Scenario-Based Prediction of Business Processes Using System Dynamics

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. Many organizations employ an information system that supports the execution of their business processes. During the execution of these processes, event data are stored in the databases that support the information system. The field of process mining aims to transform such data into actionable insights, which allow business owners to improve their daily operations. For example, a process model describing the actual execution of the process can be easily extracted from the captured event data. Most process mining techniques are "backward-looking" providing compliance and performance information. Few process mining techniques are "forward-looking". Therefore, in this paper, we propose a novel scenario-based predictive approach that allows us to assess and predict future behavior in business processes. In particular, we propose to use system dynamics to allow for "what-if" questions. We create a system dynamics model using variables trained on the basis of the past behavior of the process, as captured in the event log. This model is used to explore the effect of possibly applied changes in the process as well as roles of external factors, e.g., human behavior. Using real event data, we demonstrate the feasibility of our approach to predict possible consequences of future decisions and policies.

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

1 Introduction

Modern information systems allow us to track the execution of the business processes of an organization. *Process mining* techniques [2] have proven to be a valuable addition to the toolbox of modern-day process analysts. Process mining provides several data-driven algorithms and tools that allow us to gain a better understanding of, and insights in, the execution of the business processes at play. For example, in *process discovery* [3], techniques allow us to discover a process model that accurately describes the process as captured in the data. Similarly, in *conformance checking* [4], techniques assess to what degree a given process model is in line with the captured data. Furthermore, a multitude of techniques

exists, i.e., *process enhancement techniques* [8,9], that aim to increase the overall *view of the process*, e.g., projecting performance information in a process model.

The intrinsic value and premise of process mining are clear and widely accepted: *data does not lie*. At the same time, data-driven support for possible next steps to be taken by the organization, in order to improve the process performance, e.g., increasing workforce, is often missing. Undisputed, more advanced algorithms to predict the future behavior of a process, specifically with the aim of improving process performance, are of interest to many organizations. However, in process mining, existing work towards the prediction of future behavior w.r.t. performance of processes, typically depends on extensive knowledge about the process [13, 14]. For example, the approach presented in [14] uses discovered process models as a basis and, therefore, implicitly, depends on the quality of the discovered process model. Other techniques do not require in-depth knowledge of the process [18, 19], however, such techniques focus on short-term prediction. "What-if" analysis is different from existing techniques that try to predict at the case level. None of the existing techniques predict the effects of changes in the execution of the process on a large scale, without having explicit in-depth knowledge of the process. However, a decision maker of an organization often has a limited view and understanding of the global process, yet is interested in the prediction of global key process performance indicators by *explicitly taking the business context into account*. For example, to investigate whether replacing a resource in an assembly line reduces the overall service time.

In this paper, we present a novel approach that allows us to predict future behavior in business processes, subject to envisioned future scenarios. In particular, we exploit *system dynamics* [16], i.e., a modeling formalism designed to inspect the effects of changes within an organization. System dynamics is a widely used approach in the context of scenario-based analysis supported by software tools, e.g., `vensim` (`http://vensim.com`). An overview of the proposed approach, including its relation to conventional system dynamics and process mining, is depicted in Fig. 1.

Our approach starts with a data processing step in which we transform an event log into a collection of measurable aspects with an associated temporal ordering. Subsequently, we map these measurable aspects onto system dynamics model elements, which allows us to predict future behavior of virtually any measurable aspect of a process. To evaluate the proposed approach, we conducted a collection of experiments using both synthetic and real data sets and we mostly focus on the real data sets. Our experiments show that by using an aggregated view of the process performance by means of system dynamics, it is possible to predict the effects of changes on future process performance.

The remainder of this paper is organized as follows. In Section 2, we explain the motivation. In Section 3, we present related work. In Section 4, we introduce background concepts and basic notation used throughout the paper. In Section 5, we present our main approach. We evaluate the proposed approach in Section 6. Section 7 concludes our work and discusses interesting directions for future work.

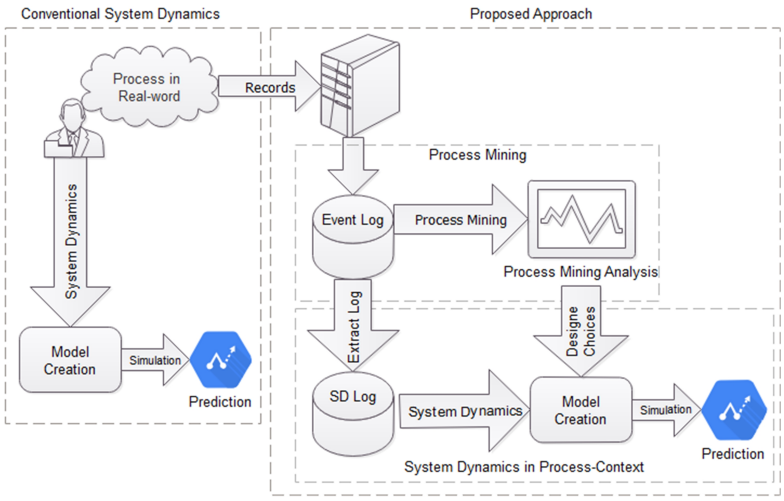


Fig. 1: Schematic overview of the proposed approach and its relation to conventional System Dynamics and Process Mining. In the proposed framework, we populate a system dynamics model with process performance statistics distilled from the previous execution of the process as captured in an event log.

2 Motivation

Business owners and decision makers are highly interested to improve the performance of their processes. However, considering the cost of changes in the process, it is required to have insights about the effects of the new changes in the processes before applying them in reality. Different techniques propose the simulation and prediction of the processes, e.g., discrete event simulation. Discrete event simulation techniques need extensive knowledge of the process, and are not able to take the context of prediction into account. Context is often neglected during future analysis in process mining [5]. Also, it is not possible to incorporate the effects of external factors in the models, e.g., human behaviors or environmental variables such as economic. Moreover, the level of detail in these types of approach does not allow for high-level modeling and long-term predictions.

As Fig. 2 represents, most of the prediction techniques in process mining focus in the center of the circle, i.e., instance level. As opposed to the existing simulation techniques, system dynamics allows us to assess the impact of changes in the process from a *global perspective* as well as the effects of *external factors*. Using different levels of granularity in the modeling, we can address major drawbacks of discrete event simulation techniques.

The motivation of our new approach is to move from the center of Fig. 2, i.e., instance level to the outside layers and providing "what-if" analysis at a higher level of abstraction which also takes the context into account. There is

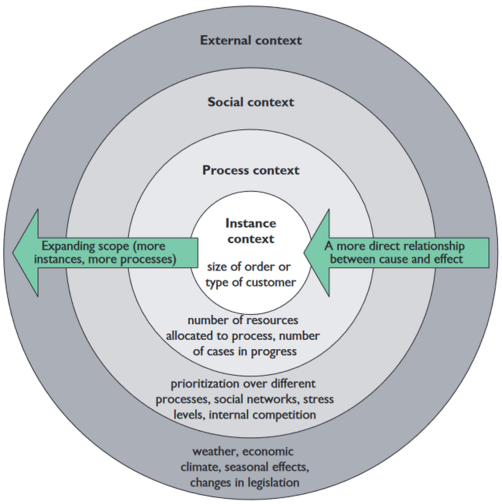


Fig. 2: Various contexts and levels of abstraction in process mining [5].

a trade-off between the amount of knowledge inside the process and the chosen abstraction level. Also, freedom of considering the external factors influences the accuracy of the results. The detailed level of designed system dynamics models, i.e., benefiting from knowing the detail of a process will lead the simulation to be more similar to the discrete event simulation. However, we mainly focus on the roles of external variables and providing the missing bridge between the "as-is" state of the processes to the future state, i.e, "to-be" based on "what-if" analysis.

3 Related Work

To the best of our knowledge, this is the first work that proposes to combine the fields of process mining and system dynamics techniques for the purpose of scenario-based prediction. We refer to [16] and [2] for an overview of system dynamics and process mining respectively.

Process mining research is mostly "backward-looking". Compared to the "backward-looking" approaches, a few "forward-looking" approaches exist. In [14] discrete event simulation on the basis of discovered process models is introduced. The approach in [15] is based on a combination of workflow management and simulation. Work using discrete event simulation requires many details. As a result, modeling and tuning models can be very time consuming. It is also impossible/difficult to simulate human behavior at a very detailed level as mentioned in [1]. There are approaches that focus on the prediction and recommendation, e.g., predicting the remaining process time or outcome of specific cases in [6]. The right abstraction level is very important for creating a model. These work are based on a detailed and case-based view and not at an aggre-

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
...	...	...	...	...

gated level. A considerable number of methods have been put forward to address the problem of predictive process monitoring at the instance level [20]. Also, it is difficult to assess the reliability of the prediction results [1]. Moreover, some work generating models using statistical analysis. Considering time intervals in performance analysis is proposed in [17]. The proposed framework allows for a systematic approach to performance-related analysis beyond the capabilities of existing log-based analysis techniques.

In the field of system dynamics, different work focus on simulation and prediction. In particular, use of system dynamics in the context of business process management, e.g., using both Petri net models and system dynamics to develop a model for the same situation [7, 12] can be mentioned. In [12], a standard SAP reference business process is used. The authors use system dynamics models to determine how the business process can be changed to achieve improvement in the employee productivity. Furthermore, [7] demonstrates how common problems, e.g., finding the average waiting time, are addressed with different models using a comparison of Petri net and system dynamics. In this work, the elements in Petri net (places and transitions) are considered as elements in the system dynamics models (stocks and flows).

The approach presented in this paper differs from existing approaches in various ways: (1) there is no need to modle the process at a fine-grained level, i.e., our approach is based on an aggregated level (using system dynamics), (2) designing the system dynamics model at an aggregated level is relatively simple in comparison with methods such as Colored Petri Net tools (CPN) (which is complicated for the larger/complex processes and need complete knowledge of the processes), (3) the approach uses valid models which behave the same as reality, and, (4) the approach provides a platform, allowing us to involve the external factors/human behavior and their effects in the simulation results.

4 Background

In this section, we formalize concepts related to process mining and system dynamics.

Process mining Historic data, captured during the execution of a company's processes, play a central role in any process mining analysis. The execution of an *activity*, in the context of some process instance, identified by a unique *case*

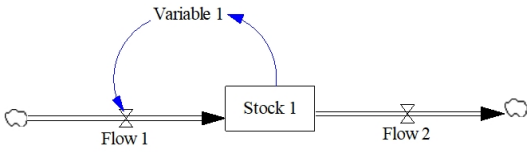
ID, is referred to as an *event*. Consider Table 1, in which we present a simplified example of an event log. Observe that, in the event log, there are events depicted of three different process instances, identified by Case IDs 1, 2 and 3. The first event in the event log describes that *Rose* started performing a *Register* activity at 10/1/2018 7:38:45 and completed the activity at 10/1/2018 7:42:30. Note that, as exemplified, multiple process instances run at the same time, i.e., the second event refers to Case ID 2, whereas the third event again refers to Case ID 1. Table 1 depicts the basic form of an event log. Typically, an event log includes more data attributes related to the process, e.g., the costs of an activity, account balance, customer id, etc.

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 ξ is defined as the Cartesian product of the aforementioned universes, i.e., ξ=$\mathcal{C} \times \mathcal{A} \times \mathcal{R} \times \mathcal{T} \times \mathcal{T}$. Furthermore, we define corresponding projection functions $\pi_{\mathcal{C}}: \xi \to \mathcal{C}$, $\pi_{\mathcal{A}}: \xi \to \mathcal{A}$, $\pi_{\mathcal{R}}: \xi \to \mathcal{R}$ and $\pi_{\mathcal{T}}: \xi \to \mathcal{T} \times \mathcal{T}$, where, 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) where t_s and t_c represent the start and complete time of event e. $L \subseteq \xi$ is defined as an event log. Also, we consider T_s and T_c as the start and completion time of the event log respectively.*

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).

System Dynamics System dynamics modeling describes a collection of approaches, techniques, and tools, that help in understanding how complex systems change over time [16]. It allows us to model complex, dynamic systems, in a structured manner and to capture the factors affecting the behavior of such a system by modeling the typically *nonlinear behavior* of such systems. Within system dynamics, we use a specific modeling notation, i.e., a *stock-flow diagram*, that allows us to simulate possible future behavior of a system [10], e.g., a (business) process. Such a diagram captures the different relations between a given collection of variables. Moreover, it allows us to calculate, subject to the aforementioned relations, the future values of these variables during different steps in time. The basic structure of a system dynamics model is a set of mathematical equations such as first-order differential (or integral) equations.

A stock-flow diagram consists of three basic elements, i.e., *stocks*, *flows* and *variables*. A stock represents any entity that is able to accumulate over time, e.g., the number of patients waiting in a hospital. A flow is either an *inflow* or *outflow*. An inflow increases the accumulated entity represented by a stock, whereas an outflow reduces the accumulated entity. Finally, any environmental factor that is able to influence the in-/outflow of a stock is modeled as a variable. A variable is also able to influence other variables. Furthermore, the value of a stock, in turn, is able to influence a variable. In Fig. 3, an example stock-flow diagram is shown and the equation depicted on the right-hand side of Fig. 3



$$st1 = st1_0 + \int_{t_0}^{t_n} (+f_1 - f_2)\mathrm{d}t \quad (1)$$

Fig. 3: A simple example stock-flow diagram and the underlying relation of Stock 1 (st_1) w.r.t. its in- and outflow (Flow 1 (f_1) and Flow 2 (f_2)).

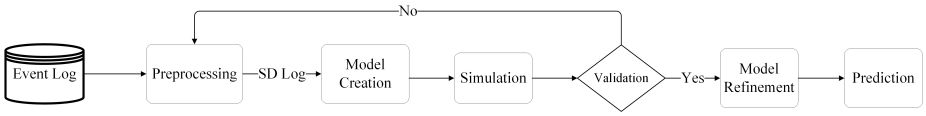


Fig. 4: General overview of the proposed framework. An event log is transformed into a *System Dynamics Log* (SD-Log), which describes the values of a collection of parameters of the process over time. The calculated values of the process parameters are used to populate a stock-flow diagram, which is used for simulation. After validation, model refinement and adding external parameters to the stock-flow diagram is possible. The model is used to predict future behavior of the process.

describes the underlying relation for the diagram. Consider t as time, Stock 1 is equal to the amount in Stock 1 at time t_0 plus the integral over the difference of the Flow 1 and Flow 2 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. For example, if the number of the patients arriving for a visit (pa) in a hospital is about 5 patients per hour (flow), and in one hour, 4 patients is being visited (pv), the number of patients waiting to be visited (pw) (stock) after 5 hours is 5 patients (time step 1 *hour* and at first there is no patient waiting ($pw_0 = 0$)) according to the equation $pw = pw_0 + \int_0^5 (+pa - pv)\mathrm{d}t$.

5 Approach

In this section, we describe the main approach presented in this paper, i.e., using system dynamics for scenario-based prediction of business processes, on the basis of past process executions. Consider Fig. 4, in which we present an overview of the proposed architecture. First, we transform a conventional event log into a *System Dynamics Log (SD-Log)*. An SD-Log describes the values of different process parameters over a predefined fixed set of time windows. Using an SD-Log the behavior of the process parameters, i.e., their patterns over time are identified. We use the SD-Log to populate the stocks, flows, and variables in a given stock-flow diagram. Ideally, the designed stock-flow model does not contain

any external parameters from outside the SD-Log. External parameters which are not provided in the SD-Log, complicate the validation step. Having both the values/patterns of the process parameters in the SD-Log and the simulation results, we check the validity of the model. If the model is unreliable, we change the time window granularity and repeat the aforementioned steps. When we have a reliable model, we are able to refine the stock-flow diagram to represent a specific scenario, e.g., by adding external parameters outside the process into it. Subsequently, we generate predictions by simulating the model.

5.1 Preprocessing

We populate the elements in the stock-flow diagrams with values originating from the process execution, extracted from the event log. We translate the conventional event log into a sequence of process parameter values, e.g., the arrival rate of cases for the execution of the process, measured per window of time. Hence, we first transform the event log into an SD-Log, which describes the values of these parameters over a sequence of discrete time windows.

The first step in the transformation of an event log to be used in system dynamics is to find its *measurable aspects*. We refer to these measurable aspects as process parameters, which provide values for the stock-flow diagrams elements. To do so, we define *scopes* and *types*. A variable scope represents the entity that we measure whereas the type represents what we measure for a given scope. In principle, numerous scopes and associated types are possible, e.g., measuring how the number of patients waiting (type) to be visited (scope) in a hospital changes over time.

In the context of this paper, we define a collection of *standard scopes and types*. The scopes are defined based on the granularity of different perspectives of business processes. Particularly covering the *general process*, *organization level*, *control-flow/process milestones* and *roles/resources* perspectives of business processes. The type of a parameter can be a *rate*, a *duration* or a *number*. In Fig. 5b, we schematically present the aforementioned pre-defined types. Fig. 5a shows an example set of possible process parameters for the general perspective which is at the instance level and resource perspective at the event level. By taking the possible scopes and types of the process, we are able to generate a log with all possible process parameters which are usable in a stock-flow diagram used for prediction.

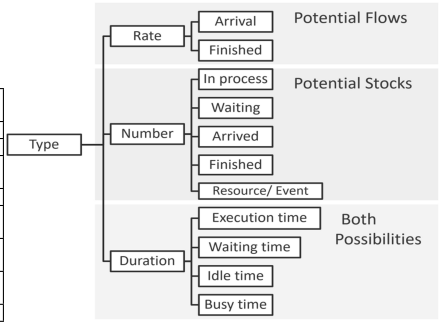
Using a predefined time window δ, in which we have the most similar behavior of the process parameters, we calculate values of process parameters and structure SD-Log. The time window can be derived in multiple ways, however, in this work, we consider selecting the time window based on ground knowledge.

Definition 2 (SD-Log). *Let $\mathcal{V}$ be a set of process parameters, δ be the selected time window, and $k \in \mathbb{N}$. An SD-Log is a function $SD: \mathcal{V} \times \{1, ..., k\} \to \mathbb{R}$, where $k = \lceil (T_c - T_s)/\delta \rceil$.*

We split the event log based on the selected time window (δ) and calculate the possible process parameters in each time window. Reconsider the

General Process Parameters	Resource Perspective Parameters
Arrival Rate of Case	Arrival Rate of Event
Finish Rate of Case	Finish Rate of Event
Number of Cases in Process	Number of Active Events
Busy Time of Process	Busy Time
Number of Unique Resources	Idle Time
Execution Time per Case	Execution Time per Event
Case Duration in Process	Waiting Time
...	...

(a) An example set of process parameters regarding the general process and resource perspective, e.g., types of "Arrival Rate of Case" and "Waiting Time" are rate and duration respectively.



(b) Three types of process parameters. The possible detail parameters regarding each type. The relation between the stock-flow diagram elements and the types of parameters is shown.

Fig. 5: Different types of process parameters and an example set of process parameters.

example of number of patients waiting to be visited in the hospital, assume that $L \subseteq \xi$ is an event log and δ=1 *hour*. Let v_1 and $v_2 \in \mathcal{V}$ be the *arrival rate of patients for the visit* and *number of patients waiting for the visit* in each time window. Assume the duration of L is 10 hours, implying $k = 10$. The SD-log regarding $\delta = 1$ *hour* and the two parameters is: $\{((v_1, 1), 12), ((v_1, 2), 11), ..., ((v_1, 10), 13), ((v_2, 1), 6), ..., ((v_2, 10), 8)\}$, i.e., representing that in the first hour, 12 patients arrived to be visited and 6 patients were waiting to be visited, and so on.

Duration and rate based parameters occur multiple times in one time window, we consider the average of the values in each time window. In some cases, the information regarding calculation of specific parameters is not included in the selected parts of the event log. If an activity or a resource does not appear in some parts of the event log, we assign specific policies to tackle the situation. For the duration based parameters, in the absence of a specific activity/resource in one of the time window, we consider a value of 0. If it is running in more than one time window, then in each time window the complete duration is taken into account. For the number and rate based parameters, in the absence of specific activity/resource in one of the time windows, we consider a value of 0.

Time Window Stability Test and Behavior Detection The ultimate goal of our approach is to have a model which is able to perform a scenario-based analysis. For this reason we need to have a model which behaves same as in reality. Therefore, the values, which represent the process parameters in the stock-flow diagram should behave similar in the selected time window δ. *Behavior Detec-*

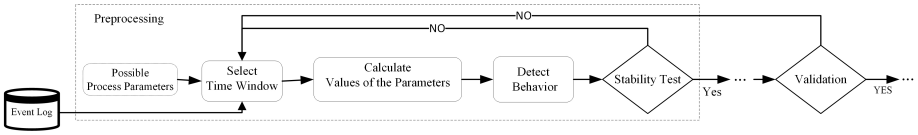


Fig. 6: Detailed view of preprocessing step in the approach (Fig. 4). *Time Window Stability* and *Behavior Detection* continue through the framework.

tion provides insights inside the patterns of values and the closest distribution which they can fit into. In *Behavior Detection*, we use Kolmogorov-Smirnov test [11], to discover the closest distribution in the selected time window. The coefficient of variance is also used to define the accepted variance among values of a parameter which we refer to it as *Stability Test*. Stability test helps us to inspect whether the values of the process parameters behave similar enough in provided δ or not. The threshold for the distributions similarity (p-value) and for the coefficient of variation (less than 1) is customized based on the level of freedom for accepting the difference between values in reality and simulation. The time window selection in preprocessing step Fig. 6, is a repetitive process, i.e., it continues until the stability test is passed and finishes when validation is passed for the simulation results.

An event log represents the events up to the specific point in time, therefore, the model can be populated with the values of parameters until the event log is recorded. In order to have an aggregated model, the values can be replaced by a single value, e.g., mean or their behavior which are defined by discovered distributions and attributes using behavior detection.

It should be noted, in some cases, variables in the event log are not supposed to show a similar behavior in each time window. Therefore, we mainly focus on the aggregated level of the variables. For instance, it is difficult to find the small time window of hours or days for the event log of the emergency room in a hospital. The arrival rate or duration of activities are not similar enough, however we are able to extend the time windows to capture more similar behavior.

5.2 Designing Stock-Flow Diagrams

In the second step, we design a stock-flow diagram with the process parameters contained in the SD-Log. Such a stock-flow diagram is either given such as the aggregated model in Section 6 or, designed based on the scenarios of interests. The generated SD-Log based on the scenarios and the detected behavior from the prepossessing step are the inputs for designing the stock-flow diagram. Fig. 5b provides some constraints on how to map the process parameters inside the SD-Log on the stock-flow diagram elements. The rate-based parameters are allowed to represent flows, the number-based parameters are allowed to represent stocks, and duration types are either flows, stocks or simple variables. For example, parameter “patient arrival rate for visit” per hour has the scope of activity-flow (visit) and type of arrival (rate), therefore in a stock-flow diagram it can act

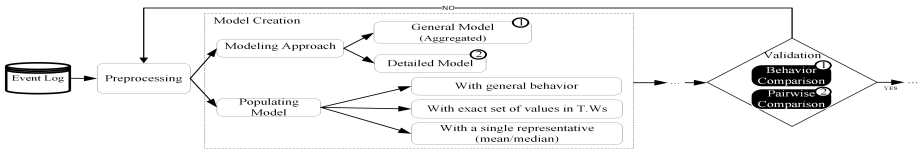


Fig. 7: The "model creation" step in the approach including "modeling approaches" and "populating approaches". Aggregated and detailed modeling regarding the abstraction level and three approaches for model population. The validation after simulation depends on the modeling approach of choice (labeled with 1 and 2).

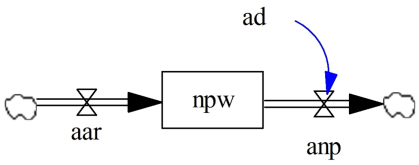
as an (in)flow for the activity "number of patients waiting for the visit". This parameter can be a stock element since it has the same scope (activity-flow) and type of number (waiting).

As Fig. 7 shows, two approaches are possible for designing the models, designing general models or detailed models. In the general models, we are looking for the aggregated level of the process without having extensive knowledge from inside the process. For designing detailed models more knowledge of the process is required. Since modeling benefits from detail knowledge of the process, it is possible to perform the validation including pairwise comparison (labeled as 2 in Fig. 7), which is explained in Section 5.3.

The main steps for designing the stock-flow diagrams and simulation are: (1) identify related process parameters for the desired scenario/change, (2) identify the relationship between the parameters, and (3) define the mathematical relationship between the parameters (equations). The design choices in the scenarios and model creation can be addressed more effectively using contextual knowledge from the process mining, i.e., decide on diverting more cases to a specific resource based on the idle time that performance analysis reveals.

After determining the involved parameters (which affect the target of simulation) in the scenario, and their relations, adding the equations lead to a complete stock-flow diagram. Observe that, the values of some of the parameters are directly derived from the SD-Log whereas the values of the stocks are calculated by mathematical equations based on flows. Also, the values of flows are allowed to be based on the values of variables. After defining the equation, the values of the stock-flow diagram elements get updated automatically (simulation).

For the example of patients visiting the hospital, a sample scenario for the hospital is to decrease the *number of patients waiting (npw)* (stock) to be visited in each time window δ. It is clear that the *average arrival rate (aar)* of patients (flow) and the *average duration of the visit (ad)* (variable) are process parameters which directly influence the *npw* in δ. Also, the *average number of patients being visited* in δ (one hour) (*anp*) (flow) can be derived from (*ad*). Fig. 8a shows the simple stock-flow diagram for the example scenario. The underlying equations for the designed stock-flow diagram which update the values for *npw* in each δ are mentioned in Fig. 8b. The values of *ad* and *aar* in each time window are



(a) The stock-flow diagram

(b) Relations of the variables
$npw = npw_0 + \int_0^{k\delta} (+aar - anp)\mathrm{d}\delta$
$anp = \frac{\delta}{ad}$, $ad = SD(ad, n-1)$
$aar = SD(aar, n-1)$

Fig. 8: Simple stock-flow diagram and equations representing underlying relations of the variables for predicting the number of patients waiting to be visited.

calculated using the presented SD function. We can simulate the model and calculate the values of npw for k time window of δ.

5.3 Simulation and Validation

The next step in the proposed approach is performing the simulation and validation, i.e., predicting and checking similarity of the values of the parameters in simulation and reality in each time window. We populate stock-flow diagrams with the values of the process parameters. Using the exact values of the parameters, for each element in the stock-flow diagram, current value gets updated by the value in the previous time window. From the specific point in time in which there is no value for the event log, we are able to use the most similar distribution derived from time window selection section. We generate the data based on the distribution and its features and populate our model with them. This is done based on the same time window as described by the SD-Log. Moreover, it is possible to use a representative for the values of parameters over time in the stock-flow diagram, i.e., the average of the values. Fig. 7 includes three aforementioned approaches. Also, it illustrates the populating approaches in the modeling step which affect validation step. We refer to SD_l and SD_m as functions which return the values from SD-log (SD_l) and values from simulating the model (SD_m). We want to ensure that any prediction we perform is meaningful. Consider $v_1, v_2, v_3 \in \mathcal{V}$, where $SD_l(v, n)$ represents the value of process parameters v in the SD-Log in time window n. Assume that v_3 is calculated based on the values of v_1 and v_2, $v_3 = F(v_1, v_2)$, where F represents the equation in the model. Performing simulation, $SD_m(v_3, n+1) = F(SD_m(v_1, n), SD_m(v_2, n))$, where $SD_m(v_1, n)$ and $SD_m(v_2, n)$ can be provided directly from SD-Log, i.e., $SD_m(v_1, n) = SD_l(v_1, n-1)$ or can be generated by their behavior, i.e., the distributions of the values in the SD_l. As Fig. 7 indicates the other possibility is to use a representative of values in SD_l such the average. Consider Table 2a and Table 2b as examples of the SD_l and the peer SD_m. We used the exact set of values in each day to perform simulation. The values of "Number of Cases Arrived" and "Number of Cases Finished" are updated in the SD_m by their previous values in the SD_l. The values of "Number of Cases in the Process" is calculate in each day using values of two other parameters.

Table 2: Part of an example SD_1 and the generated simulated log SD_m for the process perspective. TW, Arrival Rate, Finished Rate, Cases in the process are time window (one day), number of cases arrived for the process in each day, number of cases that finished in each day and number of cases which remains unfinished per day in the process respectively. Notation "S" indicates that the values of the parameters are simulated.

(a) Part of an example SD_l, including three process parameters for general perspective.

TW	Arrival Rate	Finished Rate	Cases in the Process	...
0	42	41	1	...
1	54	49	6	...
2	51	55	2	...
3	46	45	1	...
...	...	...	...	...

(b) Part of the generated SD_m containing results of simulation using the provided SD_l.

TW	SArrival Rate	SFinished Rate	SCases in the Process	...
0	42	41	1	...
1	42	41	2	...
2	54	49	7	...
3	51	55	3	...
...	...	...	...	...

In the validation step, the level of similarity of our simulation results with reality (SD-Log) is being investigated. Based on the populating approach in the simulation, we perform validation. We perform a pairwise comparison of values for each process parameter, which is defined as $SD_l(v, n) - SD_m(v, n)$, in the cases that we chose exact values from SD_l. In the cases that we use the aggregated level and the values are generated using SD_b, the validation comprises only similarity between the generated values for the process parameters in the simulation (their distributions). In fact, we compare whether simulated results are not significantly different from the SD-Log. Background knowledge allows us to define the maximum allowed difference considering the scale of variables, purpose of simulation and the underlying subject of simulation.

5.4 Prediction

In the prediction phase, we assess the effect of different scenarios, e.g., policy changes within the process, on the process performance characteristic of interest. We do so by systematically altering the values of the different parameters, i.e., elements in the stock-flow diagram, or changing the underlying equations. In the example of patients waiting to be visited, we are able to predict the change in the number of patients waiting to be visited, by changing the average patient arrival rate in the model. The results of the prediction (using `vensim`) are shown in Fig. 9. Assume the arrival rate is a normal distribution with a mean of 6 patients per hour and the average duration of visiting of each patient follows the normal distribution with the mean of 18 minutes. A change such as the arrival of 1 more patient per hour leads to a higher number of patients waiting over the sequence of time steps in 12 hours.

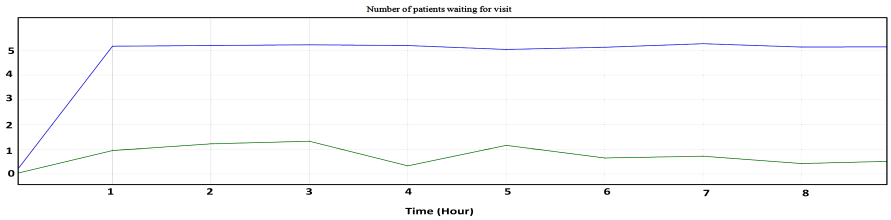


Fig. 9: The prediction result of a simple stock-flow diagram for the number of patients waiting for the visit. The green chart (bottom) shows the number of waiting patients in each time step when the average arrival rate is 6 and the blue chart (up) indicates the same variables with 7 as average arrival rate.

6 Evaluation

To validate the proposed approach, we performed different evaluations including real and synthetic event logs. In this section, we provide the results of evaluating our approach using real event logs. The purpose of the evaluation is to illustrate that our proposed approach is able to predict the result of specific changes in the process at an aggregated level without specific knowledge about the process model. Having the event log and generated SD-log, our model is able to simulate and show similar behavior to reality. After the validation, we are able to enrich our model for further change/policy analysis.

We applied our framework on the real event log *BPIChallenge2017* [21] to test the feasibility of the approach in reality. The event log includes different executions of processes for taking a loan by customers. Using our framework we assess different scenarios. Our goal is to achieve a stock-flow diagram which behaves same as reality and then perform further scenario-based analyses such as resource allocation. Note that, we have no explicit knowledge of any policy/change being applied in the process over the time period captured in the event log. We design a model at an aggregated level without having any information from the steps inside the process. Starting from the event log we choose the time window of one week and create the SD-Log regarding the *general process* perspective. The process parameters in a holistic view are case arrival rate, case finish rate and maximum capacity of the process in the time window. Fig. 10 shows the designed model for a general process. After performing time stability test and identifying the behavior of the values of parameters (distributions), we populate the designed model with the existing behavior of process parameters in the SD-Log. Since the parameters in the model are generated using random functions following the identified distributions, specific conditions regarding reality should be considered. The sum of the numbers of arriving cases and the number of cases in the process in each time window should be always less than the sum of the number of cases that finished and maximum capacity of the process in the same time window. The equation below represents the aforementioned conditions: $Case\,in\,Process + Case\,Arrival\,Rate < Finish\,Rate + Max\,Process\,Capacity$.

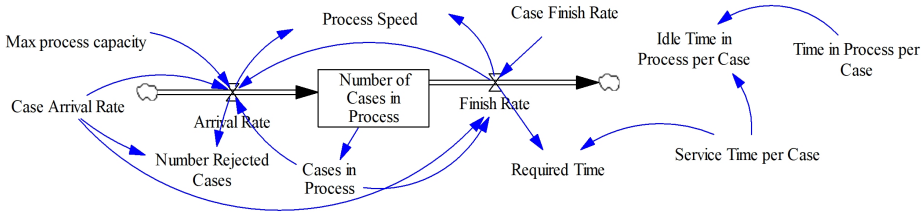


Fig. 10: Model designed based on the general process perspective. The generated behavior is similar to reality.

Table 3: Table of underlying equations in the general process perspective stock-flow diagram.

Stock-flow element	**Value**
Case Arrival Rate	*Poisson distribution with mean* 440 *case per week*
Case Finish Rate	*Poisson distribution with mean* 440 *case per week*
Number of Cases in Process	*Arrival Rate - Finish Rate*
Max process capacity	56
Number Rejected Cases	*Case Arrival Rate - Arrival Rate*
Time in Process per Case	*Normal distribution with average of* 8.30 *hours*
Service Time per Case	*Normal distribution with average of* 7 *hours*
Required Time	*Finish Rate * Service Time per Case*
Idle Time in Process per Case	*Time in Process per Case - Service Time per Case*

Also, the finish rate of the process in each time window cannot be bigger than the number of arrived cases and the number of cases in the process: $Finish\,Rate < Case\,Arrival\,Rate + Cases\,in\,Process$.

Table 3 shows the values and equations of the stock-flow diagram. As the result of validation shows the behavior of the elements in the model such as the number of cases in the process is similar to reality. Therefore our model as a valid model can be refined by inserting more external factors including resource efficiency in each time window.

Sample Scenario 1 Using the extended model in Fig. 11, we are able to predict several different scenarios, e.g., the effect of increasing in the arrival rate with the same finish rate on the number of rejected cases. Moreover, the effect of resource efficiency in the required time of resources can be predicted. For instance, in the process, 56 unique resources exist and in the case that they work 48 hours per week for the current state of the process, variables *Resource Required Time* and *Resource Idle Time* show the needed and idle hours of the resources in the process. Inserting variable *Resource Efficiency* into the model, provides the possibility to manage the required time of resources realistically. Fig. 12 shows the changes in required time per week with the efficiency of 85 percent. It reflects the reality more clear regarding the facts that resources does not work whole time with full efficiency. Benefiting from the result, the business owners are able to set policies such as providing more resources or setting more working hours in the current state of the process. There are variety of scenarios, e,g. effect

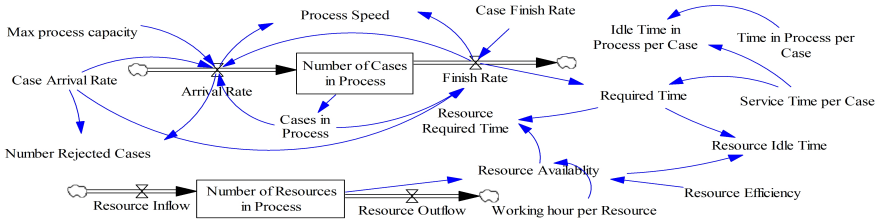


Fig. 11: Extended model based on the validated general model. The effect of changes in the resource aspects is predictable on the general aspect of the process. The effect of change in the aspects of process such as increase in arrival rate on the resources aspects is also predictable.

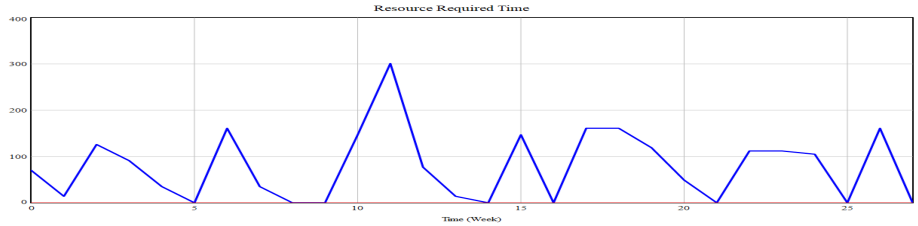


Fig. 12: The change in the required hours of resources based on the external factor, resource efficiency. Red chart (below) shows that with 100 percent efficiency the resource working hours regarding the current state of the process are enough. Blue chart indicates the required time after importing effect of resource efficiency of 85 percent.

of performing an activity such as "A_REGISTERED" in this event log using online forms instead of resources performing that manually. Changes in the service time in the process will influence the required hours of resources in the process.

This simple model illustrates the ability to perform scenario-based prediction regarding different aspects of a process and including external factors regardless of any knowledge inside the process. For the same purpose, existing techniques such as discrete event simulation need to know every step inside the process to simulate the behavior of the process even in aggregate level or the effect of changes. In long term policy analysis, the level of accuracy is highly different from the short term prediction such as discrete event simulation. Moreover, as the extended model shows the effects of different factors which may not explicitly exist in the process, e.g., resource efficiency can be considered in the results.

7 Conclusion

The approach presented in this paper provides a platform for organizations to inspect, in a scenario-based manner, the effect of changes in the process on

process performance metrics of interest. We introduced a novel approach where we use system dynamics to predict the future state of the process. Our approach is based on the past behavior of a process captured in the form of event logs. The past behavior is transformed to a set of values of the process parameters over time. We evaluated our framework, choosing the aggregated level of a process and scenarios regarding changing arrival/finish rate and capacity of the process. The evaluation is based on a real event log, and the results demonstrate the ability of the proposed approach to predict the effects of changes similar to the way which would happen in reality. Moreover, presenting the stock-flow diagram of the specific scenario (the effect of increasing the arrival rate of cases and resource efficiency in the process), for the real event log, shows the feasibility of the approach.

Since this paper is the first work combining process mining and system dynamics, there are ample opportunities to extend our work. As a next step, we consider the automated generation of system dynamics models focusing on the general prospective. Furthermore, we aim to extend the scopes and types beyond the general perspective, control-flow and resource dimension, which exist in this work such as organization level. Using knowledge inside the process, the balance between aggregation level and the accuracy is also an interesting next step. Finally, we aim to investigate to what degree time window selection can be completely automated.

Acknowledgement

This work is partially funded by the German Research Foundation (Deutsche Forschungsgemeinschaft –DFG) in the context of the Cluster of Excellence Internet of Production (EXC 2023, 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 (2015)
2. van der Aalst, W.M.P.: Process Mining - Data Science in Action. Springer (2016)
3. Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Bruno, G.: Automated discovery of structured process models from event logs: The discover-and-structure approach. Data Knowl. Eng. **117**, 373–392 (2018)
4. Carmona, J., van Dongen, B.F., Solti, A., Weidlich, M.: Conformance Checking - Relating Processes and Models. Springer (2018)
5. van Der Aalst, W.M.P.and Dustdar, S.: Process mining put into context. IEEE Internet Computing **16**(1), 82–86 (2012)
6. 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. pp. 319–336 (2008)

7. 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)
8. 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, Eindhoven, The Netherlands, September 10, 2014. p. 46 (2014)
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. pp. 130–134 (2015)
10. Pruyt, E.: Small system dynamics models for big issues: Triple jump towards real-world complexity (2013)
11. Razali, N.M., Wah, Y.B., et al.: Power comparisons of shapiro-wilk, solmogorov-smirnov, lilliefors and anderson-darling tests. Journal of statistical modeling and analytics **2**(1), 21–33 (2011)
12. Rosenberg, Z., Riasanow, T., Krcmar, H.: A system dynamics model for business process change projects. In: International Conference of the System Dynamics Society. pp. 1–27 (2015)
13. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering colored Petri nets from event logs. STTT **10**(1), 57–74 (2008)
14. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering simulation models. Inf. Syst. **34**(3), 305–327 (2009)
15. 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)
16. Sterman, J.D.: Business dynamics: systems thinking and modeling for a complex world. No. HD30. 2 S7835 2000 (2000)
17. Suriadi, S., Ouyang, C., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Event interval analysis: Why do processes take time? Decision Support Systems **79**, 77–98 (2015)
18. 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
19. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: Advanced Information Systems Engineering - 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings. pp. 477–492 (2017)
20. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. ACM Trans. Knowl. Discov. Data **13**(2), 17:1–17:57 (Mar 2019), http://doi.acm.org/10.1145/3301300
21. Van Dongen, B.F. (Boudewijn): Bpi challenge 2017 (2017). https://doi.org/10.4121/UUID:5F3067DF-F10B-45DA-B98B-86AE4C7A310B