Analysing and Improving Business Processes through Hybrid Simulation Model: a Case Study *

Francesca Meneghello[1,2], Massimo Coletti[3], Debora Di Marco[3], Massimiliano Ronzani[1], Chiara Di Francescomarino[4], and Chiara Ghidini[5]

[1] Fondazione Bruno Kessler, Trento, Italy {fmeneghello,mronzani}@fbk.eu
[2] Sapienza University of Rome, Rome, Italy
[3] Ernst & Young (EY) {Massimo.Coletti,Debora.Di.Marco}@it.ey.com
[4] University of Trento, Trento, Italy
[5] Free University of Bozen-Bolzano, Bolzano, Italy

Abstract. The increasing amount of process execution data, i.e. the event logs stored by the company, can be exploited using Business Process Simulation (BPS). BPS serves as a valuable tool for business analysts, enabling them to analyze and compare business processes and identify changes that optimize key performance measures. Especially when evaluating alternative scenarios, it is crucial to start with an accurate simulation of the current process. Recent research in the field of BPS has demonstrated that Hybrid Simulation Model (HSM) approaches reliably replicates business process behaviour, overcoming the unrealistic or oversimplified assumptions often found in traditional discrete event simulators. In this paper, we present a case study conducted in collaboration with EY, where we apply the HSM to a real-life business process log. This study demonstrates the benefits of the HSM for business process analysis and its potential to improve process performance.

Keywords: Business Process Simulation · Hybrid Simulation · Optimization

1 Introduction

Over the last few years, facilitated by the increasing amount of available data, researchers and analysts have shown interest in applying Business process simulation (BPS) techniques to real-life processes supported by ERP systems or databases. Especially in the healthcare field, numerous case studies have employed the BPS approach. For instance, it has been used to support healthcare managers in capacity management (CM) decisions [8], reduce patient waiting times through optimal physician scheduling [1], or enhance the performance of outpatient clinics by adjusting the number of receptionists, nurses, and doctors [13]. BPS [7] is a widely used and flexible technique for analyzing and enhancing business processes. It consists of simulating the behaviour of business processes by defining a simulation model, which extends a basic process model with additional probabilistic information, such as case arrival rates, activity durations, routing

* This work was partially supported by the Italian (MUR) under PRIN project PINPOINT Prot. 2020FNEB27, CUP H23C22000280006 and H45E21000210001 and PNRR project FAIR- Future AI Research (PE00000013), under the NRRP MUR program funded by NextGenerationEU.

probabilities and resource utilisation. This enriched model enables the generation of a large number of process runs, providing valuable insights into process performance and identifying critical areas for improvement. After identifying the critical points of the process, the simulation model can be leveraged to create various what-if scenarios, allowing the assessment of the potential impact of different changes to the process.

In the current state-of-the-art research of BPS, three main approaches can be identified: DDPS (Data-Driven Process Simulation), DL (Deep Learning), and HSM (Hybrid Simulation Models). DDPS simulation models are discrete event simulation (DES) models, constructed using knowledge derived from process execution data, such as event logs [4, 11, 12]. While DDPS models benefit from their white-box properties and adaptability for what-if scenarios, they often involve unrealistic or oversimplified assumptions during the model-building process, due to the limited expressiveness of the DES simulator. The second group of approaches uses DL models to generate event logs trace by trace, without simulating them collectively [5]. While DL models outperform DDPS in capturing time perspectives, they are black-box models that are not suitable for analyzing simulations or creating what-if scenarios, thus limiting their usefulness for process analysts or managers. Finally, *Dsim* (DeepSimulator) [6] and RIMS (Runtime Integration Machine Learning Simulation) [10] represent hybrid simulation methods. The main idea here is to retain a white-box simulation model, as in DDPS, but enhance it with information provided by DL models. In particular, an HSM model is defined by a BPS model integrating DL models for one or more simulation parameters [10]. Finally, [10, 6] demonstrate that the combinations of DDPS and DL approaches outperform the performance of each individual method.

In this paper, we present a case study conducted in collaboration with EY, where we apply the HSM approach to a real purchase-to-pay (P2P) process. Specifically, the data pertains to the *order-to-goods receipt* and *invoice-to-payment* segments of the P2P process. After extracting event logs from an ERP system based on SAP, we use them to create the HSM model. This model provides an accurate representation of the real process and enables us to analyze and evaluate potential performance improvements. The HSM model facilitates the identification of critical points in both the current process and potential overloaded scenarios, particularly in terms of waiting times. Finally, we demonstrate how the simulation model can be used to evaluate potential solutions for addressing the critical points identified in the process.

The remainder of the paper is structured as follows. First, Section 2 introduces the HSM approach, followed by Related Work (Section 3). Section 4 details the methodology applied and describes the case study. Section 5 presents the analysis and explores various what-if scenarios based on the previously defined HSM model. Finally, Section 6 concludes the paper.

2 Hybrid Simulation Model

A simulation model $\mathcal{M} = (\mathcal{N}, \mathcal{P})$, is composed of a business process model $\mathcal{N}$ (e.g., a BPMN), and a set of simulation parameters $\mathcal{P} = \{\mathcal{P}_R, \mathcal{P}_T, \mathcal{P}_C, \mathcal{P}_S\}$. These parameters address the resource, time, control flow, and inter-arrival time perspectives, respectively. Consider the simulation model in Figure 1 of an order-to-goods process, composed of a

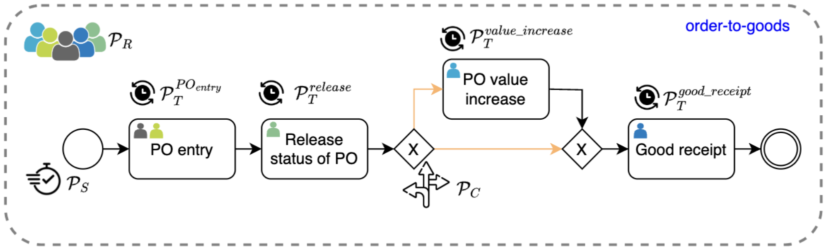


Fig. 1. Simulation model of order-to-goods process, composed of BPMN model ($\mathcal{N}$) and simulation parameters related to the resource ($\mathcal{P}_R$), time ($\mathcal{P}_T$), control flow ($\mathcal{P}_C$), and inter-arrival time ($\mathcal{P}_S$) perspectives.

BPMN model and the respective simulation parameters. $\mathcal{P}_R$ contains the five resources involved in the process and indicates the respective activities they can perform. For instance, the PO entry activity can be performed by gray and light green resources. The simulation time parameter, $\mathcal{P}_T$, defines the processing time needed to complete each activity in $\mathcal{N}$, such as $\mathcal{P}_T^{PO\ entry} \in \mathcal{P}_T$ which specifies the processing time for PO entry. Finally, $\mathcal{P}_S$ defines the rate at which orders arrive in the process and, $\mathcal{P}_C$ determines the path of the order, i.e., whether the PO value increase activity is executed or not.

HSM differs from DES simulation models by incorporating one or more predictive models into the simulation parameters, each addressing different aspects of the process. Predictive models are typically applied to $\mathcal{P}_\mathcal{T}$ and $\mathcal{P}_S$ parameters [6, 10], as they can capture the relationship between different trace elements and the distribution of output variables. Unlike probability distributions used in DES models, which do not consider previous activities, assigned resources or other attributes of the event, this approach aims to enhance simulation accuracy and address the often unrealistic or oversimplified assumptions of DES, which arise from the limitations of its simulation parameters.

Once $\mathcal{M}$ is defined, the traces composing a simulated log $\mathcal{L}^{sim}$ are generated by executing the model. Essentially, HSM models are discrete event simulation (DES) models that use stochastic methods to generate new traces based on inter-arrival times defined by $\mathcal{P}_S$. Each trace is simulated according to the control-flow semantics of the process model ($\mathcal{N}$) and the probabilities or predictions specified by $\mathcal{P}_\mathcal{C}$. Events are simulated with respect to the resource allocation for all traces in execution. A task is executed only if it is enabled and a resource is available. If no resource is available, the activity waits until the required resource is released and starts immediately once it becomes available. Resource availability can also be constrained by calendars that define the resource schedules, such as Monday to Friday, 9:00-18:00. Therefore, when calendars are included in $\mathcal{P}_R$, a resource can only perform a task if it is not already busy and if it falls within the time frame designated by its calendar. Finally, $\mathcal{P}_\mathcal{T}$ defines the processing time of each task and any potential waiting time between tasks, excluding delays caused by resource contention and the presence of resource calendars. The execution of each activity is recorded in $\mathcal{L}^s$ as an event that composes a trace. Finally, the cycle time of a completed trace is calculated as the difference between the timestamps of the last and first events in the trace.

Table 1. Event logs.

Log	#Traces	#Events	#Activity	#Resources	#Variants	Avg. trace length	Mean case Duration	Median case Duration	Log Timeframe
$\mathcal{L}^{PO}$	425	5810	34	27	108	6.82	3.97yrs	4.4yrs	12.57yrs
$\mathcal{L}^{IN}$	2762	5470	18	18	97	1.98	2.2mths	2.25hrs	12.07yrs
$\mathcal{L}^{G}$	372	3835	25	112	125	10.31	1.88yrs	6.8mths	10.92yrs

3 Related Work

The BPS simulation model is a well-known method for optimizing and evaluating possible scenarios, especially in healthcare processes, where the allocation of resources during shift work plays a crucial role in minimizing waiting times for services offered. Van Hulzen et al. in [8] present a real-life case study at the radiology department of a Belgium hospital, aiming to recommend solutions to capacity management (CM) regarding the required number of radiology devices, waiting area size, and reception staffing. Similarly, [13] improves the performance of the outpatient clinic process by redesigning the resource distribution and capacity of roles. Antunes et al. [1], on the other hand, used the simulation model to evaluate the schedule derived from optimization. The approaches described in [8] and [13] are not fully automated in creating simulation models and require expert intervention during the process. However, process experts often lack complete knowledge of the process and rely on data-driven approaches to analyze and detect non-compliant behavior. In addition, the DES simulation models defined in [8], [13] and [1] present several oversimplifications of the simulation parameters due to the limitations of the simulator employed. The HSM approach allows us to overcome the latter simplifications and enables the definition of the simulation model with the support of process experts or by discovering all the information directly from the data.

4 Methodology

In this section we describe the different steps of the case study as the pipeline reported in Figure 2. The aim of the pipeline is to analyze and improve two interconnected real processes recorded from a real ERP system based on SAP. First, the pipeline starts by extracting data from the ERP system and converting it into standard event logs, as described in Section 4.1. Event logs are then used to create an HSM model suitable for $\mathrm{RIMS}_{\mathsf{Tool}}$ [9], a hybrid business process simulator. Finally, in the last step, we exploit the simulation model to analyze the process performance under heavy load and to improve it by creating various what-if scenarios (Section 5). While the extraction and definition of the HSM with our framework can be generalized to other process models, being fully automated starting from the data, the analysis and what-if scenarios are specific of the process itself.

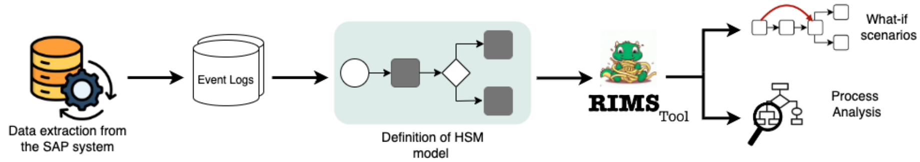


Fig. 2. Pipeline of the methodology applied to the case study

4.1 Data extraction

The SAP application tracks user changes to application data, storing these records as *change documents* in the database. Each *change document* is uniquely identified and reports the changes made by a single user through a specific *transaction*, a term used in SAP to refer to an application. A *transaction* relates to a particular business object, such as vendor master data, an order, or an accounting document. Specifically, *change document* may include updates to several database tables linked to the business object, like order headers and lines. The modified field is used to assign an activity identifier to the operation. For instance, if the due date field is altered, an activity like Due date delay or Due date anticipate is assigned based on comparing the previous and new values. Finally, EY employs a non-reversible algorithm to anonymize and convert this data directly into an event log in XES format, without the need for manual processing. Anonymization is applied only to information referring to resources or companies and their descriptions. These values are replaced with random ones, maintaining a 1:1 relationship with the originals to preserve resource workloads and social networks.

The extracted event logs, denoted as $\mathcal{L}$, capture the execution of *change documents* for the business object as a series of ordered events. A trace, in turn, contains the ordered events related to the same business object, which is identified by a unique case ID. The latter may also include trace attributes, which remain constant across the events, such as the vendor code, the released value, or the order type. Events contain additional attributes that identify the timestamp and the user who performed the activities. Therefore, each event is a detailed structure that includes the activity label, its timestamp, and the resource(s) involved in the activity. In our case, the extraction process can retrieve non-instantaneous events, meaning both the start and end timestamps are recorded. This structured representation of events in the event log provides a comprehensive view of the business process execution, enabling detailed analysis and allowing us to define a hybrid simulation model.

4.2 Case study and data description

The data considered in this case study pertains to the purchase-to-pay (P2P) process related to subcontracting in the construction and real estate industry. Specifically, we analyze three types of logs, as detailed in Table 1: one for the order-to-goods receipt part ($\mathcal{L}^{PO}$), another for the invoice-to-payment part ($\mathcal{L}^{IN}$), and a third that encompasses the entire order-to-payment process ($\mathcal{L}^{G}$). $\mathcal{L}^{G}$ is retrieved by combining the traces that are present in both $\mathcal{L}^{PO}$ and $\mathcal{L}^{IN}$, using their case IDs. Therefore, $\mathcal{L}^{G}$ in Table 1 contains

Table 2. Activities descriptions

Activity	Process	Frequency	Relative Frequency	Mean Duration	Median Duration	Max Duration
PO entry	PO	425	14.63%	1.82hrs	2.1 hrs	2.25hrs
Release status of PO	PO	1507	51.87%	1.70hrs	2hrs	2.25hrs
Good receipt	PO	828	28.5%	1.73hrs	2hrs	2.25hrs
Invoice entry	IN	387	7.07%	2hrs	2.25hrs	2.25hrs
Payment release	IN	1623	29.67%	1.82hrs	2hours	2hrs
Post clearing	IN	2779	50.8%	1.82hrs	2hrs	2.25hrs
Due date delay	IN	138	2.52%	1.75hrs	2hrs	2hrs

only a subset of the orders, excluding the incomplete ones present in $\mathcal{L}^{PO}$ and $\mathcal{L}^{IN}$ that affect the log timeframes. Although the *PO* and *IN* processes belong to the same purchase-to-pay (P2P) process at different stages, they operate independently of each other. Therefore, we use the individual logs to discover the simulation model for each process i.e. *PO* and *IN* processes, as they contain more traces and events, as shown in Table 1. Meanwhile, we use the global log to understand how these processes interact within the overall P2P system.

Figure 3 provides a simplified representation of the main activities in the two processes, *PO* and *IN*. The *PO* process begins with the creation of a new order, represented as PO entry. The order may transit through 7 different statuses, triggered by activities such as Release status of PO changed in 0/1/2/3/4/5/6. Following this, the client or company may adjust the order's value with PO value increase. Finally, when the goods are delivered, the Goods Receipt activity is recorded. The delivery may occur in multiple instalments over time, resulting in several records of the same activity. The *IN* process starts on average one month after the order has been created and approved. In particular, different activities can be performed to finalize the order, such as Invoice entry and Payment release, to proceed with the payment, which may include deferrals or payment in instalments.

From Table 1 and Table 2, we can observe that the average case duration is on the order of years for the *PO* process and months for the *IN* process, even though the main activities last at most two hours. This suggests that the waiting times between activities significantly impact the overall cycle times. Waiting times between activities may occur for various reasons. The main causes are typically resource contention when a resource is occupied by other activities, and the work schedules of the resources themselves, as explained in Section 2. However, in an *PO* process like our case study, waiting times are closely tied to the size and value of the order, as well as the requests, especially when the orders concern urbanisation contracts, which may require years to complete. In the *IN* process, delays can be influenced by payment extensions, delays, or potential changes to the final invoice but also by the nature of the process itself. In fact, in the construction and real estate industry, invoices are typically closed by the customer only after verifying the proper fulfillment of various guarantees by the supplier, which generally takes a long time. Indeed, $\mathcal{L}^{PO}$ log includes only traces that began within 6 months (from February 2010 to August 2010), but 36% of these traces continue for over 5 years.

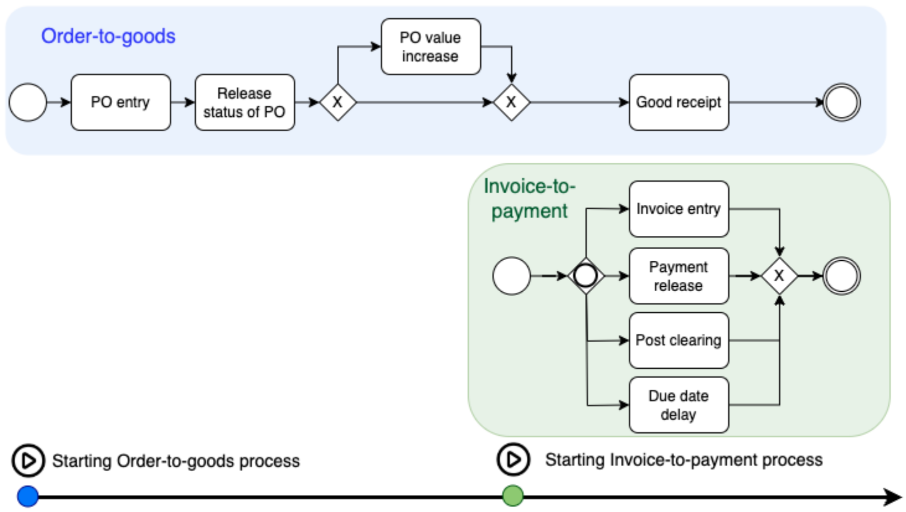


Fig. 3. Interaction between the *Order-to-goods* (PO) and *Invoice-to-payment* (IN) processes.

4.3 Definition of Hybrid Business Process Simulation

In this section, we describe how we define the HSM model using the extracted event logs, as detailed in Section 4.1. In particular, we outline the methods for discovering all elements of $\mathcal{M}$, including the process model $\mathcal{N}$ and the set of simulation parameters $\mathcal{P}$. To define the HSM model, we follow the approach proposed by RIMS [10], which is compatible with $\text{RIMS}_{\text{Tool}}$ [9] and allows us to customize the simulation parameters to properly represent our case study. Specifically, $\text{RIMS}_{\text{Tool}}$ requires as input a Petri net process model and the corresponding simulation parameters, which can be specified either through the tool's configuration or by defining specific ones.

Process Model ($\mathcal{N}$). To define $\mathcal{N}$, we first derive $\mathcal{N}^{PO}$ from $\mathcal{L}^{PO}$ and $\mathcal{N}^{IN}$ from $\mathcal{L}^{IN}$, as BPMN models, using the SplitMiner algorithm [2]. Then from $\mathcal{L}^{G}$, we identify the interactions between the two processes by the definition of the final $\mathcal{N}$. Specifically, we define when the *IN* process begins to handle the order that was processed by the first one, as shown in Figure 3. Subsequently, the two processes continue their execution in parallel. This approach allows us to create a unified process model while maintaining the independence of each process. Finally, $\mathcal{N}$ is transformed into a Petri net model as required by $\text{RIMS}_{\text{Tool}}$.

Control-flow parameters ($\mathcal{P}_C$). For each decision point in $\mathcal{N}$, we define the routing probabilities by aligning the traces in the event log with the discovered process model [4]. For instance, $\mathcal{P}_C$ includes the probability of executing the activity PO value increase from the XOR gateway present in the *PO* process, as shown in Figure 3.

Inter-arrival parameters ($\mathcal{P}_S$). The arrival time of a new trace is defined only for *PO* process since the start of *IN* is triggered by the first. The creation of a new order is estimated by an exponential distribution with a mean of 10 hours as we observe in $\mathcal{L}^{PO}$. Unlike RIMS [10], we do not use a time series to estimate $\mathcal{P}_S$, as the distribution function provides a simpler way to configure and evaluate the process with a high volume of orders (see Section 5). Finally, from $\mathcal{L}^{PO}$, we also identify a calendar that restricts arrivals i.e. Monday through Friday, from 9 a.m. to 6 p.m.

Time parameters ($\mathcal{P}_T$). As proposed in [10, 6], the $\mathcal{P}_T$ parameter is defined using two distinct LSTM (Long Short-Term Memory) models: one for predicting activity processing times and another for predicting waiting times between activities. These predictions are based on the current activity (for processing times) or the next activity (for waiting times), along with additional attributes such as the timestamp of the current activity, the day of the week, and inter-case features like work-in-progress and resource occupancy.

Resource parameters ($\mathcal{P}_R$). The resources involved in the simulation model are defined in $\mathcal{P}_R$. From $\mathcal{L}^G$, we observe that the resources involved are not shared among the two processes. Therefore, even though we define a unified simulation model, we keep the resources distinct from each other. In particular, resources are grouped into roles based on the activities they are allowed to perform by applying the method proposed in [3] to event logs. From $\mathcal{L}^{PO}$, six different roles are identified, while $\mathcal{L}^{IN}$ reveals five roles.

Finally, we set the number of traces generated in the initial HSM to 2000,[6] and the start timestamp of the simulation is aligned with that of $\mathcal{L}^{PO}$

5 Analysis and Improving

In this section, using the HSM model defined in Section 2, we first analyze the performance of the current process and its response to an increased order load, focusing on the waiting times, which significantly impact the cycle times in our case study. Based on the results of this analysis, we evaluate two what-if scenarios ($w1$ and $w2$) to determine whether they can reduce the process waiting times. Finally, $w1$ and $w2$ are validated by process experts as achievable what-if scenarios. For each scenario we performed 10 simulations and the reported results are the mean values from all simulations.

5.1 Analysis of waiting times

As mentioned in Section 4.2, both processes exhibit significant waiting times, as evidenced by the large final cycle times of the traces compared to the processing times of the main activities (Table 1 and Table 2). Before attempting to optimize process performance in terms of waiting times, we need to identify which aspects of the process

[6] As reported in Table 1, each log contains a different number of traces. Therefore, we assume 2000 as a plausible number of traces. For the same reason, a direct comparison between $\mathcal{L}^{sim}$ and the original logs $\mathcal{L}^{PO}$, $\mathcal{L}^{IN}$, and $\mathcal{L}^{G}$ is not feasible.

Table 3. Analysis of the results of the increased order load.

Log	#Traces	Median Cycle Times	Mean Waiting Times	Median Waiting Times	Mean Queue	Median Queue
$\mathcal{L}^{sim}$	2000	8.10mnths	41d	0.07d	0.009	0
$\mathcal{L}^{+25\%}$	2500	8.87mnths	42d	0.065d	0.014	0
$\mathcal{L}^{+50\%}$	3000	8.41mnths	40d	0.067d	0.013	0
$\mathcal{L}^{+75\%}$	3500	8.17mnths	39d	0.071d	0.013	0

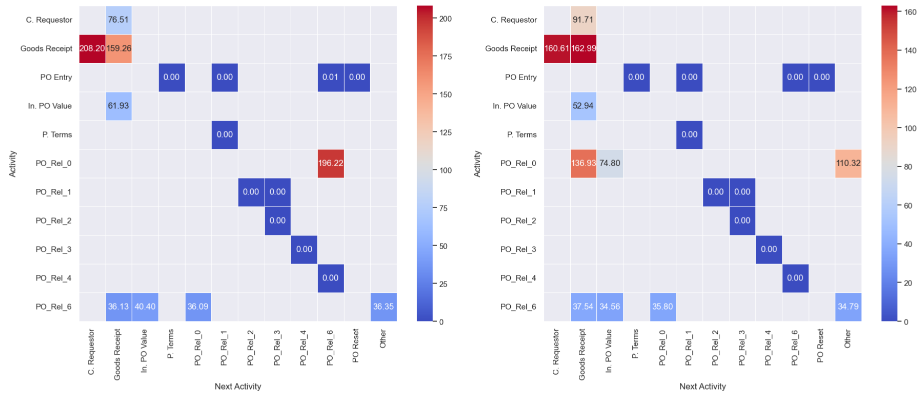
can be modified to create plausible what-if scenarios. For example, waiting times caused by external factors or the processing time of activities dictated by procedures to be followed cannot be changed. In our case study, potential optimizations to reduce waiting times may involve reallocating resources to better manage congestion within the process or making minor adjustments to the control flow.

To verify that the waiting times are not due to resource contention but are inherent to the process itself, we increase the number of traces simulated in the HSM model by 25%, 50%, and 75% compared to the initial setting, by defining three what-if scenarios $\mathcal{M}^{25\%}$, $\mathcal{M}^{50\%}$ and $\mathcal{M}^{75\%}$. Table 3 compares the queue, cycle and waiting times of the resulting simulated logs. Despite the substantial increase in the number of traces handled by the process, performance remains largely unchanged (Table 3).

Therefore, to minimize the waiting times in the process, we should focus on the control-flow perspective rather than rethinking the allocation of resources within roles. To pinpoint where waiting times are most prevalent related to the control-flow, we generated the heatmap plots displayed in Figure 4(a) and 5(a) to represent their distribution in $\mathcal{L}^{sim}$. Figure 4(a) illustrates the median waiting times, in days, between one activity and those directly following it in the *PO* process, while Figure 5(a) does the same for the *IN* process. For instance, when Goods Receipt activity is directly followed by Change Requestor in $\mathcal{L}^{sim}$ the median for the waiting times is 156 days (Figure 5(a)). Regarding the *PO* process we can observe that we have a strong presence of waiting times between the activity ⟨Good Receipt,Change PO requestor⟩, ⟨Good Receipt,Good Receipt⟩ and, finally, ⟨Release status of PO changed in 0,Release status of PO changed in 6⟩. For the first two pairs of activities, the waiting times are closely associated with the type of order and the specific goods requested. In contrast, the waiting times for the last pair of activities are related to the various stages the order may pass through before the goods are dispatched. On the other hand, for the *IN* process, the waiting times are more prevalent between the ⟨Payment Release, Reset Invoice Clearing⟩ and ⟨Payment Release, Post Clearing⟩ activities. Therefore, the activities that generate the most waiting times are those related to authorizing and executing payments.

5.2 Definition of what-if scenarios

Starting from the analysis presented in Section 5.1, we define two what-if scenarios — one for each process — with the goal of minimizing waiting times and, consequently, reducing the overall cycle times of the traces. For the *PO* process, we aim to minimise the waiting time between ⟨Release status of PO changed in 0,Release status of PO



(a) (b)

Fig. 4. Heatmap representing the median waiting times, in days, between *PO* activities performed in $\mathcal{L}^{sim}$ and $\mathcal{L}^{sim}_{w1}$.

changed in 6⟩, as Figure 4(a) shows that the waiting times between ⟨Release status of PO changed in 6,Release status of PO changed in 0⟩, by contrast, are lower. Therefore, as our first what-if scenario $\mathcal{M}^{w1}$, we modified the initial simulation model to ensure that the activity Release status of PO changed in 0 is executed before the activity Release status of PO changed in 6. The simulated log $\mathcal{L}^{sim}_{w1}$ is obtained from the simulation of $\mathcal{M}^{w1}$, and the resulting waiting times are shown in Figure 4(b).

Since the position of Release status of PO changed in 0 is modified in $\mathcal{M}^{w1}$, this activity is now followed by different ones compared to the initial simulation model $\mathcal{M}$. Indeed, Figure 4(b) shows new directly follow relations between activities in the *PO* process, such as ⟨Release status of PO changed in 0,Goods receipt⟩, ⟨Release status of PO changed in 0,Invoice PO value⟩, and ⟨Release status of PO changed in 0,Other⟩. Specifically, for these new relations, we observe moderate waiting times, indicating that the high waiting times originally associated with ⟨Release status of PO changed in 0,Release status of PO changed in 6⟩ are redistributed. As a result, the median cycle times of traces in $\mathcal{L}^{sim}_{w1}$ remain similar to those in $\mathcal{L}^{sim}$ without any statistically significant difference, at 7.92 months and 8.10 months, respectively (see Table 3). Therefore, the application of the $w1$ what-if scenario in the real process should be evaluated by the process owner based on their preferences regarding the distribution of waiting times, taking into account factors such as client satisfaction and costs.

The second what-if scenario ($w2$) defined for the *IN* concerns the activities ⟨Payment Release, Reset Invoice Clearing⟩. In the process model $\mathcal{N}$ contained in $\mathcal{M}$, the execution of the activity Reset Invoice Clearing in $\mathcal{L}^{PO}$ is always followed by the activity Post Clearing. Thus, we hypothesise to postpone the execution of Reset Invoice Clearing with the Post Clearing activity. Specifically, whenever Reset Invoice Clearing occurs in the process, it is combined with Post Clearing and performed as a single activity, with the processing time being the sum of the individual processing times. Figure 5(b) shows the resulting waiting times in $\mathcal{L}^{sim}_{w2}$, which is generated by running $\mathcal{M}^{w2}$. With the what-if

Fig. 5. Heatmap representing the median waiting times, in days, between *IN* activities performed in $\mathcal{L}^{sim}$ and $\mathcal{L}^{sim}_{w2}$.

scenario $w2$ we are able to eliminate the waiting times between Payment Release and Reset Invoice Clearing without a relevant increase of the waiting times on $\langle$Payment Release,Post Clearing$\rangle$. In this case, we do not observe a reduction in the median cycle times in $\mathcal{L}^{sim}_{w2}$ because Reset Invoice Clearing is not a frequent activity and therefore does not impact most traces. However, Figure 5(b) shows that scenario $w2$ effectively reduces waiting times, which, in turn, lowers costs and resource utilization, as the two activities are performed together.

6 Conclusion

In this paper, we present the results of a case study conducted in collaboration with EY, focusing on the application of Hybrid Simulation Model (HSM) approach to enhance the analysis and improvement of business processes within a real purchase-to-pay (P2P) process. Leveraging event logs extracted from an ERP system, we developed an HSM model that accurately reflects the actual process, without the need for prior domain knowledge. The HSM model enabled us to analyze the process and identify critical points, particularly concerning waiting times. Building on the identified weaknesses, we demonstrated how to define and evaluate two what-if scenarios to reduce these waiting times. Moreover, we demonstrated the ability and flexibility of the HSM approach to generate various what-if scenarios. This capability proved invaluable for assessing the potential impact of different changes and proposing targeted solutions to optimize performance. The limitation of our approach lies in its inability to represent external factors, as well as information absent from the event log, that influence process performance. These factors cannot be simulated and are therefore not addressable in a what-if scenario for potential optimization. Nevertheless, we show that the HSM approach offers a powerful tool for process managers and analysts, facilitating a detailed understanding of process behavior and supporting decision-making to improve process performance. Future work will focus on exploring and defining additional what-if scenarios to further optimize the process, incorporating detailed information about individual orders.

References

1. Antunes, B.B., Manresa, A., Bastos, L.S., Marchesi, J.F., Hamacher, S.: A solution framework based on process mining, optimization, and discrete-event simulation to improve queue performance in an emergency department. In: Business Process Management Workshops: BPM 2019 International Workshops, Vienna, Austria, September 1–6, 2019, Revised Selected Papers 17. pp. 583–594. Springer (2019)
2. Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Polyvyanyy, A.: Split miner: automated discovery of accurate and simple business process models from event logs. Knowledge and Information Systems **59**, 251–284 (2019)
3. Burattin, A., Sperduti, A., Veluscek, M.: Business models enhancement through discovery of roles. In: 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM). pp. 103–110. IEEE (2013)
4. Camargo, M., Dumas, M., González-Rojas, O.: Automated discovery of business process simulation models from event logs. Decision Support Systems **134**, 113284 (2020)
5. Camargo, M., Dumas, M., Rojas, O.G.: Learning accurate LSTM models of business processes. In: Business Process Management - 17th International Conference, BPM, Proceedings. LNCS, vol. 11675, pp. 286–302. Springer (2019)
6. Camargo, M., Dumas, M., Rojas, O.G.: Learning accurate business process simulation models from event logs via automated process discovery and deep learning. In: Advanced Information Systems Engineering - 34th International Conference, CAiSE 2022, Proceedings. LNCS, vol. 13295, pp. 55–71. Springer (2022)
7. Dumas, M., La Rosa, M., Mendling, J., Reijers, A.H.: Fundamentals of business process management. Springer (2018)
8. van Hulzen, G., Martin, N., Depaire, B., Souverijns, G.: Supporting capacity management decisions in healthcare using data-driven process simulation. Journal of Biomedical Informatics **129**, 104060 (2022)
9. Meneghello, F., Di Francescomarino, C., Ghidini, C.: Rimstool: a hybrid simulator for business processes (2023)
10. Meneghello, F., Di Francescomarino, C., Ghidini, C.: Runtime integration of machine learning and simulation for business processes. In: 5th Int. Conf on Process Mining, ICPM 2023. pp. 9–16. IEEE (2023)
11. Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M.P.: Supporting automatic system dynamics model generation for simulation in the context of process mining. In: Business Information Systems - BIS 2020, Proc. LNBIP, vol. 389, pp. 249–263. Springer (2020)
12. Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M.P.: Discovering simulation models. Inf. Syst. **34**, 305–327 (2009)
13. Zhou, Z., Wang, Y., Li, L.: Process mining based modeling and analysis of workflows in clinical care-a case study in a chicago outpatient clinic. In: Proceedings of the 11th IEEE international conference on networking, sensing and control. pp. 590–595. IEEE (2014)