IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 June 2022, accepted 20 July 2022, date of publication 25 July 2022, date of current version 29 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3193507

RESEARCH ARTICLE

Discovering System Dynamics Simulation Models Using Process Mining

MAHSA POURBAFRANI[1] AND WIL M. P. VAN DER AALST[2], (Fellow, IEEE)
[1]Process and Data Science Group, RWTH Aachen University, 52074 Aachen, Germany
[2]FIT GmbH, 53757 Sankt Augustin, Germany

Corresponding author: Mahsa Pourbafrani (mahsa.bafrani@pads.rwth-aachen.de)

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.

ABSTRACT Process mining techniques are able to describe and model real processes using historic event data extracted from the information systems of organizations. Later, these insights are used for process improvement. For instance, Discrete Event Simulation (DES) uses process models that are able to mimic real-world events. However, the aggregated performance status of processes over time reveals various hidden relationships between process variables. Coarse-grained process logs are sets of performance variables over steps of time, generated using event data from processes. The coarse-grained process logs describe processes at higher levels. *System Dynamics* completes process mining by capturing the relationships between various process variables at a higher level of abstraction. In this paper, we propose a new framework for capturing conceptual models of processes using transformed event data. The main idea is to automatically discover the underlying relations as equations. This allows us to generate system dynamics simulations of processes. We employ a variety of statistical and machine learning techniques to discover the hidden relationships between process variables. The framework supports the simulation modeling task in the context of system dynamics simulations. The experiments using real event logs demonstrate that our approach is able to generate valid models and capture the underlying relationships.

INDEX TERMS Process mining, scenario-based predictions, system dynamics, what-if analysis, simulation, event logs, coarse-grained process logs.

I. INTRODUCTION

Historical data of executions of processes stored in information systems provide a valuable source of knowledge for improving processes inside organizations. Running business processes consist of different events that shape the event data. Process mining is a set of data-driven techniques for unlocking the power of event data in organizations [1]. It provides a variety of insights into processes, such as discovering process models, determining whether the discovered models and event data are aligned [2], and revealing performance and bottleneck analysis [3]. These process views in different aspects should be put into action, i.e., the discovered status of a process and its problems should be addressed w.r.t. process improvement.

Process mining has proven its ability to deliver *backward-looking* insights, but there is a growing demand for *forward-looking* insights that can be used to change processes. All techniques in process mining that intend to undertake future analysis are referred to as forward-looking techniques. We divided them into two categories: simulation and prediction techniques. The mainstream forward-looking techniques in process mining are also at a detailed level, e.g., predicting the remaining time of a case using machine learning techniques [4] or simulating processes in detail [5]. Simulation techniques are well-known forward-looking techniques that were introduced into the process mining field 15 years ago [6]. *Discrete Event Simulation* (DES) is a commonly used approach to play-out process models at a detailed

The associate editor coordinating the review of this manuscript and approving it for publication was Jesus Felez.

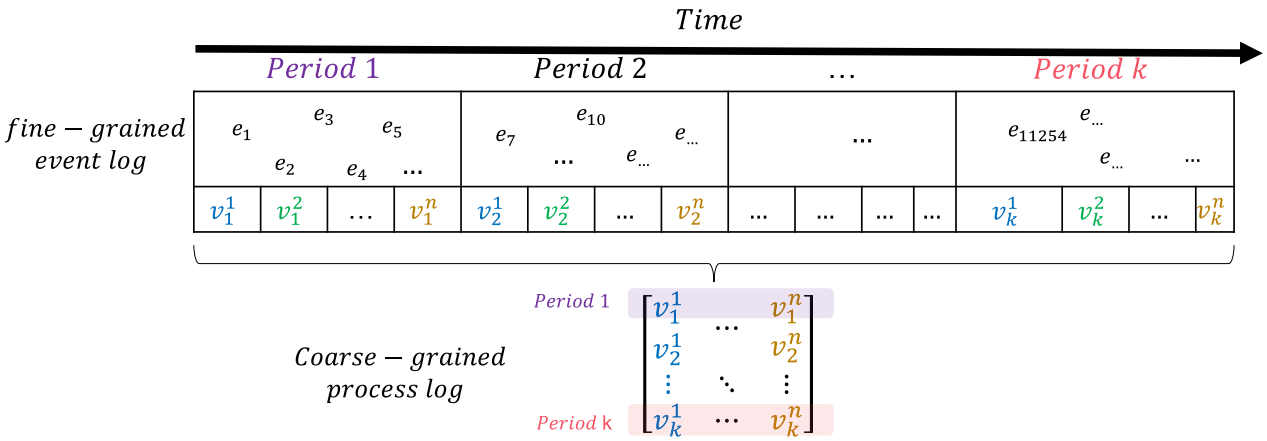

FIGURE 1. The transformation of fine-grained event logs into coarse-grained process logs, where the steps are time steps, e.g., days, rather than individual events.

level [7]. Simulation models and simulation outcomes are both improved by using process mining approaches such as [8]. However, at detailed levels, some aspects of a process remain concealed and can only be captured at a higher level of aggregation. The impact of strategic and high-level decisions, as well as external factors such as resource expertise, are, for example, overlooked [9].

In contrast to discrete event simulation or other detailed modeling techniques that are based on individual entities, system dynamics techniques are based on aggregation, e.g., the number of people or products per day [10]. These techniques are able to cover a wide range of effects, including human factors, and model nonlinear relations at an aggregated level [11]. System dynamics tends to describe and capture a system using its variables and the underlying effects among them. Such approaches seek to provide a holistic model of a system that incorporates all possible effective variables in the system over steps of time [12]. However, most simulation-based approaches, including system dynamics, highly rely on users and their understanding of the system.

Given that system dynamics simulation models can capture the relationships between external components and integrate different business processes using intermediate variables, system dynamics simulation models aim for higher levels of simulation for decision-making in the context of business processes. For example, the impacts of marketing department actions such as advertisements or human resource department actions such as hiring are not directly part of the business processes. However, any decision may change and affect the business processes. The number of produced items can change over time by defining the hiring rate as a variable and modeling its effect over time on the number of resources in the process. As a result, system dynamics simulation models can capture these effects and assist decision makers and business owners in the business process.

The proposed approach in [12] transforms detailed event data of processes (*fine-grained event logs*) into aggregated event logs (*coarse-grained process logs*) as presented in Figure 1. The coarse-grained process logs, i.e., a collection of measurable aspects from an event log, are referred to as System Dynamics Logs (SD-Logs) since they are used for designing and generating system dynamics models. Table 1 presents different levels of capturing a process: (1) fine-grained event logs as standard event logs that steps are events, and (2) coarse-grained process logs as SD-Logs that the steps are a period of time (time steps). Each level can be used for different simulation techniques, as proposed in [13], where the results of the coarse-grained simulations are used to update processes at detailed levels and later simulate the DES models at operational levels. Performing activity *Test* for patient *P335* by resources *Lisa* at a specific timestamp is an event in common event logs, referred to as fine-grained event logs. When simulating at the detailed level, each step of the simulation is to generate such events. While a single patient is not the focus of coarse-grained process logs, the number of patients tested per day is. While simulating at the higher level, the number of patients is considered, and each simulated value is the value of that variable per day.

Then, the designed models are populated with the values of these measurable aspects in SD-Logs. Afterward, the validation step is performed to measure the similarity of the generated results by the models with the real values in the SD-Logs. The proposed framework is shown in Figure 2.

In this paper, the focus is on the *Model Creation* steps and making these steps data-driven to support users. We propose a data-driven framework that exploits statistical and machine learning techniques to capture processes at a higher level as *Causal-loop Diagrams* (CLD) in system dynamics. Causal-loop diagrams are the system dynamics representations of systems, where system variables and their negative/positive relations are shown and designed based on the user's domain knowledge. The data-driven CLDs are used to support high-level simulation modeling. The main focus is to determine relationships between process variables and formulate corresponding equations. We enhance the presented approach in [14] by utilizing data and automating the CLD

TABLE 1. The presented terms in the paper w.r.t. simulation and event data. A process can be simulated at different levels of granularity using different levels of data. *SD* is an example of high level simulation of processes using *System Dynamics*.

Simulation models	logs	Example of logs for simulations	Steps	Examples of simulation models
Detailed process simulation	Fine-grained event log	Standard event logs	Events	DES
High-level process simulation	Coarse-grained process log	SD-Logs	Time steps	SD

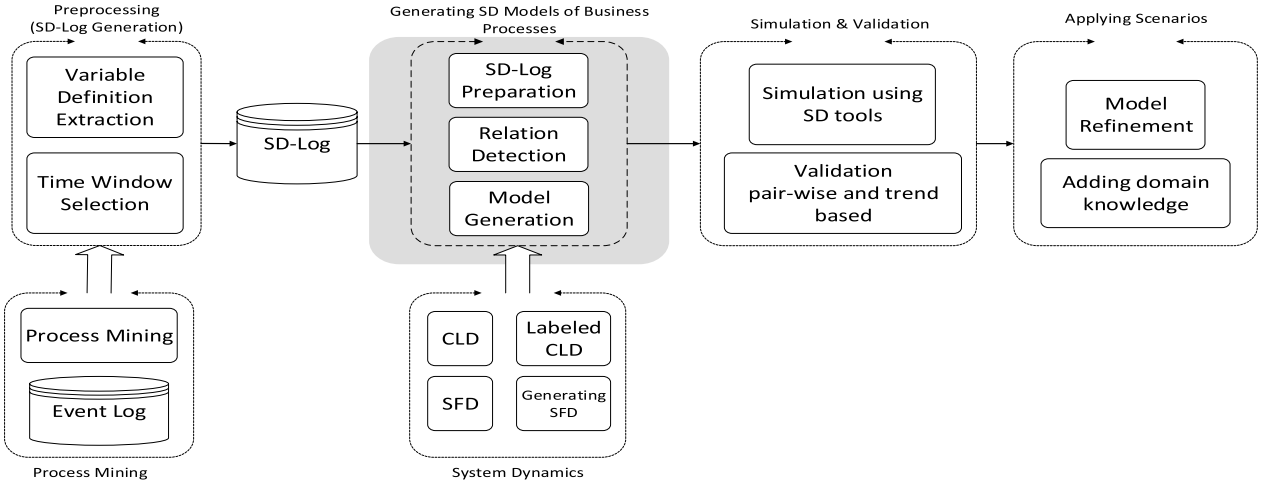

FIGURE 2. An overview of the proposed framework, integrating process mining and system dynamics in order to design valid models to enable scenario-based prediction of business processes [15]. The emphasis of this paper is on the highlighted step, *Model Creation*. The introduced process mining and system dynamics concepts (bottom) are exploited in the SD model generation section.

generating process. Our approach enables the construction of process system dynamics models to be supported. The generated models are data-supported, including mathematical relationships between process variables, i.e., equations. Therefore, assessing the potential effects of future changes in business processes is possible. The validity of the designed system dynamics models for the processes and the improvement in what-if analysis results are shown using real event logs.

A. STRUCTURE OF THE PAPER

To explain the approach, we expand on the relationships and positions of the introduced concepts in Sections III and IV. First, we define event logs, where events are the components of event logs. Then, system dynamics notations and concepts regardless of process mining context are explained using an example. Furthermore, the required notations of CLD and SFD, which are the purpose of the paper to generate data-driven Causal-loop diagrams and later support designing stock-flow diagrams, are explained. From this point, the defined event logs are used to generate the presented SD-logs in Definition 9. The techniques are applied to the generated SD-Logs as a dataset and, using Algorithm 1 and the defined methods inside, the results are transformed into the defined form of CLD on the basis of the definition of CLD.

The remainder of this paper is organized as follows. We present related work for using system dynamics and process mining in Section II. In Section III, we introduce background concepts and basic notations used throughout the paper. In Section IV, we present our main approach, including the detailed design and implementation steps. We evaluate the proposed approach in Section V. Section VI concludes our work and discusses potential directions for future work.

II. RELATED WORK

Various approaches provide backward-looking process mining techniques for analyzing event data of processes. Most of the techniques in the process discovery and conformance checking areas are derived from the general approaches proposed in [1]. Process enhancement approaches are also proposed and widely employed in practice [16]. It encompasses a wide range of analyses, from performance and bottleneck analysis to process improvement.

Multiple data science methods are used in order to support the process improvement aspects, such as simulation and prediction. Statistical analysis and machine learning techniques have been introduced in the field of process mining for different purposes. For instance, in [4], LSTM networks are used to predict the remaining time and the next activity of process instances. In [17], the clustering techniques with the goal of process discovery and decision mining on top of the features are exploited. Moreover, multiple approaches address the simulation of business processes using process mining insights [7] uses the provided insight at an instance level to generate a simulation model for a business process. In [8], the authors employ process mining techniques

to design the detailed simulation model of a process while employing hyperparameters to discover the best simulation settings.

However, the majority of the provided techniques either blindly use all the extracted features from event data as inputs to machine learning techniques or extract the features at the instance level, i.e., single cases in a process matter. Since the provided insights in backward-looking process mining are mainly at the detailed level, the actions taken for simulations and predictions are also mainly at the detailed level.

Recent approaches have tried to capture different aspects of processes and add process context into the simulation and prediction techniques in process mining. For instance, in [18], the performance spectrum is proposed to be used for monitoring the real-time analysis of processes and, to some extent, using the aggregated variables such as queue concepts in the process. In [19], the importance of the current workload for handling cases in a process is presented as one of the determining factors besides the detailed information of a single process instance. In addition to the necessity to incorporate process context and higher levels of aggregation into forward-looking approaches, external variables, such as the effects of human or legal aspects on simulation results, should not be overlooked.

The requirement at the decision-making level for considering different actors and analyzing the future of processes at an aggregated level makes techniques such as system dynamics a suitable tool for modeling business processes, which we employ system dynamics technique to achieve this goal. In addition to the level of the simulation models, the design of reliable models and the validity of their results can be supported with the process mining techniques and the existing historical data as used in discrete event simulation in process mining and decision support systems on the basis of process mining [20]. Therefore, we identify the tools and techniques based on their capabilities that are used to fulfil this aim w.r.t. the motivation of this research, i.e., combining process mining and system dynamics.

System dynamics is able to model the behavior of a system in terms of system variables over steps of time [11]. In business process management, system dynamics modeling is used to simulate business processes [21]. The authors in [22], performed a case study to show the use of both process mapping and system dynamics modeling in improving business processes. System dynamics modeling has also been used in supply chain management [23]. These approaches are based on conventional modeling and are based on the user's domain knowledge to model the systems. In [14], a different type of simulation for business processes using system dynamics is proposed. System dynamics is used in the new approach in process mining to model processes at different levels. The goal is to look for the hidden relations inside the processes. The process variables are defined and extracted at a higher level, e.g., the average daily arrival instances in the process. Furthermore, the clarity of relationships and the ability to track the effect of changes in the process are significant advantages of simulation models in process mining over training machine learning black box models. The proposed approach in [14] uses aggregated process variables over steps of time and discovers the linear and nonlinear relations between the variables. This framework reveals the effects of the variables on each other and generates causal-loop diagrams.

The proposed framework in [12] is capable of covering all possible process variables from an event log based on process mining general insights, such as a bottleneck in one of the organizations. The size of the time window in which the variables are calculated is also influential when generating simulation models of processes at aggregated levels. The designed framework in [24] returns the best time window using time series analysis and training different models over steps of time for the possible time windows in the process, e.g., day. The SD-Log generation, time window selection, and the relation detection modules are implemented in [25]. Based on the existing work, we provide a framework for not only automatically conceptualizing aggregated processes, i.e., causal-loop diagrams (CLD), but also for providing underlying equations that enable the generation of aggregated simulation models, i.e., stock-flow diagrams (SFD).

III. PRELIMINARIES

In this section, we define basic concepts for process mining and system dynamics, as well as the functions that are used in the proposed approach.

A. PROCESS MINING

Process mining uses past executions of processes in the form of event logs. An event log captures events that include case ID, timestamps, activity, resource, and other possible attributes, e.g., case type. In Definition 1, we introduce event logs as a set of events. The majority of process mining approaches define event logs as multisets of traces. Since our focus is on simulation and transforming the event logs into different levels and then breaking down the event logs over time, the events are considered the main components of the event logs. It should be noted that it reflects the same concept, and this form of definition is considered for design and implementation of the approach.

Definition 1 (Event Log): Let $\mathcal{C}$, $\mathcal{A}$, $\mathcal{R}$ and $\mathcal{T}$ be the universe of cases, activities, resources, and timestamps, respectively. 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. 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 $\pi_{\mathcal{T}_C}: \xi\rightarrow\mathcal{T}$ are defined for attributes of events. Events are unique given their attributes, and an event log L is a set of events, i.e., $L\subseteq\xi$.

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

TABLE 2. A part of a sample event log. Each event is presented in a single row, with the Case ID, Activity, Resource, and Start and Complete Timestamps.

Case ID	Activity	Resource	Start Timestamp	Complete Timestamp
154	initiate request	John	02/03/2021 10:30:52	02/03/2021 11:02:00
155	initiate request	Max	02/03/2021 10:34:02	02/03/2021 11:17:00
154	check request	Eric	02/03/2021 10:31:50	02/03/2021 11:14:10
154	decide	Max	02/03/2021 13:30:40	02/03/2021 14:02:00
155	check request	Eric	02/03/2021 12:17:35	02/03/2021 14:23:00
156	initiate request	Rose	02/03/2021 10:41:13	02/03/2021 10:50:00
...	...	...	...	...

of events with the same case identifier and ordered in time represents a process instance, i.e., a trace. Table 2 represents a part of a sample event log for the process of requesting a loan in a financial company. For instance, the event e_1 represents that the activity *initiate request* (a) was started at timestamp *10:30:52 02.03.2020* (t_s) by resource *John* (r) and was completed at timestamp *11:02:00 02.03.2020* (t_c) for a customer with case ID 154 (c). The sequence of events for the same customer w.r.t. start time is referred to as a trace in the process, e.g., the sequence of activities for Case ID 154 is *initiate request, check request, assess the credit, decide, and accept the request*.

B. SYSTEM DYNAMICS

System dynamics techniques are employed in order to model complex systems and the relationships between system variables and their environments. These techniques are used to model systems with various types of internal and external interactions, information feedback, and effect/change relations. The modeled systems using system dynamics techniques are able to capture the effects of decisions and applied strategical changes on the systems [26].

1) CAUSAL-LOOP DIAGRAM

In system dynamics, systems or the corresponding problems/scenarios can be visualized in order to illustrate the cause and effect relations between the system variables. A *Causal-loop Diagram* (CLD) is one of the representative techniques that shows constituent components and their interactions, i.e., directions of relations and whether they are positive or negative. Therefore, it is possible to understand the behavior of the systems over time [27].

CLDs are sometimes considered a qualitative representation of a system [10]. We define CLDs formally in Definition 2. CLDs are directed graphs including nodes, i.e., systems' variables, and arcs, i.e., their relations. The idea behind CLDs is to capture the feedback loops (positive/negative effects) inside a system.

Figure 3 shows a CLD modeling the Covid-pandemic. The example is just for illustration purposes and does not reflect reality.

Definition 2 (Causal-Loop Diagram): Let $\mathcal{V}$ be the universe of variables. Let $V \subseteq \mathcal{V}$ be the set of variables for a system, $R \subseteq V \times V$ be the set of directed links, and function $\eta : R \rightarrow \{+, -\}$ specifies the labels of the directed links, i.e., either $+$ or $-$. CLD=(V, R, η) is a causal-loop diagram of the system. The CLD is represented visually as a graph where V are nodes, $R \subseteq V \times V$ are the arcs and the labels of arcs are specified by function η. A directed link $r=(v, v') \in R$ connects nodes v and v' using a directed arc.

Figure 3 depicts a sample CLD with six variables, such as $r_1 =$ *(number of Covid deaths, number of vaccinated people)* and $\eta(r_1) = +$, which indicate that the *number of Covid deaths* has a positive effect on the *number of vaccinated people*. The label of the arc means that if the *number of Covid deaths* increases or decreases, so will the *number of vaccinated people*, i.e., they will change in the same direction.

Given the fact that a CLD represents the effects of a system's variables on each other, the dependent and independent variables of the system can be extracted. Independent variables can be from the environment of the system and are not affected by other system variables, e.g., *cold weather* in Figure 3. Consider the number of people applying for a loan per day. It can be an independent or dependent variable w.r.t. the system, e.g., the policy of the company. When it is not influenced by other variables, such as assigned resources, it is independent. However, when there is an advertisement policy in the company based on the revenue and the previously provided loan, the number of people applying for loans per day is a dependent variable. We use the dependent and independent variables later to discover the equations inside the simulation models.

We consider variables to be independent with respect to the discovered and provided evidence in the data in our approach, and it is limited to domain knowledge about the system. As a result, as shown in the example of the advertisement policy, the dependency might exist but was not detected by the data or the user. Since the goal is to provide data-driven modeling, when we refer to variables being independent, we rely on the data and existing domain knowledge.

Definition 3 (Dependent and Independent Variables): Let $\mathcal{V}$ be the universe of variables, $V \subseteq \mathcal{V}$ be the set of variables of a system, and $R \subseteq V \times V$ be the set of relations between variables in a CLD=(V, R, η). We define $\overrightarrow{V} = \{v \in V | \bullet v = \emptyset\}$ as the set of independent variables where $\bullet v = \{v' \in V | (v', v) \in R\}$. $\overrightarrow{V}$ is the set of variables not influenced and affected by the other variables directly, i.e., these have no incoming arcs in the CLD. We also define $\overleftarrow{V} = V \setminus \overrightarrow{V}$ as the set of dependent variables that are influenced and affected at least by one variable.

For the sample CLD in Figure 3, $\overrightarrow{V} =$ *{cold weather}* and $\overleftarrow{V} =$ *{number of vaccinated people, number of infected*

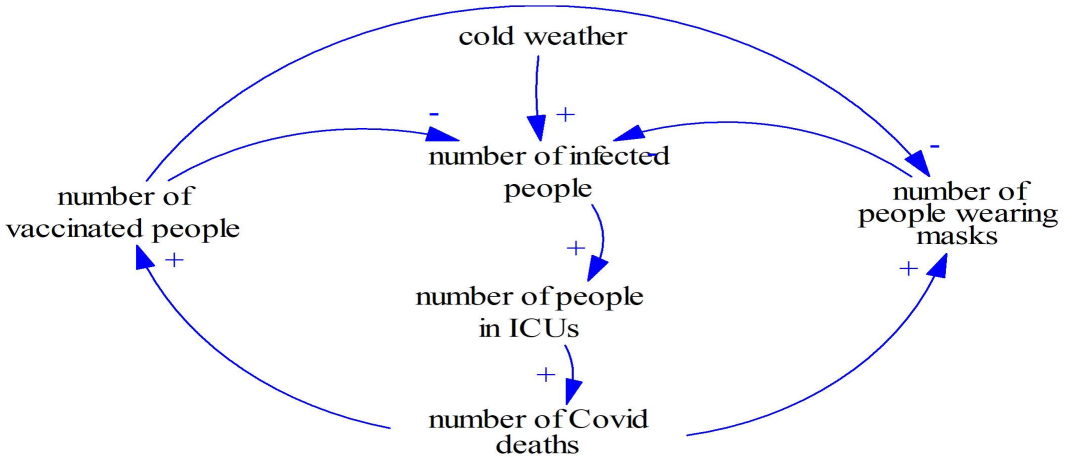

FIGURE 3. A sample Causal-loop Diagram (CLD) capturing the effects of different factors (variables) on each other, e.g., the negative sign indicates that if the number of vaccinated people increases, the number of infected people decreases.

people, number of people in ICUs, ...}. Cold weather as an independent variable is not affected by other variables in the defined CLD but is able to affect the *number of infected people*.

2) STOCK-FLOW DIAGRAM (SFD)

The main focus of system dynamics simulation is accumulation behavior in the system w.r.t. stocks. System dynamics aims to simulate systems at an aggregated level, therefore, the notations of stocks (accumulative over time) and flows (adding/removing to the stock over time) are introduced. *Stock-flow Diagrams* (SFDs) are designed to add the mathematical equations for calculating and simulating the value of stocks over time [28].

To design the SFD for the purpose of simulation, every $v \in V$ in $CLD{=}(V, R, \eta)$ should be assigned to one of the types of elements in SFD, S (Stocks), F (Flows), or A (Auxiliaries). Each element is defined as follows [26]:

- *Stocks* are variables that accumulate over time and are represented numerically. Their values are increased or decreased as a result of inflows and/or outflows. Stocks can only be changed through inflows and outflows.
- *Flows* are rate-based variables, such as monthly income, which can be considered as flows that can add to or subtract from stocks.
- *Auxiliaries* are additional variables that can have static values or change over time. They represent system components whose values are influenced by other system components or influence others.

In addition to the types of elements, the types of relations between pairwise elements in an SFD are either information dependencies (I) or items/materials flows (M). System dynamics rules indicate a couple of constraints on the types of relations as indicated in [10], e.g., variables only influence variables or flows using information dependencies. We elaborate on the constraints in the generating stock-flow diagram step.

Definition 4 (Stock-Flow Diagram): A Stock-flow Diagram (SFD) is a tuple $(A, S, F, I, mapf)$ *where* $V{=}A \cup S \cup F$ *is the set of pair wise disjoint system variables,* $S \neq \emptyset$ *are stocks,* F *are flows, and* A *are auxiliaries.* $mapf : F \rightarrow S \times S$ *is the function that defines flows of items/materials between two stocks, and* $I \subseteq V \times (F \cup A)$ *are the information dependencies. We denote* $\text{☁} \in S$ *to be the stock representing the system boundary.*

Each type of SFD element introduced in Definition 4 is visualized with a specific shape, see Figure 4. Table 3 presents the sets of elements in the example SFD in Figure 4. $S{=}\{$☁, *number of people in ICUs, number of vaccinated people, ...*$\}$, $F{=}\{$*number of infected people per week, death rate, ...*$\}$, and $A{=}\{$*cold weather*$\}$ are the set of stocks, flows, and variables, respectively. *(cold weather, number of infected people per week)*$\in I$ is an information dependency where the value of variable *cold weather* affects the value of flow *number of infected people per week*. Also, *mapf (number of infected people per week)*$=$(☁, *number of people in ICUs)* represents the flow of items/materials that are added from the environment to the *number of people in ICUs*, e.g., the weekly rate of infected people, which adds to the number of people in ICUs. The information dependencies and items/materials flows in the SFD are taken from relations (R) in the corresponding CLD. Note that auxiliaries only influence auxiliaries or flows using information dependencies, and flows cannot influence other flows using items/materials flows [28].

Since system dynamics simulates the system over specific steps of time, variables in each step have values that can be updated in the next step. For a given CLD and the corresponding SFD, the underlying equations are indicated in Definition 5. Inserting the underlying equations makes the quantitative simulation of SFDs possible. Note that, given the fact that values of stocks are accumulative values of the flows influencing them over time, we distinguish among the

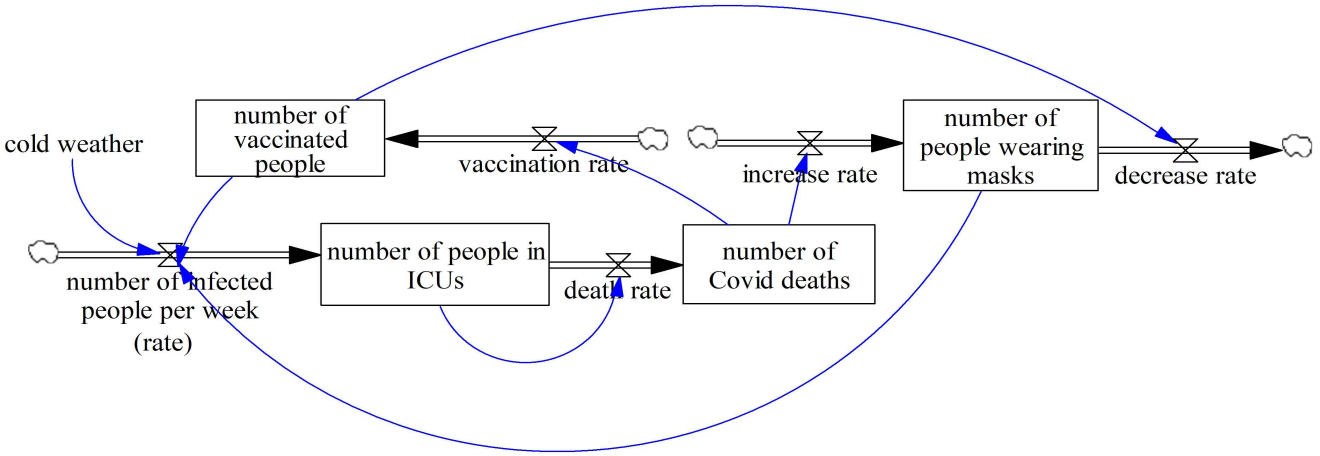

FIGURE 4. An example Stock-flow Diagram (SFD) for the given CLD in Figure 3. For a better visualization, environment shape is presented multiple times as the representor of outside and environment of the system.

TABLE 3. Sets of elements in the example SFD in Figure 4. The stocks (*S*), flows (*F*), auxiliaries (*A*), and the information dependencies (*I*) are specified.

S={number of vaccinated people, number of people in ICUs, number of Covid deaths, number of people wearing masks}
F={number of infected people per week, vaccination rate, death rate, increase rate, decrease rate}
A={cold weather}
I={(cold weather, number of infected people), (number of vaccinated people number of infected people),(number of people in ICUs, death rates), (number of Covid deaths, vaccination rate), ...}

underlying equations of stocks and other variables (flows and auxiliaries).

Definition 5 (Underlying Equations): For a given CLD=(V, R, η) and its corresponding SFD=$(A, S, F, I, mapf)$, we define the underlying equations as follows, where v_i represents the real/simulated value of variable $v \in V$ at $i \in \mathbb{N}_{\geq 1}$ (a step of time) and $\bullet v=\{v^1, \ldots, v^j\}$ is the set of variables that affect the value of v:

- *if $v \notin S$ then $v_i=Eq^{\bullet v}(.)$, where function $Eq^{\bullet v}(.)$ calculates the value of variable $v \in V$ at time step i based on previous/current values of variables in $\bullet v$.*
- *if $v \in S$, $v_i=v_{i-1} + \Sigma_{(v^j,v)\in M}\ \eta((v^j, v)) * v_i^j$ for $i \geq 1$, where v_0 is considered as a given initial value for $v \in S$, and $M=R \setminus I$.*

In each step, values of stock-flow elements get updated based on the current/previous values of the other elements that influence them. For instance, Equation 1 and Equation 2 illustrate the underlying equations in the sample SFD in Figure 4. At time step *i*, e.g., fourth week, the *number of people in ICUs* is equal to the number of people already in ICUs, plus the difference of the *number of infected people per week* and the *death rate* at time *i*, e.g., fourth week. Note that the values of the *number of infected people per week* and the *death rate* are rate-based and dependent on the time step (*i*), e.g., per week.

$$\begin{aligned} &number\ of\ people\ in\ ICUs_i \\ &= number\ of\ people\ in\ ICUs_{i-1} \\ &+(number\ of\ infected\ people\ per\ week_i - death\ rate_i) \end{aligned} \tag{1}$$

$$\begin{aligned} &death\ rate_i \\ &= number\ of\ people\ in\ ICUs_i * 10\% \end{aligned} \tag{2}$$

3) GENERATING STOCK-FLOW DIAGRAM (SFD)

To convert a CLD to an SFD, the first step is to label the CLD. There are different strategies to label a CLD [29], we start with labeling the stocks and flows. Then, given the stocks and flows, the flows of items/materials and information dependencies are identified. Note that labeling the variables for SFD should be done based on the domain knowledge of users.

Definition 6 (Labeled CLD): Let $V \in \mathcal{V}$ be the set of a system's variables, $R \subseteq V \times V$ be the set of relations between the variables, and $\eta : R \rightarrow \{+, -\}$ presents the labels of relations. For a CLD (V, R, η), V is partitioned into three disjoint sets of stocks S, flows F, and auxiliaries A based on the type of variables, i.e., $V=A \cup S \cup F$. We denote a CLD with the labeled sets of variables and relations to be a labeled CLD, i.e., $CLD'=(A, S, F, R, \eta)$.

For the presented CLD in Figure 3, the result of the first step for generating labeled CLD is presented in Table 3. After assigning V to three different sets of elements, S, F, and A, the relations in R should be examined and divided into I and M. R is divided into two subsets of flows of items/materials M and information dependencies I where $M=R \cap (F \times S)$ and $I=R \setminus (F \times S)$. In Figure 3, *(number of infected people per week, number of people in ICUs)* $\in M$ is an items/materials

TABLE 4. Based on the assigned labels of every $v, v' \in V$ and the link $r = (v, v') \in R$, the following constraints and modifications are required. ✓ indicates that it is possible to directly keep the r. Black cells show the relations that are not possible. For (s, s') and (a, s) as an information dependency between two stocks or information dependency between auxiliary variables and a stock, further refinement is required.

r	I			M
v \ v'	S	F	A	S
S	$insF, insM$	✓	✓	✓
F		✓	✓	✓
A	$insF$	✓	✓	

flow since *number of infected people per week* $\in F$ and *number of people in ICUs* $\in S$. So far, we have labeled CLDs with the set of elements in SFD relations, but one more step is required to ensure that the labeling adheres to the system dynamics criteria. The constraints supplied in Table 4 are taken into account while generating the models.

a: SD CONSTRAINTS CHECK

The given constraints in Definition 4 for the relations between elements of a stock-flow diagram should be considered while separating M from R. The main relationships are taken from generated CLDs and the rest are built on top of them. Therefore, some mentioned constraints, such as auxiliary variables are not being able to affect stocks directly, should be double-checked after assigning the stocks and flows roles. In Table 4, the possibility of types of relations (I or M) is mentioned after assigning stocks and flows. For instance, it is not possible to have an items/materials flow from a stock to an auxiliary variable. However, information dependency is possible between two stocks, given the fact that model refinement is required. Table 4 provides the guidelines for potential model refinements.

Consider the presented example in Figure 3 where the modeling is started by dividing the variables in the CLD into three sets of stocks, flows, and auxiliaries, see Table 3. The next step is to check the constraints on the set of items/materials flows and information dependencies. For instance, r=*(number of Covid deaths, number of vaccinated people)* $\in R$ is an information dependency, i.e., it affects the *number of vaccinated people* but does not directly add to the *number of vaccinated people*. As shown in Figure 5, a flow is inserted ($insF$). Flow $\widetilde{f}$=*vaccination rate* is added and as a result r_1=*(number of Covid deaths, vaccination rate)* and r_2=*(vaccination rate, number of vaccinated people)* are generated and inserted into R. Therefore, the set of relations (R), flows (F), information dependency I, and items/materials flows (M) are updated as follows: $R'=(R \setminus \{r\}) \cup \{r_1, r_2\}$, $F' = F \cup \{\widetilde{f}\}$, $I'=I \cup \{r_1\}$, and $M'=M \cup \{r_2\}$. Two notations in the constraint check step are the representors of potential actions, as follows:

- When there is a relation between two stocks, and it is an information dependency, or from auxiliary to a stock, *insF* insert a flow to comply with the SFD restrictions, see the example in Figure 5.
- When there is a relation of the type of information dependency between two stocks, *insM* inserts a flow directly between the two stocks. For instance, in Figure 5, a flow can also be directly inserted between the *number of Covid deaths* and the *number of vaccinated people*, indicating that one adds to the values of the other.

Given a labeled CLD', the corresponding SFD should be generated, where the transformation is defined in Definition 7. It is important to note that system dynamics diagrams are CLDs and SFDs, and we define and utilize the labeled CLDs only for concrete transformations of CLDs to SFDs.

Definition 7 (SFD Generation): An SFD $(A, S', F, I, mapf)$ *is defined based on the labeled* $CLD'=(A, S, F, R, \eta)$ *of a system where* $S'=\{S\} \cup \{\text{☁}\}$, $I=R \setminus F \times S$, *and* $M=R \cap (F \times S)$. *For* $f \in F$, *we define function mapf to represent the items/materials flows as follows*:

- $mapf(f) = (s, s')$ *if* $\exists_{s,s' \in S} : (f, s) \in M \wedge (f, s') \in M \wedge \eta((f, s)) = - \wedge \eta((f, s')) = +$.
- $mapf(f) = (s, \text{☁})$ *if* $\exists!_{s \in S} : (f, s) \in M \wedge \eta((f, s)) = -$.
- $mapf(f) = (\text{☁}, s)$ *if* $\exists!_{s \in S} : (f, s) \in M \wedge \eta((f, s)) = +$.

b: SIMULATION-READY SFD

After designing the CLD of a system and the corresponding SFD, the following steps are required to make the designed SFD ready to be simulated. By simulating the SFD, the values get updated at each step of time based on the SD-Logs and the underlying equations.

- Indicating the size of time steps for updating the values of variables, e.g., one day, and the number of steps to be simulated, e.g., set the simulation duration to 30 days.
- Inserting the underlying equations and the initial values for the stocks and dependent variables $\overleftarrow{V}$.
- Specifying the values of independent variables $\overrightarrow{V}$ which can be static values or can be set as external values from outside the system, e.g., temperature per day which is not getting affected by other system variables.

The concept of delay in SFDs, in which the value of a variable is updated after a couple of steps by other variables, is considered in Definition 5 by using the previous values at different steps of time in the underlying equations. For instance, the effect of the invested budget on the advertisement will appear after 6 months, i.e., the number of new customers per months will only increase 6 months after the advertisement. It means that 6 steps of a time delay of effects when the time window is considered to be one month.

For the provided example and the designed SFD model, if we simulate the designed model using a week as the time window and simulate it for a couple of weeks, the results of the simulation for some variables are presented in Table 5. The values of variables in each week are either calculated based on the previous values of other variables, i.e., using the equations, or come from domain knowledge and historical

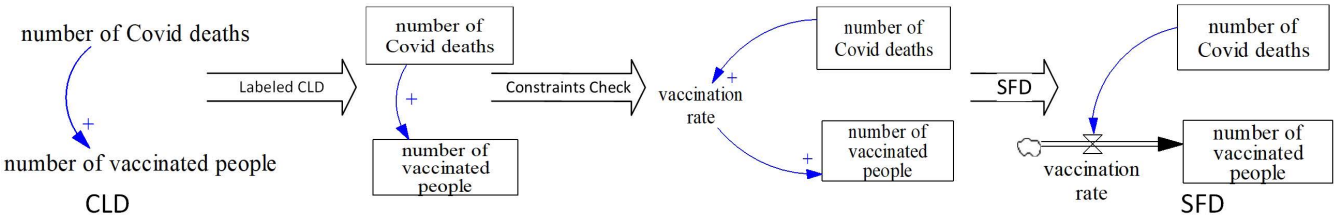

FIGURE 5. After labeling the example CLD in Figure 3, an example of changing the CLD to the SFD while checking constraints is shown. The detected constraint, i.e., information dependency between two stocks, is addressed utilizing the function *insF* to incorporate vaccination rate as a flow (left arrow). The conversion to an SFD using Definition 7 is also shown by the right arrow.

TABLE 5. A part of sample simulating the presented example in Figure 4. The time window is one week and the results includes 3 weeks and 4 variables.

Time Window (Week)	vaccination rate	number of vaccinated people	number of infected people per week	number of people in ICUs
1	1000	0	5000	68
2	1000	1000	4950	83
3	1150	3000	4845	107
⋮	⋮	⋮	⋮	⋮

information, e.g., the weather in each week. This example result indicates the format of the system dynamics simulation results and how we have to transform fine-grained event logs into coarse-grained process logs. We need to define process variables considering event log attributes, which are extractable over aggregate windows of time, such as a week here. We elaborate on the SD-log generation in Section IV-A.

C. PREDICTION METHODS

The analysis of sequences of real values and/or sequences of tuples of real values is often referred to as statistical and machine learning techniques [30]. The specific techniques to generate/predict the next values based on the previous values of influential variables are irrelevant to our approach. We use possible methods that allow us to select the most accurate models. Furthermore, machine learning techniques that act as black boxes and do not provide the underlying equations are not used, e.g., neural networks. We consider a list of possible models that can be trained using a data set of values and be represented as equations, e.g., different types of regression models, multivariate autoregressive, and curve fitting methods [31]. Therefore, the existing variables and their coefficients in the trained model are used to rebuild the equations. We propose a generic definition of a prediction method in Definition 8.

We categorize the statistical and machine learning techniques that we use based on the types of relations and equations that they are capable of discovering. The first category is linear equations, including regression models, and the second category is nonlinear equations, including curve fitting and support vector machines techniques [31].

Definition 8 (Prediction Methods): Let X be a matrix representing the values of variables for the set of variables V over steps of time. X is used as our training set. The values of variable $v^j \in V$ are in the j^{th} column in X which is a vector of values over steps of time. $\phi^j_X(.)$ is a trained prediction model which predicts the values of variable v^j, given the training set X. We denote IV^j to be the set of variables used by $\phi^j_X(.)$.

For instance, a simple linear regression model generates the values of observations of variable v^j for the given matrix of variables' values (X). To measure the accuracy of the trained prediction method for a variable, we use *Mean Absolute Error (MAE)*. For each set of observations of variables in X and the corresponding generated observations of v^j ($\hat{v^j}$), the generated values are compared with the real observations, i.e., $MAE(v^j, \hat{v^j}) = \sum_{i=1}^{n} |\frac{v^j_i - \hat{v^j}_i}{v^j_i}|$.

IV. GENERATING SYSTEM DYNAMICS MODELS OF BUSINESS PROCESSES

In this section, we explain our main approach for the generation of system dynamics models from fine-grained event logs. As Figure 6 illustrates, we transform an event log into a sequence of measurable performance variables representing the process at a higher level, referred to as SD-Logs. The performance questions in the context of scenario-based analysis are obtained during the preprocessing step. For example, how does an increase in the number of arrivals influence the average waiting time in the process? Then, over time, we extract the possible measurable variables associated with the questions. The SD-Log is formed by the computed values of these parameters throughout the selected time period. SD-Logs are a good fit for creating aggregated process models (CLDs). Later, the simulation models of the generated CLDs in the form of SFDs are designed and used for performing decision-making scenarios and capturing the effects of changes in processes over a period of time.

Compared to conventional modeling approaches that let users design the relations between the process variables, our approach is based on exploiting the generated SD-Logs to detect possible relationships between the process variables.

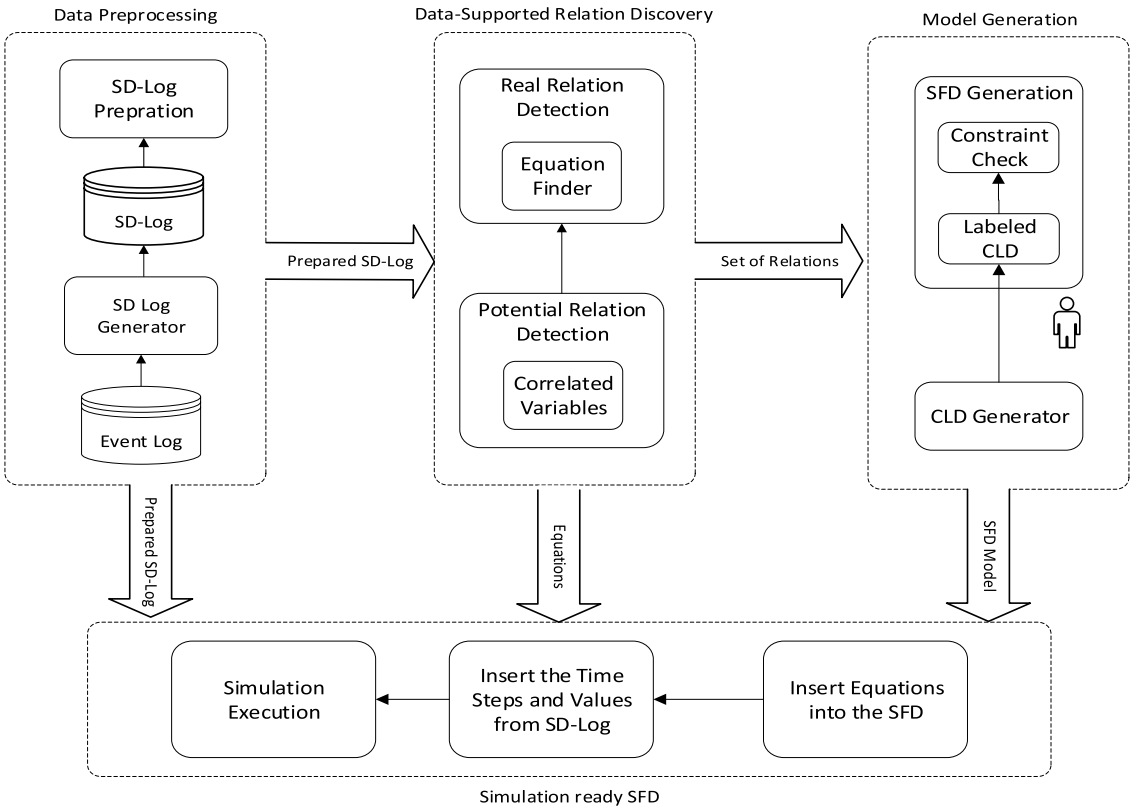

FIGURE 6. The main approach includes the SD-Log generation, relation detection, and the discovery of the type and direction of the relations. Our approach generates causal-loop diagrams (CLD) of a process over time and supports transforming them into stock-flow diagrams (SFD).

However, using the discovered relations in practice for performing a quantitative simulation requires knowledge about the type and direction of the relations, e.g., the number of infected people per day affects the arrival rate of the hospitals, not vice versa. A relation between two variables has three aspects: a type, a direction that indicates which variable affects the other, and a time step where the effect of one is visible on the other. The type of relation can be *positive/negative linear/nonlinear*. For instance, there is a strong relationship between the values of the arrival rate per hour and the average waiting time per hour in a process. The type of this relation can be negative/positive and linear/nonlinear. The direction of the effect is from arrival rate to average waiting time, which means the hourly arrival rate influences the hourly average waiting time. Meanwhile, the effect of increased arrivals may only be visible with some delay, e.g., after a couple of hours in the average waiting time, which shows the direction of relations in time.

In order to identify the type and direction of relations, we use multiple statistical and machine learning methods to automatically search for the possible equations. These equations define the values of one variable based on the values of other variables over time. The equations indicate the direction of the relations and their types. Discovered directions are used to build the system dynamics simulation models. First, a conceptual model (CLD) is generated, and after inserting the values from SD-Logs and the discovered equations, the ready-to-simulate models (SFD) are formed. The abstract steps of the approach for detecting the relations between variables are as follows:

1) The corresponding SD-Log is generated, given an event log and a time window.
2) The correlation of the values over time with other variables is assessed for each variable in the corresponding SD-Log. It should be noted that the variables' lagged correlation is also investigated to see whether the strongest relationship existed in the shifted time window.
3) Each variable's strong relationships are extracted.
4) Given a set of relations for each variable, different models are trained to predict the values of the variables, and the best equations are chosen based on prediction error.
5) The chosen equation for each variable demonstrates the existing strong relationships between the variables and their signs, i.e., the signs are derived from the coefficients.
6) The final set of relations is created and can be converted into CLDs to support SFDs later on.

Step 1 is explained in detail in Section IV-A. Steps 2 to 5 are the abstract level of the presented Algorithm 1 where the

function to detect and discover the relations is Definition 11 (step 2 and 3) and Definition 10 used for shifting the variables values. Section IV-C represents step 6. To demonstrate the approach, we use a real-world event log of a business process, i.e., BPI Challenge 2017, as an example throughout this section. The example provided in Section III was intended to demonstrate the system dynamics concepts in general.

A. PREPROCESSING

In this section, we describe the SD-Log generation step in detail and prepare the generated data for training models to identify the equations. We change the perspective and level of describing a process to a quantitative and aggregated level. We define aggregated variables over a certain length of time ($\delta \in \mathbb{N}$) instead of extracting and computing process variables at the instance level. We define and extract variables that describe the process over time, such as the average arrival rate of cases per day. The new coarse-grained process log is referred to as an SD-Log. Definition 9 defines the SD-Log, which is generated given an event log, a set of aggregated process variables, and a time window.

1) SD-LOG GENERATION

An event log is the starting point of any analysis in process mining. Therefore, the possible process variables are highly dependent on the available data in the event log. In our approach, we consider the basic attributes of event logs, which are defined in Definition 1. Hence, time-related performance variables w.r.t. events, cases, resources, and activities, e.g., the average service time of a case/activity per day, can be generated [12].

Table 6 represents the possible combination of mathematic aggregation functions, process aspects, and performance indicators that can be extracted from a standard event log. For example, applying the average function ($AF = Average$) to the number ($IN = Number$) of activities ($AT = Activity$) in an event log is not possible. The valid combinations form the set of process variables V to generate coarse-grained process logs.

Consider that the first event in the event log (L) in Table 2 w.r.t. timestamp starts at time $p_s(L)$, and the last event is completed at time $p_c(L)$. Given a time window $\delta \in \mathbb{N}_{\geq 0}$, there are $k = \lceil p_c(L) - p_s(L)/\delta \rceil$ subsequent time steps in the event log for the time window δ.

For all the *aggregation functions (AF)*, the set of *performance indicators (IN)*, and the set of *process aspects (AS)*. The set of process variables is the set of valid tuples. We denote $\mathcal{V} = AF \times IN \times AS$ based on Table 6. Given the defined variables, the calculation per each time window in the event log is implemented. For instance, $\frac{\sum_{i=1}^{|\overline{L}|} p_c(\sigma_i) - p_s(\sigma_i)}{|\overline{L}|}$ is how the value of *average time in the process* for cases based on tuple *(Average, Time in process, Case)* is calculated in each time window, i.e., the event log is divided, and the value is for each part. $p_s(\sigma_i)$ and $p_s(\sigma_i)$ are the start time of case σ_i and complete time of the cases, respectively. $\overline{L}$ is the set of cases in the event log L, and $|\overline{L}|$ is the number of cases in the event log.

The performance indicators are designed to be comprehensive to cover all the possibilities. Therefore, the service time is the time between the start and complete timestamp, the waiting time is the time between the complete timestamp of the previous activity and the start timestamp of the next activity, and the time in the process is the aggregation of waiting and service time. Note that if the aggregation function sum is considered, then the overall time for each of the performance indicators is calculated over each period of the time.

Given event log L, set of variables V as can be defined based on Table 6, and a window δ, the event log is transformed into an SD-Log, defined in Definition 9.

Definition 9 (SD-Log): Let $L \subseteq \xi$ be an event log, 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 L and δ is $sd_{L,\delta} \in \{1, \ldots, k\} \times V \rightarrow \mathbb{R}_{\geq 0}$, i.e., $sd_{L,\delta}(i, v)$ represents the value of the process variable $v \in V$ in the i^{th}-time window ($1 \leq i \leq k$).

If L and δ are clear from the context, we omit them and write sd. Given sd and $v \in V$, we write $\Pi_v(sd) \in \mathbb{R}^*$ returning the sequence of values $\langle x_1, \ldots, x_k \rangle$ for variable v. Furthermore, π_i returns the i^{th} value in a sequence, for instance, $\pi_i(\Pi_v(sd)) = x_i$.

Each process variable is mapped onto a sequence of real numbers, and each real number is computed over the event log and focuses on the valid combination of aspects, performance indicators, aggregation function, and a time window, e.g., the number of arrived cases in one day. Consider Figure 1 where we project an event log to a fixed period of time and for each period calculate the process variables.

Figure 7 shows the meta model of SD-Logs. An SD-Log has one or more process variables, and one or more time steps, and for each process variable at a time step there is one value. The details of the selection and calculation of process variables are presented in [12].

The sample generated SD-Log for the event log BPI Challenge 2017 [32] is shown in Table 7. The arrival rate, average service time, finish rate, and the number of unique resources are the process variables. The values of the process variables are calculated per day, i.e., $\delta = 1$ day.

2) SD-LOG PREPARATION

The presented values of process variables in SD-Logs are used to discover their possible relations and underlying mathematical equations. Therefore, the quality of the data determines the accuracy of the discovered equations and relations. Furthermore, as proposed in [24], the time steps in which the process does not have executions are detected. To do so, time series models such as *ARIMA* models are applied to the process variables in the generated SD-Logs. Based on the best-trained model, i.e., the minimum error, the regular patterns of process variables are discovered. For the details of the approach, we refer to [24]. The unexpected inactivity steps

TABLE 6. Using various Aggregation Functions (AF) on Performance Indicators (IN) for various Aspects (AS). The possible combinations marked as True. ⊥ denotes that no aggregation function is used.

Validator	IN												
	Value	Number					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
⊥	False	True	False	True	True	True	False	False	False	False	False	False	False

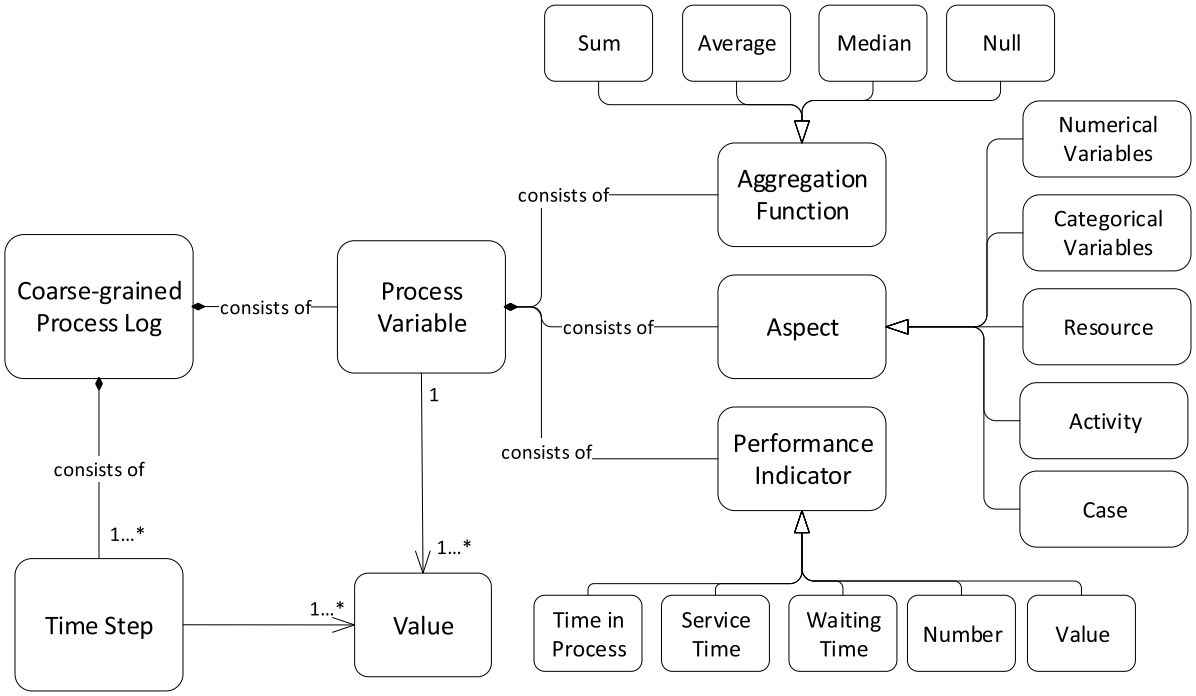

FIGURE 7. The coarse-grained process logs meta model Process logs consist of process variables and time steps. Each process variable and specific time step has one value. The process variables are a combination of different aggregation functions, aspects, and performance indicators.

TABLE 7. A part of an SD-Log generated for the general process of BPI Challenge 2017 event log over daily time window for 5 steps (day). The SD-Log includes 5 process variables, e.g., *Number of cases in the process*.

Time Window (Day)	Arrival rate	Finish rate	Number of unique resource	Average service time per case	Number of cases in the process
1	36	9	37	2643.102	27
2	33	0	36	3220.305	60
3	85	62	39	2099.366	83
4	68	46	41	2290.759	105
5	99	62	39	2056.365	142
⋮	⋮	⋮	⋮	⋮	⋮

in the process are removed, i.e., no weekends or lunchtime, which directly reduces the anomaly and potential noise within the real-world data. When generating the SD-Logs, regular inactivity of a process is also taken into account, e.g., in a business process with 5 working days, for δ=1 *day*, the corresponding rows for the weekends, such as 6^{th} and 7^{th} rows, are removed.

B. DATA-SUPPORTED RELATION DISCOVERY

Without considering the historical executions of processes, a sample CLD can be designed with background knowledge for the whole process, such as Figure 8.

As a process expert, the list of process variables is considered, and the relationships between them are later defined based on experience and common sense. These are the relationships that are always observed and examined in simulation models of business processes that have more cases in the process, longer queues, and thus a longer average waiting time.

For instance, it is expected that the average service time affects the finish rate as shown with the negative sign (−), or that an increase in the arrival rate per step of time will increase the number of cases in the process shown by +. The provided information supports the generation of these

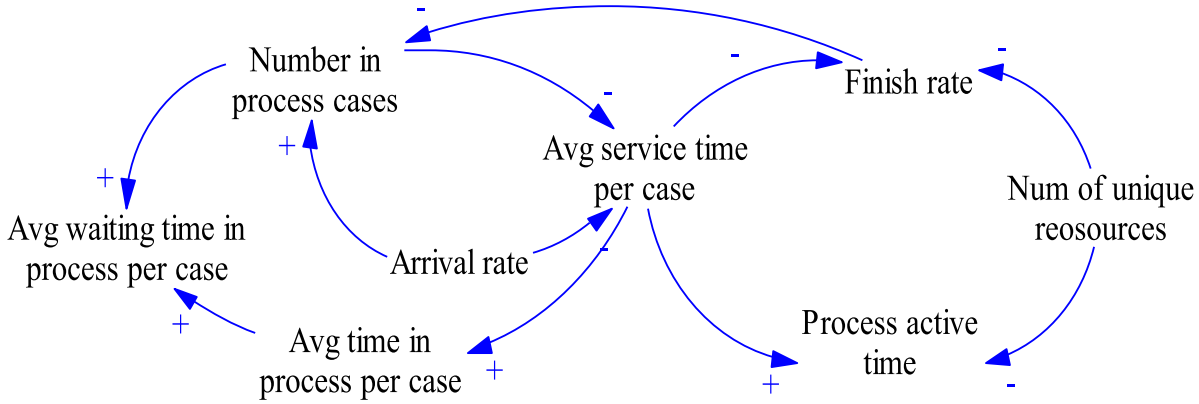

FIGURE 8. The designed conceptual model for the general process using common senses, e.g., the increase in the arrival rate per day will increase the number of cases waiting in the process. The CLD is designed in conventional modeling for the overall process, however, the data extracted from the event log can reject these hypotheses, e.g., the arrival rate does not have any effects on the number of cases in the process per day.

Process variable / Time step	v^1	v^2	...
1	v_1^1	v_1^2	...
2	v_2^1	v_2^2	...
...	...	...	...
k	v_k^1	v_k^2	...

FIGURE 9. A sample example of the shift function and finding the best shift function to check the lagged correlation between values of two process variables. In this example, the shift size of one is checked to see whether there is a high correlation between the values of v^1 and the values of v^2 in the second time step.

models. It results in valid models ensuring that all the possible relations presented in previous executions of processes are captured.

1) POTENTIAL RELATION DETECTION

To generate a CLD of a process based on the provided SD-Logs representing the process or examine the user-based generated CLD, the fact that process variables affect each other over time is used. To do so, the first step is to identify the relationships between variables and the type of relations. Calculating the linear and nonlinear correlation between values of the variables in SD-Logs is the backbone of discovering any possible relations. In our approach, both linear and nonlinear correlations using *Pearson correlation* and *Distance correlation* techniques in [33] are calculated.

The effect of changes in one of the variables can be seen after a couple of steps, e.g., the effect of an increase in the number of arrived cases per hour will appear after 3 steps of time (3 hours) in the average waiting time of the customers. These delays of effects in time indicate other properties of the detected relations that we refer to as a shift in time. The shift function in Definition 10 shifts the values of every two sets of variables. The function is used to identify the best shift size in time using the defined *fbs* function in Definition 11. The best size of shifting values of variables represents the strongest relations between the shifted values of variables and the direction of relations in time, i.e., shift size. We demonstrate an example regarding the concept of shift and finding the best shift size functions in Figure 9.

It is necessary to preserve a sufficient number of values corresponding to time steps when assessing the relations between the values of variables at different steps of the time. Assume $s \in \mathbb{N}$ as the maximum possible shift in the time windows to look for the cause and effect between process variables where $s \leq k(1-\theta_{sd})$, θ_{sd} is the minimum percentage number of values that we are willing to use, and $k \in \mathbb{N}$ is the number of values for each variable presented in the SD-Log.

Definition 10 (Shift Function): Let $s \in \mathbb{N}$ *be the maximum possible shift. We define function* $Shift_i : \mathbb{R}^* \times \mathbb{R}^* \rightarrow \mathbb{R}^* \times \mathbb{R}^*$, *such that for a given shift size* $0 \leq i \leq s$, *shifts the values of two sequences for* i *steps. For* $\sigma^1 = \langle x_1^1, \ldots, x_k^1 \rangle \in \mathbb{R}^*$, $\sigma^2 = \langle x_1^2, \ldots, x_k^2 \rangle \in \mathbb{R}^*$ *and shift* i, $Shift_i(\sigma^1, \sigma^2) = (\sigma'^1, \sigma'^2)$ *where* $\sigma'^1 = \langle x_1^1, \ldots, x_{k-i}^1 \rangle$, $\sigma'^2 = \langle x_{i+1}^2, \ldots, x_k^2 \rangle$.

Consider v^1 and v^2 as the arrival rate and the number of waiting cases per day, for the shift size of i, $Shift_i(\Pi_{v^1}(sd), \Pi_{v^2}(sd)) = (\langle \pi_1(\Pi_{v^1}(sd)), \ldots, \pi_{k-i}(\Pi_{v^1}(sd)) \rangle, \langle \pi_{i+1}(\Pi_{v^2}(sd)), \ldots, \pi_k(\Pi_{v^1}(sd)) \rangle)$.

Note that $k = |\Pi_{v^1}(sd)| = |\Pi_{v^2}(sd)|$. In Table 7, for instance, $\Pi_{v^1}(sd) = \langle 36, 33, 85, 68, 99, \ldots \rangle$ and $\Pi_{v^2}(sd) = \langle 9, 0, 62, 46, 62, \ldots \rangle$ are the values of two variables $v^1 =$ *arrival rate* and $v^2 =$ *finish rate*, applying $Shift_2$ will result in $\langle 36, 33, 85, 68, 99, \ldots \rangle$, and $\langle 62, 46, 62, \ldots \rangle$, for v^1 and v^2, respectively.

Definition 11 (Find the Best Shift): Function $Corr: \mathbb{R}^* \times \mathbb{R}^* \rightarrow [-1, 1]$ *calculates the correlation between two sequences of real values. Let* $s \in \mathbb{N}$ *be the maximum shift size, we define* $fbs_s: \mathbb{R}^* \times \mathbb{R}^* \rightarrow \mathbb{N}$ *to return the best possible shift size. Function* $fbc_s: \mathbb{R}^* \times \mathbb{R}^* \rightarrow [-1, 1]$ *returns the maximum value of correlation between two sequences for the given maximum shift size* $s \in \mathbb{N}$. *For* $\sigma^1, \sigma^2 \in \mathbb{R}^*$, $fbc_s(\sigma^1, \sigma^2) = (Corr(shift_{fbs_s(\sigma^1,\sigma^2)}(\sigma^1, \sigma^2)))$, *where* $Corr(shift_\tau(\sigma^1, \sigma^2))$ *is the correlation and* $fbs_s(\sigma^1, \sigma^2) = \underset{0 \le \tau \le s}{argmax}(Corr(shift_\tau(\sigma^1, \sigma^2)))$ *returns the best shift size.*

It should be noted that the *max* and *argmax* functions return a set of values if there is more than one maximum value. In this case, we consider the highest correlation value with the smallest shift. For instance, for v^1=*arrival rate* and v^2=*average waiting time* as two process variables in an SD-Log, $fbs(\Pi_{v^1}(sd), \Pi_{v^2}(sd))$=3 shows that the values of variables have the maximum value of correlation after 3 shifts in steps, e.g., the value of waiting time per hour is highly correlated with the number of the arrived cases from the previous 3 hours.

2) REAL RELATION DETECTION

Calculating the correlation identifies potential relationships between variables. Assessing the correlation among variables shows whether a relationship exists or not. The direction of effect, i.e., which variable influences the others, is not clear. The method in Definition 8 is a general form of describing the predictive methods that can be applied to a set of values to predict the values of one of them using others. These methods enable us to find the possible existing quantitative relations between the variables, i.e., the underlying equations, and their directions. Note that in models based on curve fitting, the interaction with the user is introduced in order to identify the potential relations, e.g., quadratic shape, and later the algorithm automatically fits the curve and measures the error. The complete relation detection module is then formed using the discovered relations based on the equations, as demonstrated by Algorithm 1. The inputs of Algorithm 1 are the corresponding SD-log $sd_{L,\theta}$ and the list of variables inside $sd_{L,\theta}$ as V for a given event log L and time window θ.

Algorithm 1 starts with an SD-Log of a process along with the maximum shift size $s \in \mathbb{N}$ and a threshold to consider a relation between two variables strong (θ_{rel}). In the relation detection algorithm for each pair of parameters in the SD-Log, the shift function is applied repeatedly, bounded by the maximal possible shift s. The maximum value of the correlation is compared with the threshold θ_{rel} to assess how strong the relationship is. Therefore, the relation is considered as a potential relation inside the process.

For instance, for the process variable, arrival rate in the algorithm, there is a high correlation between the average waiting time and the arrival rate, i.e., $fbc_s = +0.8$ and $fbs_s = 0$, the average waiting time is added to the set of influential variables (relations) for the arrival rate. $fbs_s = 0$ indicates that the maximum correlation is in the same time window and without any shift. Later in Algorithm 1, the set of related variables to the arrival rate is used to train different statistical and machine learning models (Φ) where the one with minimum prediction error is selected as the prediction model for that variable.

Algorithm 1 Relation Detection Algorithm

Input: The set of process variables V and the corresponding SD-Log sd
Input: Maximum possible shift s, the threshold of a strong relation θ_{rel}, and the set of prediction models Φ
Output: The set of discovered relations R

```
Create a set of relations R;
foreach v^m ∈ V do
    foreach v^n ∈ V do
        bs=fbs_s(Π_{v^m}(sd), Π_{v^n}(sd)); The best shift size
        corr=fbc_s(Π_{v^m}(sd), Π_{v^n}(sd)); The highest correlation value
        if corr ≥ θ_rel then
            Add (v^n, bs) to the relation's set for v^m;
        else
            return null;
        end
    end
    foreach prediction model ∈ Φ do
        Train prediction model for v^m;
        Return the trianed model (φ^{v^m}_{sd}(.));
        Predict values of v^m (v̂^m);
        Add MAE(v^m, v̂^m) to the set of errors for prediciton models for v^m;
    end
    Return the set of variables in the training model (IV^{v^m}) with the minimum errors;
    foreach v ∈ IV^{v^m} do
        Add (v, v^m) into R;
    end
end
return R;
```

$$number\ of\ case\ in\ process_i = 0.72 * arrival\ rate_i - 1.2 * unique\ resources_i \quad (3)$$

The variables inside the selected equations for each variable form the real relations, their types, and directions. For instance, consider Equation 3 to be the discovered equation for the variables *number of cases in the process* in the example BPI Challenge 2017 event log. Therefore, Algorithm 1 will return (*arrival rate*, *number of cases in the process*) $\in R$ as one of the real relations indicating that the arrival rate influences the number of cases in the process in a positive manner. Note that bidirectional relations between two variables are possible to appear in the final list of relations. We will discuss

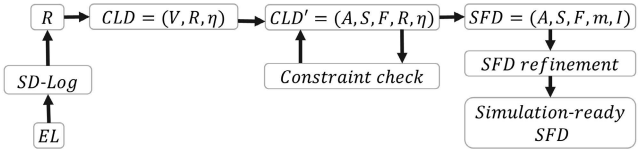

FIGURE 10. The steps in model generation phases starting from an event log to the final simulation-ready SFD model.

such bidirectional relations while generating the CLDs in Section IV-C.

The variables inside the selected equations for each variable form the real relations, their types, and directions. Consider the discovered equation for the variables *number of cases in the process* in the example BPI Challenge 2017 event log, Equation 3. As a result, Algorithm 1 will return *(arrival rate, number of cases in the process)*$\in R$ as one of the real relations, indicating that the arrival rate has a positive influence on the number of cases in the process. Note that bidirectional relations between two variables are possible to appear in the final list of relations. We will discuss such bidirectional relations while generating the CLDs in Section IV-C.

Definition 12 (Retrieve Variables of Relations): Let $\mathcal{V}$ *be the universe of variables,* $\mathcal{V} \times \mathcal{V}$ *be the universe of relations. for* $r \in \mathcal{V} \times \mathcal{V}$, $set(r)$ *retrieve the set of variables in* r. *For example, given* $r=(v^1, v^2)$, $set(r)=\{v^1, v^2\}$.

It is possible that not all the variables in the set of process variables V are in the selected relations. Consider the number of unique resources to be fixed throughout the time steps. Therefore, there is no relationship with other process variables. We need to retrieve the set of variables in the discovered relations R for the CLD generation. We use function Definition 12 to retrieve variables in the model generation phase.

C. MODEL GENERATION

We discovered all the strong relations among the process variables supported by their real values. These relations are the representation of the process conceptual model over time. We use the CLD notation of system dynamics to demonstrate the process at an aggregated level. A CLD of a process illustrates the conceptual model of the process, $CLD=(V, R, \eta)$. The next step is to design the SFD representing the process based on its CLD, i.e., conceptual model. Therefore, process variables and their relations in the CLD should be transformed into the elements in the SFD. The steps are labeling the process variable and the relations, checking the constraints, and inserting the equations for the simulation. Figure 10 as an overview represents the steps to move from the event log of a process and create executable SFD models.

1) GENERATE CONCEPTUAL MODEL (CLD)

The discovered set of relations R as a result of Algorithm 1 for an SD-Log enables process owners to identify all the supported relations between the variables. These relations are directly converted into arcs and the engaged variables in the SD-Log are the nodes in the graph using the defined function in Definition 13. Discovered relations between the variables in SD-Log are later used for forming conceptual simulation models (CLDs).

Definition 13 (Generate CLD): Let $\mathcal{CLD}$ *be the universe of causal-loop diagrams, and* $\mathcal{V} \times \mathcal{V}$ *be the universe of relations. Function genCLD* $: 2^{\mathcal{V}\times\mathcal{V}} \rightarrow \mathcal{CLD}$ *generates the CLD of a process given the discovered relations based on the corresponding SD-Log. For the set of discovered relations* $R \subseteq \mathcal{V} \times \mathcal{V}$, $genCLD(R)=(V, R, \eta)$. $\eta : R \rightarrow \{+, -\}$, *where for* $r \in R$ $\eta(r) = -$ *if* $fbc_s(r) < 0$ *and* $\eta(r) = +$ *if* $fbc_s(r) > 0$, *and* $V = \bigcup_{r \in R} set(r)$.

Given set of relations R as the result of Algorithm 1, the set of corresponding process variables V, function *genCLD* in Definition 13, insert a unique node for all $v^n \in V$, and for all the relations $(v^n, v^m) \in R$ form an arc from v^n to the v^m.

For instance, using the corresponding SD-Log of the BPI Challenge 2017 event log with a time window of $\delta=1$ *week*, the designed sample CLD in Figure 8 by domain knowledge will be changed into Figure 11. The presented relations between the nodes are supported by the real values of their variables.

It should be noted that cycles and transitive dependencies in the detected relations are possible. For example, both directions of the relationship between the arrival rate and the number of cases in the processes appear in the R as the set of discovered relations. Consider the following R as a result of Algorithm 1: $R=\{$*(arrival rate, number of unique resources), (arrival rate, avg time in process), (number of unique resources, avg time in process),* $\ldots\}$, the transitive dependency among three variables is implemented to be automatically removed. Arrival rate already affects the number of unique resources, e.g., if more cases arrive, then more resources are assigned, and the number of unique resources affects the average time in the process for cases. Therefore, there is no need for a direct relation between the arrival rate and the average time in the process. However, the user can decide whether to keep or remove such relations in our approach. In that situation, the user's domain knowledge, as well as the higher coefficients in the relations, are considered to avoid the cycles.

2) GENERATE STOCK-FLOW DIAGRAM (SFD)

We generated CLDs of processes automatically, the next step is to label the generated CLD of an SD-Log and later convert it to SFD for simulation purposes. Labeling the variables for SFD is based on the defined *Assign Stocks and Flows*. It considers the type of the possible extracted variables from event logs, e.g., time aspect, or number-based values. It is highly dependent on the scenario of simulation and the variables, e.g., the number of unique resources, whether it acts as an accumulative variable can be both stock or auxiliary variable.

a: ASSIGN STOCKS AND FLOWS

The values of the process variables in SD-Logs and the possible external factors play decisive roles in assigning the

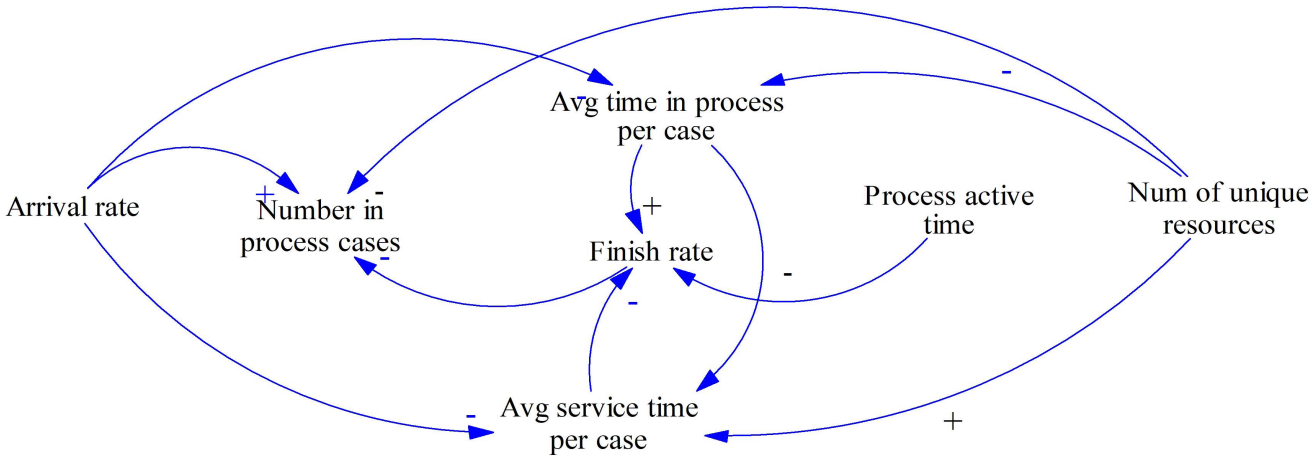

FIGURE 11. The designed conceptual model for the general process of BPI Challenge 2017 event log with a time window of one week. The presented nodes and arcs in the designed CLD are supported by the corresponding SD-Log, compared to the Figure 8 where the CLD is designed only based on the user background knowledge. For instance, the arrival rate has a negative effect on the number of cases in the process according to the extracted SD-Log and trained model.

elements of the stock-flow diagram. These types are based on different performance indicators (IN) used to define valid process variables in Table 6. The possible types of values are as follows:

- Number-based values are potential *stocks* since their value can be accumulated over time. They also can play the role of *flows* if they add or remove from the other variables over steps of time.
- Time-based (Duration) values are the possible *auxiliary* variables if they do not accumulate or are not considered to be countable.

For instance, the presented variables in Figure 11 can be assigned to the following sets, S=*{number of cases in process}*, F=*{finish rate, arrival rate}*, A=*{number of unique resources, average service time, ...}*. Note that the number of cases in the process can be an accumulative number over steps of time, and the arrival and finish rates over time add/remove from that. In this case, (*arrival rate, number of cases in process*) $\in M$ is an items/materials flow since *arrival rate* $\in F$ and *number of cases* $\in S$. For (*arrival rate, average service time*)$\in R$ is an information dependency and cannot be an items/materials flow based on Table 4. Furthermore, there is a link between *number of unique resources* and *number of cases in process*, where we assigned it to be a stock. Therefore, such information dependency is not possible, and we need to insert a flow (*insF*) or use the existing one. As a result, *(number of unique resources, finish rate)* $\in I$ and *(finish rate, number of cases in process)* $\in M$ will represent the same relation. CLDs are automatically generated from Algorithm 1 using SD-Logs and converted into the labeled CLD (CLD') where the stocks, flows, and auxiliaries are defined based on the scenario of interest. By applying constraints, SFDs are generated.

b: SIMULATION-READY SFDs

After assigning the stocks, flows, and auxiliaries and checking the system dynamics constraints, the generated SFDs should be enriched with values and equations for the simulation phase. To do so, three steps mentioned in Section III-B are performed including specifying the underlying equations as defined in Definition 8:

- The used time window to generate the SD-Log (sd), δ, is the set to be the time step size for the simulation, e.g., hourly, or daily.
- To specify the values of independent variables $\overrightarrow{V}$, for each step ($i \in \mathbb{N}_{\geq 1}$) the values are taken directly from the SD-Log (sd) such that for $v \in \overrightarrow{V}$, $v_i = \pi_i(\Pi_v(sd))$.
- For $v \in \overleftarrow{V}$, i.e., they have at least one incoming arc or influential variable, the values are simulated at each time step (i) as follows:
  - If $v \in S$, $v_i = v_{i-1} + \Sigma_{(v^m, v) \in M}(\eta((v^m, v)) * (v_i^m))$, where the initial values for stocks are filled by $\pi_1(\Pi_v(sd))$.
  - If $v \notin S$, the discovered equations in Algorithm 1 are inserted into the model, i.e., $\phi_{sd}^v(.)$ and $IV^v = \{v^m | (v^m, v) \in I\}$.

The designed SFD model can now be executed for the simulation in SD software where after validation of the results external variables can be added to the models. The main purpose of high-level simulation of processes is to perform the strategical analysis while including external and quality-based factors. For instance, the effect of training resources on overall process performance w.r.t. the cost of training can be modeled as SD with a higher level of confidence since the possible relations are supported by event logs of the process. Another example is the effect of a specific type of

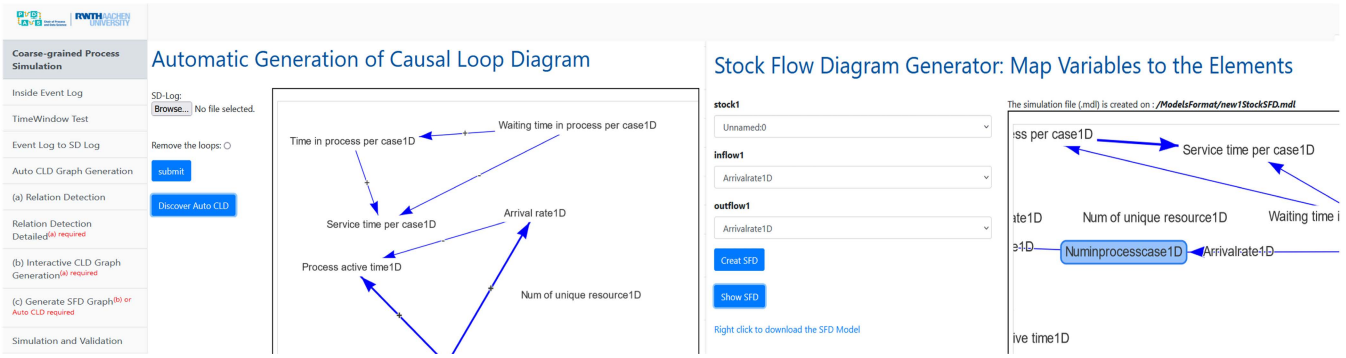

FIGURE 12. The screenshot of the tool. The CLD generator and the SFD design pages are shown.

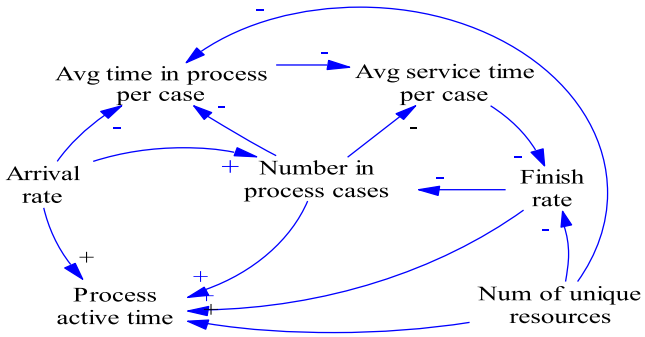

FIGURE 13. The discovered CLD of the BPI Challenge 2017 event log with 1 *day* time window.

advertisement on the overall monthly arrival rate of cases in the process.

Our main goal in this work is to capture processes from different perspectives and at a higher level of aggregation. Our approach generates data-driven CLDs of processes directly from their fine-grained event logs. Therefore, it is important to consider that although the majority of the steps from SD-Log to SFD model generation are automated, user interaction is still required in modeling tasks, assessing relations, and inserting the equations.

V. EVALUATION

The proposed approach is able to generate system dynamics models of processes given their event logs as an input, as well as assess the designed models according to the domain knowledge using the event logs. So far, we have presented steps of the approach in practice while explaining the methods in the paper. The evaluation has two main goals: (1) to demonstrate the application of the approach on real data while using the designed and developed tool, and (2) to demonstrate the use of system dynamics modeling in addition to the discrete event simulation technique for business processes. The implementation of SD-Log generation, the equation finder, and the CLD generator are publicly available as an integrated web application.[1]

[1]https://github.com/mbafrani/PMSD

We extended the SD-Log generation and the relation detection only based on correlation as presented in [25], with the automatic CLD generation and designing of SFDs. Figure 12 shows a screenshot of the web interface of the tool. In addition to the diagrams shown in the tool, the corresponding "mdl" files suited for system dynamics tools and intermediate results, such as discovered equations, are also generated.

A. TOOL-SUPPORTED CLD AND SFD DESIGN FOR A REAL EVENT LOG

We continue with the provided running example for the BPI Challenge 2017 event log and evaluate the results to design a simulation model from the SD-Logs. By applying the approach to the SD-Logs with a time window of 1 *day*, we discover the CLD of the process, as presented in Figure 13.

The set of discovered relations is used as the input of *genCLD*. A designed CLD representing the conceptual model of the process in a daily manner is created based on the detected equations. The outcome of Algorithm 1 is presented in Table 8 illustrating the relations between the process variables. For instance, value +1 in the first row (*Arrival rate, Number of cases in process*) indicates that there is a strong positive relationship between the two variables, i.e., with an increase in the daily arrival rate, the number of cases in the process per day will also increase.

Given the fact that the discovered relations based on the equations include loops, we have adjusted the relations (values in parentheses). For instance, based on the best fitted equations, the following relations (edges in the CLD) are discovered: (*Arrival rate, Finish rate*), (*Arrival rate, Avg time in process per case*, and (*Avg time in process per case, Finish rate*). In this case, we remove (*Arrival rate, Finish rate*) since the average time in process per case gets affected by the arrival rate and at the same time affects the finish rate.

Having obtained the labeled CLD, we transform it to SFD for simulating and validating the results. To do so, first, we start with the stocks and the flows and later form the relations based on Table 4 using functions *insF* and *insM*. For instance, (*Number of unique resources (Auxiliary), Finish rate (Flow)*)$\in I$ is an information dependency, therefore, it is an arc in SFD, and *(Arrival rate, Number of cases in*

TABLE 8. The detected relations (edges) in the conceptual model of the BPI Challenge 2017 event log using a time window of 1 *day*. The labeled CLD is generated based on the labeling stocks and inflows and outflows shown in the table.

	Arrival rate (Inflow)	Finish rate (Outflow)	Number of unique resources (Auxiliary) (Independent variable)	Process active time (Auxiliary)	Avg service time per case (Auxiliary)	Avg time in process per case (Auxiliary)	Number of cases in process (Stock)
Arrival rate (Inflow)	0	+1 (0)	0	+1	-1 (0)	-1	+1
Finish rate (Outflow)	0	0	0	-1	0	0	-1
Number of unique resources (Auxiliary) (Independent variable)	0	+1	0	+1	+1 (0)	-1	0
Process active time (Auxiliary)	0	0	0	0	0	0	0
Avg service time per case (Auxiliary)	0	-1	0	+1	0	+1 (0)	0
Avg time in process per case (Auxiliary)	0	+1 (0)	0	+1	+1	0	0
Number of cases in process (Stock)	+1	-1	0	+1 (0)	-1 (0)	-1 (0)	0

TABLE 9. The modified equations by the user which are inserted into the SFD. Note that the arrival rate and the number of unique resources are considered to be independent variables, and their values get updated for each simulation step from the SD-Log.

$Finish\ rate{=}2.05 * Number\ of\ unique\ resources - 0.01 * Avg\ service\ time\ per\ case - 0.013 * Number\ of\ cases\ in\ process - 56.96$
$Number\ of\ cases\ in\ process = Arrival\ rate - Finish\ rate$
$Avg\ service\ time\ per\ case{=}76 * Number\ of\ unique\ resources - 1.48 * Number\ of\ cases\ in\ process - 2157.04$
$Process\ active\ time{=}230 * Arrival\ rate + 3.56 * Finish\ rate + 83.5 * Number\ of\ unique\ resources - 15783$

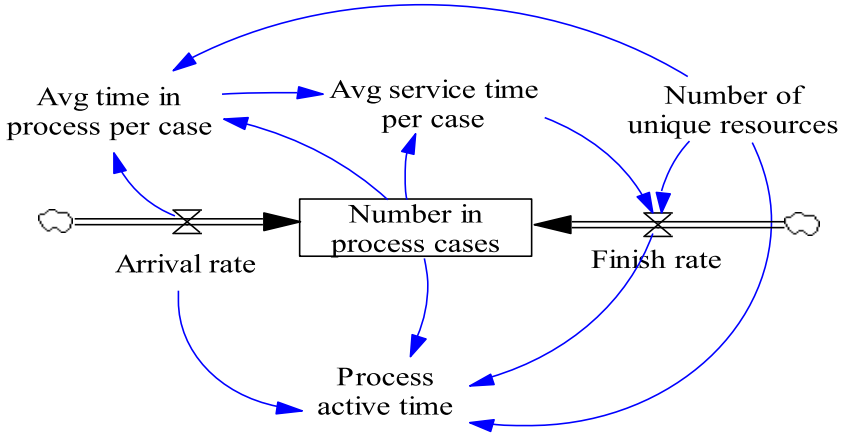

FIGURE 14. The designed SFD for BPI Challenge 2017 event log with the time window of 1 *day*.

process)$\in M$ is an items/materials flow. The designed SFD is presented in Figure 14.

The underlying equations for the dependent variables represented in Table 9 are used to make the designed SFD executable. Note that, as mentioned, adjustments to the equations are made. We run the simulation for 150 days and compare the results for the dependent variables, such as *number of cases in process* with the real values in the SD-Log, see Figure 15. It should be highlighted that the discovered relations, equations, and CLD models by the tool are potential insights into processes that can support the user. They direct the user on how to model in cases that prospective relations and equations are detectable using the implemented techniques. Furthermore, despite discrete event simulation, when modeling a process using system dynamics, the effects of variables are important. For instance, in [34], we were able to compare the DES simulation results and real event logs at the fine-grained levels, i.e., at case and event levels. For coarse-grained process simulation, the trends of the values are considered, not the exact values of the simulation.

B. THE USE OF SYSTEM DYNAMICS IN PRACTICE: MODEL REFINEMENT

The extracted process variables are highly related to the provided information and attributes in the event logs. In most cases, the ultimate goal is to assess the effects of other variables on the process at a higher level. These variables are

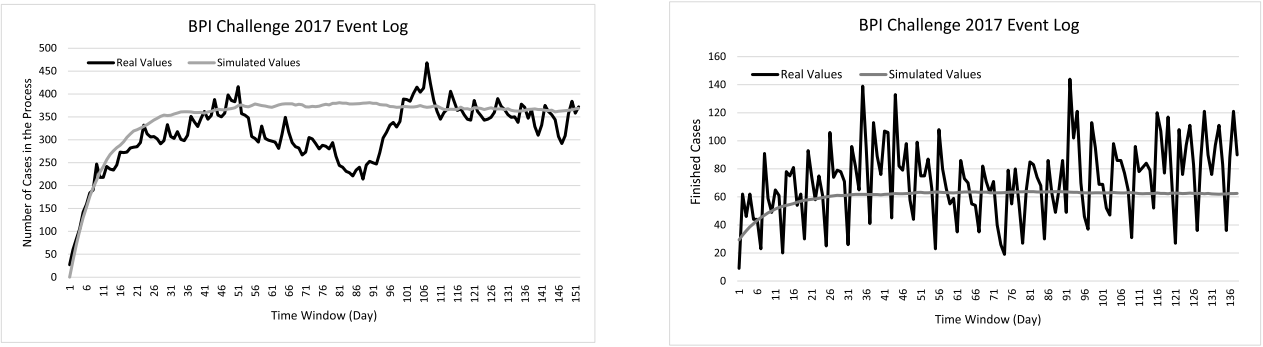

FIGURE 15. The comparison between real and simulated variables using the generated models based on SD-Logs for BPI Challenge 2017 with a time window of 1 *day*. The left graph represents the values of *number of cases in the process* over 150 *days*. The right graph represents the values of *finished cases* over the 150 *days*.

either external, i.e., not captured in the processes' event logs, or are hard to quantify, such as expertise of resources. Our presented approach enables the business owner to discover the relations between the existing process variables (based on event logs) and form aggregated simulation models.

At this step, the validity of the models can be evaluated using the SD-Logs and real values of process variables. Then, the valid models can be extended with different influential variables and what-if analyses can be performed. We do so for the sample SFD models of the BPI Challenge 2017 event log. We extend that with external factors to see their real effects on the process. The model is shown in Figure 16. The model exploits the effects of advertisement on the arrival rate of the cases (*Effect of advertisement, Expected arrival rate*), which the advertisement effects itself are also affected by the productivity and revenue of the process, i.e., (*Process productivity, Revenue*), and (*Revenue, Advertisement budget flow*). Moreover, the effect of the number of desired finished cases per day on the number of required resources is modeled, where the time to hire new resources is also considered. The simulation model is executed using the *Vensim*[2] software. The represented blue graphs show the simulated values of the variables. Note that variables such as advertisement delay of effects or time to hire can be adjusted based on the domain knowledge of the user. Also, the effects of the advertisement budget percentage on the arrival rate of the cases in the process can be examined.

We proposed a couple of scenarios that use the discovered model and can be run, including external variables in the processes. For instance, we designed the model in Figure 16 including the resource efficiency as external variables. It is able to simulate and track the effects of changes in the number of resources dynamically, w.r.t. the arrival rate, the efficiency of resources, and the number of desired finished cases per day. All of the designed models based on the approach and the presented tool (PMSD) are available for additional experimentation and evaluation details.[3]

[2]https://vensim.com

[3]https://github.com/mbafrani/PMSD/ModelsForEvaluation

C. DISCUSSION

When creating a generic approach for automatic system dynamics model creation, there are a few aspects to consider: (1) The quality of the data has a significant impact on the outcomes of the approach. As a result, most of the time, anomalies and non-stable behavior in the process make SD-Log extraction challenging, e.g., it is difficult to capture process variables over certain time steps. (2) Since the extra factors mostly have no data in event logs and only rely on the user's domain knowledge, it makes it difficult to validate and compare. However, the base models with the process variables from SD-Logs provide the potential for some level of confidence in the final models. (3) The application of statistical and machine learning models might lead to overfitting and under fitting issues. For example, when utilizing the regression method, there may be overfitting of the values, which is a disadvantage when simulating the values of the process variables. (4) In some cases, finding the equations for variables in relation to each other is impracticable. In this scenario, the user's domain knowledge is required for forming the relations and equations.

The final results for the decision makers are SFDs with the potential of preforming simulation models and assessing the effect of variables on each other. There might be some conditions in which refining the baseline models based on the extracted process variables is not possible. This can affect the applicability of the discovered SFDs. The purpose of this approach is not to make the modeling task completely automatic, since that contradicts the system dynamics principle. In system dynamics modeling and system thinking, capturing and simulating systems w.r.t. quantitative and external factors at higher levels is the main focus. Therefore, the role of the user in modeling still remains vital. Consider that the discovered loops and transitive relations in CLDs, as well as adjusting and forming the equations, are decided by the user given the scenarios of the simulation and states of processes. It is not a drawback but should be noticed.

As discussed, the SFD modeling task is user-dependent and comes from the scenarios that one is interested in. What we provide is the support for revealing the existing insights in the

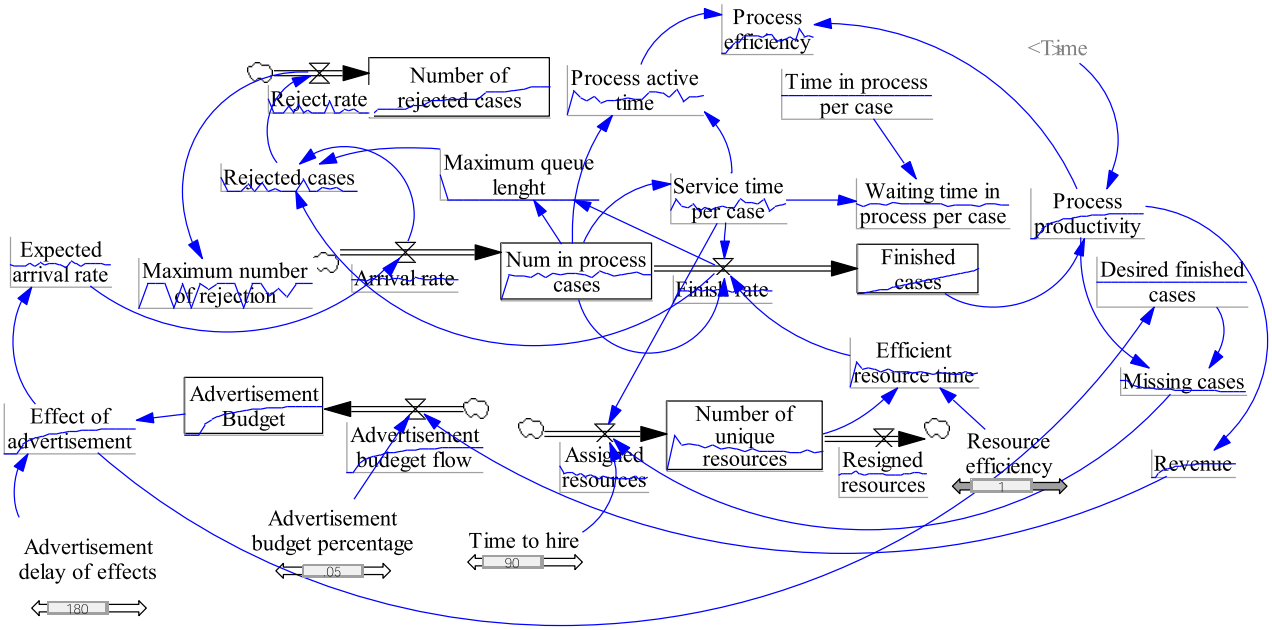

FIGURE 16. The extended SFD model is based on the evaluated model. It includes the effects of advertising based on the production rate and the arrival rate, as well as dynamic resource allocation in the process.

historical data considering the scenario. In our evaluation as a proof of concept, we address inserting domain knowledge into the SFD modeling task by considering one specific scenario, i.e., a what-if question. Then, develop the SFD on the basis of those questions. For example, in Figure 16, we developed a model to answer the questions, *what are the effects of the advertisement budget on the arrival rate of the process and how does it affect the number of required resources, which is also dependent on the hiring time?*. The number of rejected and handled cases (finished) is investigated, while two external factors of the advertisement and hiring process are considered. These models can be dynamically exploited by decision makers to find the desired balance of resources and advertisement budget with respect to the efficiency of their process, e.g., the number of accepted and rejected cases.

The core objective of our approach is to demonstrate the capability of modeling processes using system dynamics at various levels and to supplement modeling tasks with real-world process data in the form of event logs and process mining techniques. The identified relationships in the form of CLDs illustrate the causes and effects of the process variables on one another. This insight is used to identify the problematic process factors as well as the effective components that have to be investigated at the decision-making level while considering multiple factors outside business processes. For instance, the effects of the number of unique resources per day on the average time that cases spend in the process can be discovered and later used to improve the process. Another point to emphasize is that whereas traditional process simulation, such as detailed simulation using DES models, tries to mimic the details of the process and provide as close to the existing event data as possible, the goal of aggregated process simulation is to find the direction and high-level effects.

VI. CONCLUSION

Process owners require a platform to run what-if analyses for their processes. Process mining provides them with a wide range of techniques to discover the current status of their processes, which can be combined with simulation techniques for the scenario-based analysis of processes. Using simulation techniques system dynamics and process mining, we proposed an approach to support system dynamics modeling for business processes. Using simulation techniques system dynamics and process mining, Not only can the hidden underlying effects and relations at the instance level be detected, but also the existing equations that are used in the simulation models are discovered. Extracting higher levels of variables on top of event logs enables us to form aggregated simulation models that are able to consider external factors outside the process, e.g., the sickness rate of employees as resources in a process. As demonstrated by the evaluations, our approach is capable of uncovering hidden relations and constructing valid simulation models in which domain knowledge can also be applied. The presentable underlying equations between the process variables are discovered. In cases where discovering exact equations is not possible, we intend to incorporate the values generated by the taught machine learning algorithms into the models as our next step. We can ensure that possible relationships between the variables are captured. Furthermore, there are established scenarios for business processes that can be designed and used as default system dynamics models of processes, in which users are able to adjust them to their own scenarios.

ACKNOWLEDGMENT

The funding is both from Internet of Production and Alexander von Humbold.

REFERENCES

[1] W. M. P. van der Aalst, *Process Mining—Data Science in Action*, 2nd ed. Springer, 2016.
[2] J. Carmona, B. F. van Dongen, A. Solti, and M. Weidlich, *Conformance Checking—Relating Processes Models*. Springer, 2018.
[3] W. van der Aalst, A. Adriansyah, and B. van Dongen, "Replaying history on process models for conformance checking and performance analysis," *Wires Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 182–192, 2012.
[4] N. Tax, I. Verenich, M. La Rosa, and M. Dumas, "Predictive business process monitoring with LSTM neural networks," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Springer, 2017, pp. 477–492.
[5] A. Rozinat, M. T. Wynn, W. M. P. van der Aalst, A. H. M. ter Hofstede, and C. J. Fidge, "Workflow simulation for operational decision support," *Data Knowl. Eng.*, vol. 68, no. 9, pp. 834–850, Sep. 2009.
[6] W. M. P. van der Aalst, "Process mining and simulation: A match made in heaven!" in *Proc. 50th Comput. Simul. Conf., (SummerSim)*, Bordeaux, France, Jul. 2018, pp. 1–4.
[7] A. Rozinat, R. S. Mans, M. Song, and W. M. P. van der Aalst, "Discovering simulation models," *Inf. Syst.*, vol. 34, no. 3, pp. 305–327, May 2009.
[8] M. Camargo, M. Dumas, and O. González-Rojas, "Automated discovery of business process simulation models from event logs," *Decis. Support Syst.*, vol. 134, Jul. 2020, Art. no. 113284.
[9] W. M. P. van der Aalst, "Business process simulation survival guide," in *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*, 2nd ed. Springer, 2015, pp. 337–370.
[10] B. Dangerfield, "Discrete-event simulation and system dynamics for management decision making," Wiley, Chichester, U.K., Tech. Rep., 2014, pp. 29–51.
[11] J. Sterman, *System Dynamics: Systems Thinking and Modeling for a Complex World*. Cambridge, MA, USA: Massachusetts Institute of Technology. Engineering Systems Division, 2002.
[12] M. Pourbafrani and W. M. P. van der Aalst, "Extracting process features from event logs to learn coarse-grained simulation models," in *Advanced Information Systems Engineering* (Lecture Notes in Computer Science), vol. 12751. Springer, 2021, pp. 125–140.
[13] M. Pourbafrani and W. M. P. van der Aalst, "Hybrid business process simulation: Updating detailed process simulation models using high-level simulations," in *Research Challenges in Information Science* (Lecture Notes in Business Information Processing), vol. 446. Springer, 2022, pp. 177–194.
[14] M. Pourbafrani, S. J. van Zelst, and W. M. P. van der Aalst, "Supporting automatic system dynamics model generation for simulation in the context of process mining," in *Business Information Systems* (Lecture Notes in Business Information Processing), vol. 389. Springer, 2020, pp. 249–263.
[15] M. Pourbafrani, S. J. van Zelst, and W. M. P. van der Aalst, "Scenario-based prediction of business processes using system dynamics," in *Proc. Move Meaningful Internet Syst., Conf. Confederated Int. Conf., (CoopIS, ODBASE, C&TC)*, Rhodes, Greece, Oct. 2019, pp. 422–439.
[16] M. Dees, M. de Leoni, and F. Mannhardt, "Enhancing process models to improve business performance: A methodology and case studies," in *Proc. OTM Confederated Int. Conf. Move Meaningful Internet Syst.* Springer, 2017, pp. 232–251.
[17] M. de Leoni, W. M. P. van der Aalst, and M. Dees, "A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs," *Inf. Syst.*, vol. 56, pp. 235–257, Mar. 2016.
[18] V. Denisov, D. Fahland, and W. Aalst, "Multi-dimensional performance analysis and monitoring using integrated performance spectra," in *Proc. ICPM Doctoral Consortium Tool Demonstration Track Co-Located With 2nd Int. Conf. Process Mining (CEUR Workshop)*, Padua, Italy, vol. 2703, Oct. 2020, pp. 27–30. [Online]. Available: CEUR-WS.org
[19] Z. Toosinezhad, D. Fahland, O. Koroglu, and W. M. P. van der Aalst, "Detecting system-level behavior leading to dynamic bottlenecks," in *Proc. 2nd Int. Conf. Process Mining (ICPM)*, Oct. 2020, pp. 17–24.
[20] C. Fleig, D. Augenstein, and A. Maedche, "Designing a process mining-enabled decision support system for business process standardization in ERP implementation projects," in *Business Process Management Forum* (Lecture Notes in Business Information Processing), vol. 329. Springer, 2018, pp. 228–244.
[21] L. An and J.-J. Jeng, "On developing system dynamics model for business process simulation," in *Proc. Winter Simul. Conf.*, Dec. 2005, pp. 2068–2077.
[22] D. E. Bowles and L. R. Gardiner, "Supporting process improvements with process mapping and system dynamics," *Int. J. Productiv. Perform. Manage.*, vol. 67, no. 8, pp. 1255–1270, Nov. 2018.
[23] B. J. Angerhofer and M. C. Angelides, "System dynamics modelling in supply chain management: Research review," in *Proc. Winter Simul. Conf.*, 2000, pp. 342–351.
[24] M. Pourbafrani, S. J. van Zelst, and W. M. P. van der Aalst, "Semi-automated time-granularity detection for data-driven simulation using process mining and system dynamics," in *Proc. Conceptual Modeling 39th Int. Conf. (ER)*, 2020, pp. 77–91.
[25] M. Pourbafrani and W. M. P. van der Aalst, "PMSD: Data-driven simulation using system dynamics and process mining," in *Proc. Demonstration at 18th Int. Conf. Bus. Process Manage.*, 2020, pp. 77–81.
[26] J. D. Sterman, *Business Dynamics: Systems Thinking and Modeling for a Complex World*. New York, NY, USA: McGraw-Hill, 2000.
[27] D. Meadows, *Thinking in Systems: A Primer*, D. Wright, Ed. London, U.K.: Earthscan, 2008.
[28] E. Pruyt, *Small System Dynamics Models for Big Issues: Triple Jump Towards Real-World Complexity*. TU Delft Library, 2013.
[29] T. Binder, A. Vox, S. Belyazid, H. Haraldsson, and M. Svensson, "Developing system dynamics models from causal loop diagrams," in *Proc. 22nd Int. Conf. Syst. Dyn. Soc.*, 2004, pp. 1–21.
[30] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLoS ONE*, vol. 13, no. 3, Mar. 2018, Art. no. e0194889.
[31] A. Zielesny, *From Curve Fitting to Machine Learning*, vol. 18. Berlin, Germany: Springer, 2011.
[32] V. Dongen and B. F. Boudewijn, "BPI challenge 2017," Eindhoven Univ. Technol., Eindhoven, The Netherlands, Tech. Rep., 2017, doi: 10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b.
[33] P. Schober, C. Boer, and L. A. Schwarte, "Correlation coefficients: Appropriate use and interpretation," *Anesthesia Analgesia*, vol. 126, no. 5, pp. 1763–1768, May 2018.
[34] M. Pourbafrani and W. M. P. van der Aalst, "Interactive process improvement using simulation of enriched process trees," in *Proc. 2nd Int. Workshop AI-enabled Process Autom.*, 2021, pp. 1–15.

MAHSA POURBAFRANI was born in 1989. She received the B.S. and M.S. degrees in computer science from the Amirkabir University of Technology, Tehran, Iran, in 2014. She is currently pursuing the Ph.D. degree under the supervision of Prof. Wil M. P. van Der Aalst. She is a Research Assistant with the Data and Process Science Group, RWTH Aachen University. Her research interest includes process mining, which employs data science methods to turn data into actionable insights. The actions are taken using simulation, what-if analysis, and predictions in process mining regarding the performance metrics of processes. She is also a Scientist working on the "Internet of Production" project, which aims to combine process mining and machine learning techniques to support operations and decisions in production lines.

WIL M. P. VAN DER AALST (Fellow, IEEE) is a Full Professor with RWTH Aachen University, leading the Process and Data Science (PADS) Group. He is also the Chief Scientist at Celonis, part-time affiliated with the Fraunhofer FIT, and a member of the Board of Governors of Tilburg University. He has been the unpaid professorship positions at the Queensland University of Technology, since 2003, and the Technische Universiteit Eindhoven (TU/e). Currently, he is also a Distinguished Fellow of Fondazione Bruno Kessler (FBK), Trento, the Deputy CEO of the Internet of Production (IoP) Cluster of Excellence, and the Co-Director of the RWTH Center for Artificial Intelligence.