Supporting Automatic System Dynamics Model Generation for Simulation in the Context of Process Mining

Mahsa Pourbafrani[1], Sebastiaan J. van Zelst[1,2], and Wil M. P. van der Aalst[1,2]

[1] Chair of Process and Data Science, RWTH Aachen University, Germany
{mahsa.bafrani,s.j.v.zelst,wvdaalst}@pads.rwth-aachen.de
[2] Fraunhofer Institute for Applied Information Technology (FIT), Germany
{sebastiaan.van.zelst,wil.van.der.aalst}@fit.fraunhofer.de

Abstract. Using process mining actionable insights can be extracted from the event data stored in information systems. The analysis of event data may reveal many performance and compliance problems, and generate ideas for performance improvements. This is valuable, however, process mining techniques tend to be backward-looking and provide little support for forward-looking approaches since potential process interventions are not assessed. System dynamics complements process mining since it aims to capture the relationships between different factors at a higher abstraction level, and uses simulation to predict the effects of process improvement actions. In this paper, we propose a new approach to support the design of system dynamics models using event data. We extract a variety of performance parameters from the current state of the process using historical execution data and provide an interactive platform for modeling the performance metrics as system dynamics models. The generated models are able to answer "what-if" questions. Our experiments, using event logs including different relationships between parameters, show that our approach is able to generate valid models and uncover the underlying relations.

Keywords: Process mining · scenario-based predictions · system dynamics · what-if analysis · simulation

1 Introduction

Large amounts of event data are available in organizations, i.e., stored in information systems. Process mining provides the opportunity to exploit such data in a meaningful way, e.g., by discovering process models that describe the observed behavior in the organization, i.e., *process discovery* [2]. Furthermore, *conformance checking* [2] alongside process discovery assesses the level of similarity between the process model and the real executions of the process as captured in the event data. Moreover, *process enhancement* techniques improve the overall view of the process by extracting the information about the performance of the process [7,9]. For business owners, insight into their processes from different

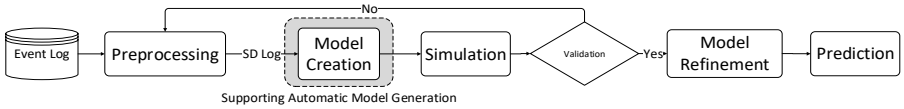


Fig. 1: Our proposed framework for using process mining and system dynamics together in order to design valid models to support scenario-based prediction of business processes in [10]. This paper focuses on the *automatic model generation*, i.e., the highlighted step.

angles, especially from the performance view is highly valuable. These insights into their processes provide a platform to look forward and improve their processes. Therefore, these insights can be used to fill the gap between the current state of the process's performance and its desired future state. Business owners need to be supported in long-term decision making. Different approaches in process mining for the purpose of predicting a process's future behavior have been introduced. Most of these techniques are suitable for short-term prediction and they act at the *process instance level*, e.g., what is the next activity for a specific customer [22]. Others are highly dependent on explicit knowledge about the detailed processes such as [16]. Furthermore, hidden effects exist among the involved factors in the simulation models, e.g., the effects of increasing the workload of resources on their speed of performing the tasks, or the relationship between the level of difficulty of a task with the number of assigned resources.

Meanwhile, system dynamics techniques are able to cover different effects including human aspects and model the nonlinear relations at an aggregated level. Such techniques try to provide a holistic model of the system and include all possible effects in the system over time. However, most simulation-based approaches, including system dynamics, highly rely on the users and their understanding of the system. In [10, 11], the idea of using process mining and system dynamics together at an aggregated level is presented which leads to designing the models including external factors. This approach generates system dynamics logs, i.e., a collection of measurable aspects from an event log. Then, the designed models are populated with the values of these measurable aspects referred to as SD-log. Hereafter, the validation step is performed to measure the similarity of the generated results by the model with the real values in the SD-log.

As shown in Fig. 1, the proposed approach depends on a modeling step which is based on the user insights into the system. In this paper, we propose a highly automated framework which supports businesses in an interactive manner for designing their simulation models. Our approach captures the influential factors in performance parameters and automatically generates the system dynamics models in order to predict the possible effects of future changes in the business processes. Afterward, these generated models can be populated with the values and the simulation and validation of the simulated results are possible.

Table 1: A simple Event log. Each row refers to an event.

Case ID	Activity	Resource	Start Timestamp	Complete Timestamp
1	Register	Rose	10/1/2018 7:38:45	10/1/2018 7:42:30
2	Register	Max	10/1/2018 8:08:58	10/1/2018 8:18:58
1	Submit Request	Eric	10/1/2018 7:42:30	10/1/2018 7:42:30
1	Accept Request	Max	10/1/2018 8:45:26	10/1/2018 9:08:58
2	Change Item	Eric	10/1/2018 9:45:37	10/1/2018 9:58:13
3	Register	Rose	10/1/2018 8:45:26	10/1/2018 9:02:05
...	...	...	...	...

The remainder of this paper is organized as follows. In Section 2, we introduce background concepts and basic notations used throughout the paper. In Section 3, we present related work. In Section 4, we present our main approach. We evaluate the proposed approach in Section 5. Section 6 concludes our work and discusses interesting directions for future work.

2 Preliminaries

In this section, we formalize the related concepts to our approach.

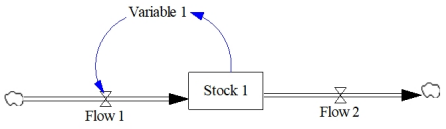
Historic data, captured during the execution of a company's processes, provide the starting point for process mining [2]. Table 1, presents a simplified sample event log. It depicts the basic form of an event log in which each row represents an *event* and each *case ID* indicates an instance. An event log may include more data attributes, but, for simplicity, we abstract from these.

Definition 1 (Event Log). *Let $\mathcal{C}$, $\mathcal{A}$, $\mathcal{R}$ and $\mathcal{T}$ denote the universe of case identifiers, activities, resources, and the time universe, respectively. The universe of events is defined as $\xi = \mathcal{C} \times \mathcal{A} \times \mathcal{R} \times \mathcal{T} \times \mathcal{T}$. An event $e = (c, a, r, t_s, t_c) \in \xi$ refers to a case c, an activity a, a resource r, a start time t_s, and a complete time t_c. We define corresponding projection functions $\pi_{\mathcal{C}}{:}\xi \rightarrow \mathcal{C}$, $\pi_{\mathcal{A}}{:}\xi \rightarrow \mathcal{A}$, $\pi_{\mathcal{R}}{:}\xi \rightarrow \mathcal{R}$ and $\pi_{\mathcal{T}}{:}\xi \rightarrow \mathcal{T} \times \mathcal{T}$. Given $e = (c, a, r, t_s, t_e) \in \xi$, we have $\pi_{\mathcal{C}}(e){=}c$, $\pi_{\mathcal{A}}(e){=}a$, $\pi_{\mathcal{R}}(e) = r$, and $\pi_{\mathcal{T}}(e){=}(t_s, t_c)$. An event log $L \subseteq \xi$ is a set of events.*

Consider the first event depicted in Table 1. In the context of Definition 1, the first row (which we denote as e_1), describes: $\pi_{\mathcal{C}}(e_1){=}1$, $\pi_{\mathcal{A}}(e_1){=}Register$, $\pi_{\mathcal{R}}(e_1){=}Rose$ and $\pi_{\mathcal{T}}(e_1){=}(10/1/2018\ 7:38:45,\ 10/1/2018\ 7:42:30)$. Using such event data, process mining techniques can be used to discover process models, check conformance, uncover bottlenecks, predict process outcomes, and steer process improvement initiatives.

System dynamics provides a collection of techniques and tools to model and analyze capturing changes in complex systems over time [20]. Two main diagrams used within system dynamics are the *causal-loop diagram* and the *stock-flow diagram* which represent the conceptual relations between variables in the system and underlying equations respectively [13].

Definition 2 (The Causal-loop Diagrams). *A causal-loop diagram $CLD = (\mathcal{V}, \mathcal{L})$ is a set of nodes $\mathcal{V}$ and a set of directed links $\mathcal{L} \subseteq \mathcal{V} \times \mathcal{V}$. Directed link $l = (v_1, v_2) \in \mathcal{L}$ connects nodes v_1 and v_2 using a directed arc.*



$$Stock1 = Stock1_0 + \int_{t_0}^{t_n} (+Flow1 - Flow2)\mathrm{d}t \quad (1)$$

Fig. 2: A simple example stock-flow diagram and the underlying relation of $Stock1$ and its in/outflows ($Flow1$ and $Flow2$).

The designed causal-loop diagram is a platform for designing a stock-flow diagram. Since all the elements and their relations are already modeled in the causal-loop diagram, a mapping between nodes in the causal-loop diagram and the elements in the stock-flow diagram should be made. A stock-flow diagram also is a diagram that indicates the same relationship in a causal-loop diagram for a system using three different basic elements, i.e., *stocks*, *flows* and *variables* [4]. Entities accumulated over time represented by numbers are usually mapped to a stock. Rate-based entities such as income per month can be considered as flows that can add to or remove from stocks.

Definition 3 (Stock-flow Diagram). *A stock-flow diagram is a tuple* (S, F, A, M) *with three disjoint set of elements, i.e., stocks* S*, flows* F*, and variables* A*, and* $M \subseteq (S \cup F \cup A) \times (F \cup A)$ *is a relation showing the flow of information between the elements.* $\mathcal{V} = S \cup F \cup A$ *and three subsets are pairwise disjoint.* $\circlearrowleft \in S$ *is the boundary of the system.*

Each of the subsets introduced in Definition 3 is visualized with a specific shape the corresponding diagram, see Fig. 2. $Stock1 \in S$, $Flow1$ and $Flow2 \in F$ and $Variable1 \in A$ are the elements and the arcs between each two elements are derived from M. Also, there are two information flows from system boundaries to $Stock1$ and vice versa. Stock-flow diagrams are used for simulation using the specified underlying equations. The equation depicted on the right-hand side of Fig. 2 describes the underlying relation for the diagram. Consider t as time, $Stock1$ is equal to the amount in $Stock1$ at time t_0 plus the integral over the difference of the $Flow1$ and $Flow2$ over the time interval $[t_0, t_n]$. In each step, values of stock-flow elements get updated based on the previous values of the other elements that influence them.

3 Related Work

In this work, we propose a framework using process mining to provide insights that support the modeling techniques in system dynamics. The resulting models are used for the purpose of scenario-based prediction. We refer to [2] and [20] for an overview of process mining and system dynamics, respectively. Different approaches and techniques have addressed forward-looking in process mining. Among these approaches, we divide the ones w.r.t. performance into two main

categories including simulation techniques and prediction techniques. Both categories of approaches aim to predict the future state of a process. In the first category, work such as [17] introduces discrete event simulation on the basis of discovered process models. Moreover, workflow management and simulation are combined in [18]. The authors considered both workflow design and event data to provide a model for the current state of the workflow. As the author in [1] mentioned, the factor of human behavior is missing in the proposed techniques.

In the second category, the approaches focus on predicting the performance aspects of the processes. The authors in [5] aim to predict the remaining process time or outcome of specific cases. In [23] a survey on the approaches which use prediction techniques in process mining is provided. The proposed approaches are mainly focused on the short-term prediction, e.g., predicting the time in which a specific process instance, i.e., a customer process will be finished or what will be the next activity [21]. The importance of context and interaction with the factors outside the processes for the prediction and simulation has been shown extensively [3, 8]. Yet, existing approaches tend to abstract from these.

A combination of system dynamics with the process management field is proposed in [6]. In the context of processes also in [15], a business process of SAP is introduced in the form of system dynamics models that covers the factor of employees' productivity. However, in system dynamics models such as most of the simulation models, it is difficult to assess the reliability of the prediction results [1]. Moreover to the provided techniques in process mining and system dynamics, a combination of both fields in order to perform scenario-based analyses in the business processes has been recently proposed [10]. In this approach, the freedom in choosing the level of detail in modeling using system dynamics modeling, and the possibility of extending the factors outside of the processes are provided. Using event logs in process mining the validity of the designed models based on the result of the simulation is assessed. In our approach, we extend the main framework presented in [10] and propose a standalone interactive framework that supports the designing step using the event logs and structure of the system dynamics diagrams.

4 Approach

In this section, we explain the main approach focusing on the automatic generation of system dynamics models. As Fig. 3 shows, we transform an event log into a sequence of measurable performance parameters of a process, i.e., SD-log. In the parameter extraction module, we get the performance questions in the context of scenario-based analysis, e.g., how does the increase in the number of arrival affect the average waiting time in the process? Then, we extract the possible measurable parameters related to the questions over time. The calculated values of these parameters on the specified window of time form the SD-log.

Our approach continues using the generated SD-log to detect any possible relationship between the parameters in which each relation has a type and a direction. The type of a relation can be linear or nonlinear and the direction of

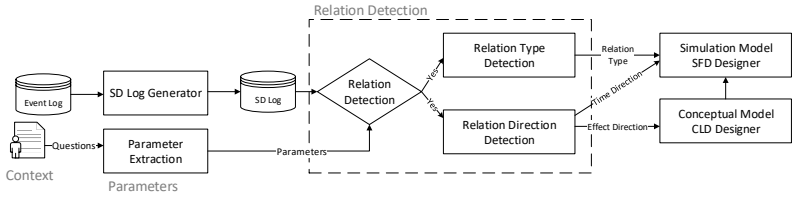


Fig. 3: The main approach including the SD-log generation, relation detection and the discovery of the type and direction of the relations. Our approach continues with the automatic generation of causal-loop diagrams (CLD) and Stock-flow diagrams (SFD). The type of relationship is used to form the underlying equations in SFD and the effect and time directions are automatically used to design the CLD as a backbone of SFD.

the relationship exists in two dimensions, time and effect. For instance, there is a relation between the arrival rate per day and the average waiting time per day. The type of this relation can be linear/nonlinear and negative/positive. The direction of the effect is from arrival rate to average waiting time that means arrival rate influences the average waiting time. At the same time, the effect of increases in the number of arrivals may only be visible with some delay, e.g., after two hours in the average waiting time, which shows the direction in time. These discovered relations and their types and directions are used to build the system dynamics simulation models.

4.1 SD-Log Generator

An event log is the starting point of any analyses in process mining, therefore, the possible parameters are highly dependent on the available data in the event log. In our approach, we consider the basic type of event logs in Definition 1. Hence, time-related performance parameters w.r.t events, cases, resources, and activities, e.g., service time of a case/event can be generated. Instead of extracting and computing the parameters at the instance level, in this work, we define aggregated parameters over a specific period of time such as δ. Reconsider Table 1. We extract the average duration of performing activity "Register" in each hour as δ in the log instead of extracting the values for each case separately. Having all the values in the possible steps considering the time window forms the SD-log, which we define as Definition 4. Performance parameters can be defined and extracted from an event log w.r.t. the scenario-based questions which we consider as a set $\mathcal{V}$.

Definition 4 (SD-log). *Let $L \subseteq \xi$ be an event log, $\mathcal{V}$ be a set of process parameters, and δ be the selected time window. Assume e_1 is the first event in the event log starting at time t_S and assume e_n is the last event in the event log completing at time t_C. Given the time window δ, there are $k = \lceil (t_C - t_S)/\delta \rceil$ subsequent*

time windows to go from t_S to t_C. An SD-log is a function $SD: \mathcal{V} \to \mathbb{R}^k$, where $\mathbb{R}^k$ is a sequence of real numbers of length k. Furthermore, for any $v \in \mathcal{V}$ and $0 \leq i < k$, we use $\pi_i(SD(v))$ to denote the $(i+1)^{th}$ value for parameter v, i.e., if $SD(v) = \langle x_0, x_1, \ldots x_{k-1} \rangle$ is the sequence for parameter v, then $\pi_i(SD(v)) = x_i$.

Assume event log $L \subseteq \xi$ with a total duration of 10 hours, $\delta = 1$ *hour* implies $k = 10$. An example SD-log including one variable *Arrival rate* is: $SD(Arrival\ rate) = \langle 11, 13, 10, 10, 12, 9, 10, 13, 8, 11 \rangle$, i.e., $\pi_2(SD(Arrival\ rate)) = 10$, representing that in the third hour, 10 cases arrived in the process.

It is important to note that each parameter is mapped onto a sequence of real numbers. Each real number is computed over the event log and focuses on the combination of a parameter and a time window, e.g., number of customers handled, number of customers queuing, average waiting time, percentage rejected, etc. Selection of the parameters is highly dependent on the "what-if" questions. Using these types of questions, the parameters are extracted and are being used in the next steps. We refer to [10] for detail of calculation method and dealing with overlapping features in multiple time windows.

4.2 Relation Detection

System dynamics models are based on the effects of the system's parameters on each other. In designing the system dynamics models such as causal-loop diagrams and stock-flow diagrams, the relations between the elements are crucial. Knowing these relations and their directions makes it possible to create causal-loop diagrams and eventually stock-flow diagrams that can be used to simulate different scenarios. In the relation detection part, we discover all the possible relations and use them to automatically design the simulation models.

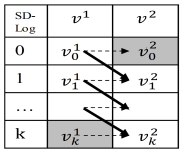


Fig. 4: A sample SD-log with k values for two parameters v^1 and v^2. The black arrows show that by investigating relations for the next time window (one step shift), one value for each parameter gets ignored, i.e., v_k^1 and v_0^2.

Considering the values of the parameters from the SD-log in the specified time window, we calculate both linear and nonlinear correlation using *Pearson correlation* and *Distance correlation* techniques [19]. In addition to the relation between two parameters regarding one influencing another one, this influence can happen in different time windows. Therefore, a relation has two aspects, i.e., the direction of the effect that shows which parameter causes changes and also the time window in which these changes would influence the second parameter. One parameter may affect another parameter at a later time, hence it is not sufficient to calculate correlations between values in the same time window in this situation. In our previous example, if we look for the relations only at the same time window the effect of changes in the arrival rate on the average waiting time which appears after two hours would not be captured for the time window of one hour.

Consider Fig. 4 showing a possible time-shifted relation between two parameters. Due to the shift, we lose some values at the beginning and end of the SD-log. Therefore, to compare the values of parameters in different steps of the time window, an indicator is needed in order to preserve a sufficient number of values. Assume $s \in \mathbb{N}$ as the maximum possible shift in the time windows to look for the cause and effect between parameters where $s \leq k(1-\theta_{sd})$ and θ_{sd} is the minimum percentage of values of the parameters that we are willing to use and $k \in \mathbb{N}$ is the number of values presented in the SD-log. Accordingly, prior to defining the relation detection algorithm, we need to define a shift function that provides the values of parameters with the given shift for detecting their underlying relations. We define the shift function in Definition 5 and use this function as an input of the relation detection algorithm in Algorithm 1.

Definition 5 (Shift Function). *Let $SD: \mathcal{V} \rightarrow \mathbb{R}^k$ be an SD-log with parameters $\mathcal{V}$ and s the maximum possible shift. For any two parameters v^1,$v^2 \in \mathcal{V}$, and a shift i with $0 \leq i \leq s$: $ShiftFun_i : \mathbb{R}^k \times \mathbb{R}^k \rightarrow \mathbb{R}^{k-i} \times \mathbb{R}^{k-i}$ relates values for v^1 with later values for v^2, i.e., $ShiftFun_i(SD(v^1), SD(v^2)) = (\langle \pi_0(v^1), ..., \pi_{k-(i+1)}(v^1) \rangle, \langle \pi_i(v^2), ..., \pi_{k-1}(v^2) \rangle)$ (note that $k = |SD(v^1)| = |SD(v^2)|$.*

Consider v^1 and v^2 as the arrival rate per day and the number of waiting cases per day and $SD(v^1) = \langle 11, 13, 10, 10, 12, 9, 10, 13, 8, 11 \rangle$, $SD(v^2) = \langle 2, 3, 0, 1, 4, 0, 1, 3, 0, 2 \rangle$ are the values of two parameters, applying the $ShiftFun_2$ will result in $SD(v^1) = \langle 11, 13, 10, 10, 12, 9, 10, 13 \rangle$, $SD(v^2) = \langle 0, 1, 4, 0, 1, 3, 0, 2 \rangle$.

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. The comparison with the threshold shows whether the relationship exists or not. Each pair of parameters as an output of the algorithm will define the relations between parameters and be the root of the automatic causal-loop diagram designer in our approach.

Conceptual Model: Causal-loop Diagram Designer Using Algorithm 1, the relations are extracted and automatically transformed into a causal-loop diagram. In the transformation step, the domain knowledge of the user is also considered in indicating and selecting the desired relations in the output causal-loop diagram. Consider, for example, the effect of an increase in the arrival rate per time window on the average waiting time of cases. In this example, the output of Algorithm 1 is $(arrival\ rate, average\ waiting\ time)$, showing that changes in the values of arrival rate over time would cause changes in the average waiting time. The parameters in $\mathcal{V}$ from the SD-log are mapped to the nodes in the CLD diagram and the relations are represented as the links L. The extracted relations such as the example, form the causal-loop diagram automatically, e.g., in the causal-loop diagram, there is a link from *arrival rate* to the node labeled as *average waiting time*.

Algorithm 1: Relation Detection Algorithm

Input: SD_Log
Input: Maximum possible shift s and threshold of accepting a relation θ_rel
Output: All relations between pair of parameters $\in \mathcal{V}$

foreach v^m *and* $v^n \in \mathcal{V}$ **do**
  foreach $0 \leq i \leq s$ **do**
    Generate $score = correlation(ShiftFun_i(v^m, v^n))$;
    Add $score$ to the set Set_scores;
  end
  return $Max(Set\_scores)$ as max_score;
  if $max\_score \geq \theta\_rel$ **then**
    return (v^m, v^n) as a relation;
  else
    return null;
  end
end

Simulation Model: Stock-flow Diagram Designer Having a causal-loop diagram, the platform for designing the stock-flow diagram is provided. The parameters as mentioned in Section 4.1 are divided into three types namely, rate, number, and duration based which automatically are mapped into the stock-flow diagram as flows, stocks or variables, respectively. For a generated CLD, all nodes $v \in \mathcal{V}$ are mapped to a $s \in S, f \in F$ or $a \in A$ and for each link $l \in L$ such as (v^1, v^2) the corresponding notation in stock-flow diagram is replaced. Therefore, information flow M in Definition 3 is corresponding to the links in CLD with the replaced notations. The constraints in relations in stock-flow diagrams definition in Definition 3, automatically preserved in the mapping. A stock and a flow can influence a variable but a variable can only influence a flow or a variable, also a flow is able to influence a stock. Also, the option of including the user's domain knowledge regarding the simulation scenarios is provided.

Domain knowledge can be inserted interactively to the mapping step, e.g., the number of cases waiting in the system as a parameter can be treated as both stock or a variable which based on the scenario are exchangeable. For all the generated models, the system dynamics files (i.e., *mdl* files) are generated. These *mdl* files can be used for the scenario-based analyses presented in [10], but also analyzed using system dynamics simulation software such as *Vensim*.[1] The ability to generate system dynamics models from event logs provides an integrated approach that is both forward-looking and backward-looking.

5 Evaluation

We use synthetic event logs including different types of relationships between process' performance parameters to evaluate our approach. Moreover, we design

[1] https://vensim.com

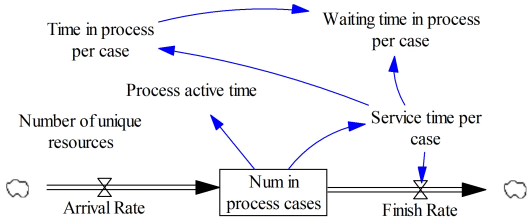


Fig. 5: The automatically generated output of the approach using the event log of the call center (the result is available as an *mdl* file that can be used in various system dynamics tools, e.g., Vensim).

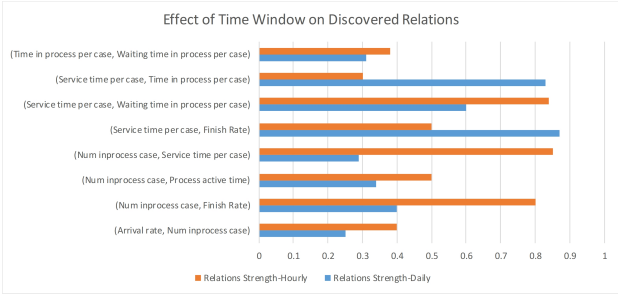


Fig. 6: A comparison between the strength of discovered relations for different SD-logs generated using *Hourly* and *Daily* time windows. As expected, the relations in the *Hourly* time window are stronger. However, the strength of the relations in the *Daily* time window are also above the threshold.

scenarios in which the input of the approach is the output of the predefined simulation model using which we can measure the similarity between the generated model and the original one.

Evaluating the Feasibility of the Approach First, we evaluate our model with an event log which is intentionally designed with multiple linear and nonlinear relations. The hidden cause and effects inside the parameters of the process are specified in the process model. The process designed using CPN tools [14] simulates the process inside the call center of a car rental agency in which two types of requests are handled, requests for cars and requests for cars with a driver. The requests are randomly generated using the ratio of 60 percent for cars and 40 percent for cars with driver. For this reason, three resources are assigned for car requests and 2 resources are assigned for a car with a driver requests. The working hours of the call center are between 8:00 in the morning until 17:00 in the afternoon for 7 days per week with a higher number of requests around 10:00 and 15:00. Also, if the number of customers in line for getting the service is above 30 customers, the call will be rejected automatically.

We designed the model in a way that the operators perform the process of the calls faster if the number of calls in the line is higher. This effect of length of queues of the inline calls on the time of the processing calls is modeled as nonlinear relation, using an exponential function. Furthermore, an increase in the arrival rate influences the number of finish rates and the number of calls waiting inline to get the service. The generated event log using the presented model including 2000 cases used as an input of the approach and the detected relation using a *Daily* time window is shown as Fig. 5. For calculating the maximum shift, we set the minimum of data to 90 percent.

All the underlying effects at the instance level are captured. Then the generated model can be used as a basis of the simulation and scenario-based analysis w.r.t the changes in the process parameters. As the designed model by the approach indicates, only for the parameter *Number of unique resources* per day which we expect to have an effect on the number of handled requests, our approach could not find any strong relations. The reason is that we consider a fixed number of the resources in our CPN model.

We extended our experiments by choosing different time windows to see if the approach is able to capture the relations in different time windows. Fig. 6 illustrates the comparison of the discovered relations in the time window of *Daily* and *Hourly*. The strength of the relations as a result of Algorithm 1 are scaled between 0 and 1 which 1 shows the strongest relationship. The relations in *Hourly* time windows are stronger than the *Daily* manner, however, in both time windows, the expected relations are discovered. For instance, consider the relation between the number of cases in the process and the service time of cases, the relation can be seen strongly in *Hourly* time window, 0.84 rather than in *Daily* time window 0.29. As we expected, the approach is able to discover the hidden relations at an aggregated level, however, choosing the appropriate time window regarding the context of the process and modeling is important.

Evaluating the Accuracy of the Approach We also designed a system dynamics model manually and generated a set of values over a defined *Daily* time window and used the generated data to feed to our approach to compare the discovered system dynamics model with the original one. Fig. 7 shows the designed evaluation scenario.

Assume R is the universe of all relations between parameters and R_o, $R_d \subseteq R$ represent the set of relations in the original diagram which is manually designed and the set of relations in the designed diagram using the proposed approach, respectively. We define two similarity measures regarding the accuracy and precision of the designed model [12]: Equation 2 and Equation 3. *Accuracy* considers the similarity between the existing relations in the original and in the discovered model and *Precision* considers the relations which in the original model do not exist but are discovered wrongly in the designed model.

$$Accuracy = \frac{|R_o| \cap |R_d|}{|R_o|} \quad (2) \qquad Precision = \frac{|R_o| \cap |R_d|}{(|R_o| \cap |R_d|) + (|R_d| \cap |R_o|/R)} \quad (3)$$

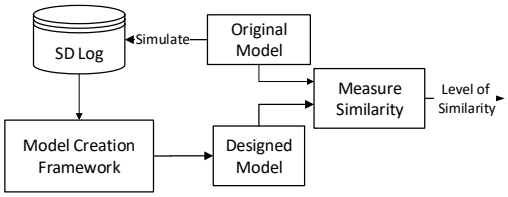
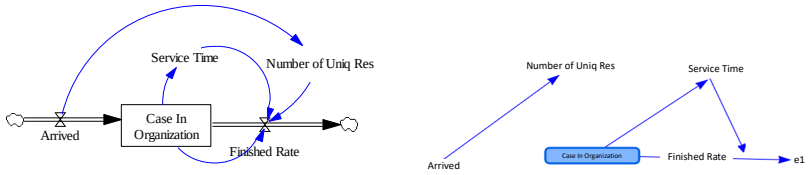


Fig. 7: The designed evaluation scenario in order to measure the similarity of the designed model using the proposed approach and the original model.



(a) (b)

Fig. 8: Evaluation results in the similarity measurement scenario. The model presented in (a) is designed and the simulation results are used to feed the proposed approach. Model (b) is the designed model by the approach.

Fig. 8a shows the original model which is simulated for 30 weeks and generates an SD-log out of it. This model includes the relationship between the number of resources and the arrival rate using the equations based on the root equation. If the arrival rate increases, the number of resources will increase up to a specific point. Also, the finish rate is based on the number of cases in the process and the average service time, whereas service time itself gets influence from the number of cases in the process, i.e., more cases waiting in the process, resources work faster hence the service time will decrease. We give the generated SD-log as an input to our framework and Fig. 8b shows the discovered model. Relations in the original model and discovered ones in the designed model result in an *Accuracy* of 0.72 and a *Precision* of 1. It should be considered the calculated measures are for the designed model without the interference of any domain knowledge from users.

The results of both simulation scenarios indicate the effectiveness of our approach. More importantly, all the relations were at the instance level, i.e., for each specific case and our approach is able to catch the aggregated effects without considering performance metrics and details of the process's steps which happened for each case.

Evaluating the Approach Using Real Event Log In order to test our approach in practice we applied our framework on the real event log *BPIChal-*

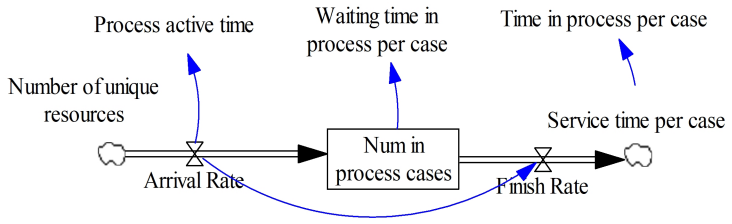


Fig. 9: Generated model for the real event log using the proposed approach.

lenge2017 [24]. The event log includes different executions of processes for taking a loan by customers. The number of cases is 31500 and the number of recorded events is 382454. The discovered relations between process parameters and the automatically generated model considering the *daily* behavior of the process is shown in 9. As expected, there is a strong relationship between the number of served people in the process per day and the number of people arrived per day. However, since not all the relations inside the event log are known, therefore the evaluation can only be performed based on the hypothesis and background knowledge of the processes. As indicated in Fig. 9, the number of people in the process of taking service is also directly influencing the process active time and the average waiting time per case. The reason for not indicating any strong relationship between the service time and the finish rate can be that the requests need a specific amount of time and not related to any other factors. The experiment using the real event log shows that the proposed approach is able to capture the most expected relations inside a real process.

6 Conclusion

In this paper, we proposed a novel approach to support designing system dynamics models for simulation in the context of operational processes. Using our approach, the underlying effects and relations at the instance level can be detected and modeled in an aggregated manner. For instance, as we showed in the evaluation, the effects of the amount of workload on the speed of resources are of high importance in modeling the number of people waiting to be served per day. In the second scenario, we focused on assessing the accuracy and precision of our approach in designing a simulation model. As the evaluations show, our approach is capable of discovering hidden relations and automatically generates the valid simulation models in which applying the domain knowledge is also possible. By extending the framework, we are looking to find the underlying equations between the parameters. The discovered equations help to obtain accurate simulation results in an automated fashion without user involvement. Moreover, we aim to apply the framework in case studies where we not only have the event data, but can also influence the process.

Acknowledgments

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

References

1. van der Aalst, W.M.P.: Business process simulation survival guide. In: Handbook on Business Process Management 1, Introduction, Methods, and Information Systems, 2nd Ed., pp. 337–370. Springer (2015)
2. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016). https://doi.org/10.1007/978-3-662-49851-4, https://doi.org/10.1007/978-3-662-49851-4
3. van der Aalst, W.M.P., dustdar, S.: Process mining put into context. IEEE Internet Computing **16**, 82–86 (2012)
4. Binder, T., Vox, A., Belyazid, S., Haraldsson, H., Svensson, M.: Developing system dynamics models from causal loop diagrams. In: Proceedings of the 22nd International Conference of the System Dynamics Society. Oxford, Great Britain, July 25-29, 2004 (2004)
5. van Dongen, B.F., Crooy, R.A., van der Aalst, W.M.P.: Cycle time prediction: When will this case finally be finished? In: On the Move to Meaningful Internet Systems: OTM 2008, OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Monterrey, Mexico, November 9-14, 2008, Proceedings, Part I. pp. 319–336 (2008). https://doi.org/10.1007/978-3-540-88871-0_22, https://doi.org/10.1007/978-3-540-88871-0_22
6. Duggan, J.: A comparison of Petri net and system dynamics approaches for modelling dynamic feedback systems. In: 24th International Conference of the Systems Dynamics Society (2006)
7. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Process and deviation exploration with inductive visual miner. In: Proceedings of the BPM Demo Sessions 2014 Co-located with the 12th International Conference on Business Process Management (BPM 2014), Eindhoven, The Netherlands, September 10, 2014. p. 46 (2014), http://ceur-ws.org/Vol-1295/paper19.pdf
8. de Leoni, M., van der Aalst, W.M.P., Dees, M.: A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. Inf. Syst. **56**, 235–257 (2016). https://doi.org/10.1016/j.is.2015.07.003, https://doi.org/10.1016/j.is.2015.07.003
9. Mannhardt, F., de Leoni, M., Reijers, H.A.: The multi-perspective process explorer. In: Proceedings of the BPM Demo Session 2015 Co-located with the 13th International Conference on Business Process Management (BPM 2015), Innsbruck, Austria, September 2, 2015. pp. 130–134 (2015), http://ceur-ws.org/Vol-1418/paper27.pdf
10. Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M.P.: Scenario-based prediction of business processes using system dynamics. In: On the Move to Meaningful Internet Systems: OTM 2019 Conferences - Confederated International Conferences: CoopIS, ODBASE, C&TC 2019, Rhodes, Greece, October 21-25, 2019, Proceedings. pp. 422–439 (2019). https://doi.org/10.1007/978-3-030-33246-4_27, https://doi.org/10.1007/978-3-030-33246-4_27

11. Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M.P.: Supporting decisions in production line processes by combining process mining and system dynamics. In: Intelligent Human Systems Integration 2020 - Proceedings of the 3rd International Conference on Intelligent Human Systems Integration (IHSI 2020): Integrating People and Intelligent Systems, February 19-21, 2020, Modena, Italy. pp. 461–467 (2020). https://doi.org/10.1007/978-3-030-39512-4_72, https://doi.org/10.1007/978-3-030-39512-4_72
12. Powers, D.M.: Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. Journal of Machine Learning Technologies **2**, 37–63 (2011)
13. Pruyt, E.: Small system dynamics models for big issues: Triple jump towards real-world complexity. TU Delft Library (2013)
14. Ratzer, A.V., Wells, L., Lassen, H.M., Laursen, M., Qvortrup, J.F., Stissing, M.S., Westergaard, M., Christensen, S., Jensen, K.: CPN tools for editing, simulating, and analysing coloured petri nets. In: Applications and Theory of Petri Nets 2003, 24th International Conference, ICATPN 2003, Eindhoven, The Netherlands, June 23-27, 2003, Proceedings. pp. 450–462 (2003). https://doi.org/10.1007/3-540-44919-1_28, https://doi.org/10.1007/3-540-44919-1_28
15. Rosenberg, Z., Riasanow, T., Krcmar, H.: A system dynamics model for business process change projects. In: International Conference of the System Dynamics Society. pp. 1–27 (2015)
16. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering colored petri nets from event logs. STTT **10**(1), 57–74 (2008). https://doi.org/10.1007/s10009-007-0051-0, https://doi.org/10.1007/s10009-007-0051-0
17. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering simulation models. Inf. Syst. **34**(3), 305–327 (2009). https://doi.org/10.1016/j.is.2008.09.002, https://doi.org/10.1016/j.is.2008.09.002
18. Rozinat, A., Wynn, M.T., van der Aalst, W.M.P., ter Hofstede, A.H.M., Fidge, C.J.: Workflow simulation for operational decision support. Data Knowl. Eng. **68**(9), 834–850 (2009). https://doi.org/10.1016/j.datak.2009.02.014, https://doi.org/10.1016/j.datak.2009.02.014
19. Schober, P., Boer, C., Schwarte, L.A.: Correlation coefficients: appropriate use and interpretation. Anesthesia & Analgesia **126**(5), 1763–1768 (2018)
20. Sterman, J.D.: Business dynamics: systems thinking and modeling for a complex world. McGraw-Hill (2000)
21. Tax, N., Teinemaa, I., van Zelst, S.J.: An interdisciplinary comparison of sequence modeling methods for next-element prediction. CoRR **abs/1811.00062** (2018), http://arxiv.org/abs/1811.00062
22. Tax, N., Verenich, I., Rosa, M.L., Dumas, M.: Predictive business process monitoring with LSTM neural networks. CoRR **abs/1612.02130** (2016), http://arxiv.org/abs/1612.02130
23. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. TKDD **13**(2), 17:1–17:57 (2019). https://doi.org/10.1145/3301300, https://doi.org/10.1145/3301300
24. Van Dongen, B.F. (Boudewijn): Bpi challenge 2017 (2017). https://doi.org/10.4121/UUID:5F3067DF-F10B-45DA-B98B-86AE4C7A310B