Business Information Processing, vol. 270. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-49451-7
7. Pourbafrani, M., van der Aalst, W.M.P.: PMSD: data-driven simulation using system dynamics and process mining. In: Proceedings of Demonstration at the 18th International Conference on Business Process Management, pp. 77–81 (2020), http://ceur-ws.org/Vol-2673/paperDR03.pdf
8. Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M.P.: Scenario-based prediction of business processes using system dynamics. In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) OTM 2019. LNCS, vol. 11877, pp. 422–439. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33246-4_27
9. 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: Dobbie, G., Frank, U., Kappel, G., Liddle, S.W., Mayr, H.C. (eds.) ER 2020. LNCS, vol. 12400, pp. 77–91. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62522-1_6
10. 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: Abramowicz, W., Klein, G. (eds.) BIS 2020. LNBIP, vol. 389, pp. 249–263. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53337-3_19
11. Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M.P.: Supporting decisions in production line processes by combining process mining and system dynamics. In: Ahram, T., Karwowski, W., Vergnano, A., Leali, F., Taiar, R. (eds.) IHSI 2020. AISC, vol. 1131, pp. 461–467. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-39512-4_72
12. Rosenberg, Z., Riasanow, T., Krcmar, H.: A system dynamics model for business process change projects. In: International Conference of the System Dynamics Society, pp. 1–27 (2015)
13. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering simulation models. Inf. Syst. **34**(3), 305–327 (2009)
14. Song, M., van der Aalst, W.M.P.: Towards comprehensive support for organizational mining. Decis. Support Syst. **46**(1), 300–317 (2008)
15. Sterman, J.D.: Business Dynamics: Systems Thinking and Modeling for a Complex World. McGraw-Hill, New York (2000)
16. Sterman, J.D.: All models are wrong: reflections on becoming a systems scientist. Syst. Dyn. Rev. J. Syst. Dyn. Soc. **18**(4), 501–531 (2002)