The use of process mining in business process simulation model construction: structuring the field

MARTIN, Niels; DEPAIRE, Benoit & CARIS, An (2016) The use of process mining in business process simulation model construction: structuring the field. In: Business & Information Systems Engineering, 58 (1), p. 73-87.

DOI: 10.1007/s12599-015-0410-4
Handle: http://hdl.handle.net/1942/20472

Accepted manuscript version. The final paper is published in Business & Information Systems Engineering (volume 58, issue 1, pages 73-87). The final publication is available at link.springer.com.

The use of process mining in business process simulation model construction: structuring the field

Niels Martin[1], Benoît Depaire[1], An Caris[1,2]

[1]*Hasselt University, Agoralaan Building D, 3590 Diepenbeek, Belgium*

[2]*Research Foundation Flanders (FWO), Egmontstraat 5, 1000 Brussels, Belgium*

niels.martin@uhasselt.be

Abstract:

This paper focuses on the use of process mining (PM) to support the construction of business process simulation (BPS) models. Given the useful BPS insights that are available in event logs, further research on this topic is required. To provide a solid basis for future work, this paper presents a structured overview of BPS modeling tasks and how PM can support them. As directly related research efforts are scarce, a multitude of research challenges are identified. In an effort to provide suggestions on how these challenges can be tackled, an analysis of PM literature shows that few PM algorithms are directly applicable in a BPS context. Consequently, the results presented in this paper can encourage and guide future research to fundamentally bridge the gap between PM and BPS.

Keywords:

Business process simulation; Process mining; Event log knowledge; Simulation model construction

1. Introduction

Business process simulation (BPS) refers to the imitation of business process behavior using a simulation model (Melão and Pidd 2003). By mimicking the real system, BPS can identify the effects of operational changes prior to implementation (Melão and Pidd 2003) and contribute to the analysis and improvement of business processes (Rozinat et al 2008b).

BPS models are typically based on insights from process documentation, expert interviews and observations (Rozinat et al 2009), which can provide a biased process view as e.g. employees might behave differently when observed (Martin et al 2014a). Therefore, efforts to improve simulation model realism are valuable.

In this respect, a solution can originate from process-aware information systems, such as CRM systems, which record process execution information in event logs (van Beest and Măruşter 2007). These are collections of events, e.g. the start of order packing, associated to a case such as an order. It minimally contains an ordered set of events for each case, but typically also includes case

and event attributes. The extraction of knowledge from event logs belongs to the process mining (PM) field (van der Aalst 2011). This knowledge will be used as an additional input for the construction of a BPS model (Martin et al 2014a).

Given the potential of PM to improve BPS models and the clear connection between both domains, further research on their combination is required. However, to fundamentally integrate PM in BPS model construction, a profound insight in both BPS modeling tasks and methods to extract knowledge from event logs is required. This paper contributes towards bridging the gap between both areas of expertise by presenting a structured overview of the state of the art on the use of PM in BPS model construction and the challenges ahead. Starting from a conceptual BPS model, a series of BPS modeling tasks are defined. For each of them, the potential of event log knowledge to support its specification is outlined. By comparing these insights to the state of the art on the use of PM in BPS, a multitude of research challenges is identified. Moreover, useful starting points for future research are outlined by including relevant existing PM methods. In this way, a solid basis for future work is provided.

Due to the nature of its contribution, the paper is valuable for both the PM and BPS domain. On the one hand, PM researchers are offered a detailed overview of research issues that require attention. On the other hand, simulation experts gain insight in the potential of event log knowledge to support BPS modeling tasks. By presenting a broad and structured overview of the field, this paper marks a crucial first step to kick-start and guide new research on this important topic.

The relevance of this work's topic is supported by the Process Mining Manifesto, which marks the use of PM in BPS as one of the key challenges in PM research (van der Aalst et al 2012a). The use of PM in simulation also relates to several BPM use cases (van der Aalst 2013a). Discovery of models (DiscM) from event logs and performance analysis (PerfED) of event data provide new insights to repair (RepM) and extend (ExtM) existing simulation models. These more realistic simulation models allow for enhanced performance analysis (PerfM) and process improvement suggestions (ImpM). This paper focuses on the research efforts related to use cases DiscM and PerfED to enable the other aforementioned use cases.

The remainder of this paper is structured as follows. The next section introduces a running example and discusses the general BPS model structure. The third section details the use of PM to support BPS model construction. A discussion of the research findings and overview of the main conclusions is provided in respectively the fourth and fifth section.

2. Preliminaries

2.1. Running example

Throughout this paper, the simplified order picking process of a fictitious company will serve as a running example. The process model is visualized in figure 1.

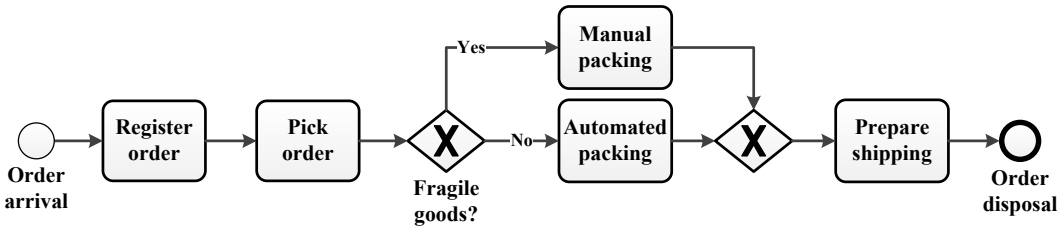

Figure 1: Running example

When an order arrives in the order picking process, it is registered by checking if all required order and billing information is available. Afterwards, the order is picked and packed. When the order contains fragile goods, it is packed manually. Otherwise, packing is automated using a high-tech machine. Following order packing, the goods are prepared for shipping and transmitted to the outbound logistics process.

The company's process is supported by a process-aware information system recording event logs. An excerpt of an example event log is shown in table 1, where each line is an event associated to a particular order. For instance: the first line refers to the start of 'Register order' for order 52156 by administrative clerk Sue. Note that table 1 only presents a sample event log structure as event logs can contain varying levels of information, e.g. additional attributes can be included or only end events might be recorded.

Table 1: Excerpt from example event log

Order id	Timestamp	Activity	Event type	Resource	…
...	...	…	...	…	…
52156	2015-08-25 14:19:22	Register order	start	Administrative clerk: Sue	…
52148	2015-08-25 14:20:07	Pick order	start	Order picker: Mike	…
52156	2015-08-25 14:21:08	Register order	end	Administrative clerk: Sue	…
52141	2015-08-25 14:24:40	Automated packing	start	Packing machine: P1	…
52141	2015-08-25 14:25:31	Automated packing	end	Packing machine: P1	...
52148	2015-08-25 14:29:04	Pick order	end	Order picker: Mike	…
52157	2015-08-25 14:31:22	Register order	start	Administrative clerk: Ruth	…
52157	2015-08-25 14:33:09	Register order	end	Administrative clerk: Ruth	…
…	…	…	…	…	…

2.2. General BPS model structure

To structure the discussion in the remainder of this paper, a general BPS model structure needs to be defined. To this end, the conceptual model introduced in Martin et al (2014b) is used, which is based on a review of simulation literature on BPS model components such as Tumay (1996) and Kelton et al (2010).

The eight main BPS model building blocks and their mutual relationships are visualized in figure 2. A description and illustration within the context of the running example are provided in table 2.

As particular building blocks logically belong together, e.g. each resource has a schedule, these are combined using dashed rectangles in figure 2 and in the last column of table 2.

Each building block in figure 2 is color-coded, representing the size of the literature base on the use of PM in a BPS context regarding this building block, as will become apparent in section 3. A red marking indicates that no research efforts are identified. For orange markings, some preliminary work is done which can be extended in future research. Green markings reflect that basic research has been performed, but improvement potential can be present. Underlining is also added: no, single and double underlining correspond to green, orange and red markings.

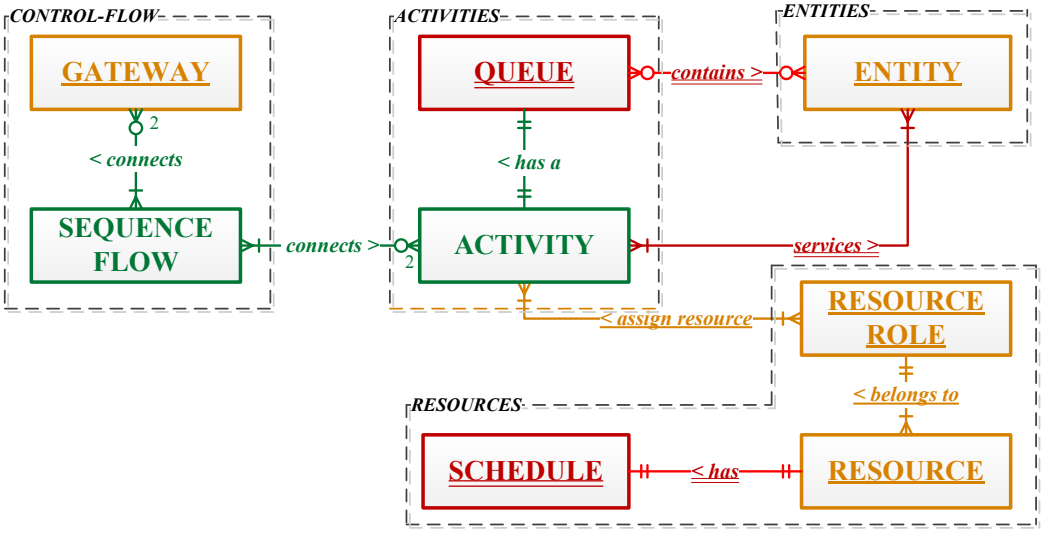

Figure 2: General BPS model components (Martin et al 2014b)

Table 2: Overview of BPS model building blocks (Martin et al 2014b)

Building block	Description	Example	Aggregated building block
Entity	Dynamic object that flows through the process	Order	Entities
Activity	Model component providing service to an entity	Pick order	Activities
Queue	Model component containing entities for which the required resources to perform an activity are not available	Queue for 'Manual packing'	
Gateway	Model component influencing the routing of entities	Fragile goods? Yes / No	Control-flow
Sequence flow	Model component expressing relationships between activities and gateways	Arrow between 'Register order' and 'Pick order'	
Resource	Model component responsible for the execution of activities	Order picker Mike	Resources
Schedule	Model component expressing the presence of a resource for the process in each time slot	Schedule of Mike	
Resource role	Group of resources performing similar activities	Order picker	

3. The use of process mining in a BPS context

This section details the use of PM to support BPS model construction. To this end, modeling tasks are specified in table 3 for each of the aggregated BPS building blocks in figure 2, extending the efforts in Martin et al (2014b). For each of them, this section outlines (i) the intrinsic potential of

event data, (ii) the state of the art on the use of PM to support this modeling task, as well as PM references that might provide a promising starting point for future research and (iii) key challenges that need to be addressed. Literature was gathered using electronic bibliographical databases such as EBSCOHost and Google Scholar, using various combinations of search terms such as 'business process simulation', 'simulation', 'process mining', 'event log' and 'workflow management system'. Moreover, an ancestry and descendancy approach is applied to relevant papers, i.e. publications in their reference list are examined and papers citing this reference are identified using Google Scholar (Johnson and Eagly 2000). Both publications in scientific journals and peer-reviewed conference proceedings are included. All relevant literature is summarized in table 3. The cited references will be discussed when the associated BPS modeling task is treated, as indicated between brackets in the first and second column of table 3.

The remainder of this section is composed of subsections focusing on a single modeling task, where the discussions are retained at a general, platform-neutral level. Specifying the relationship between the issues raised on the one hand and the specific platform used to perform the simulation on the other hand is beyond the scope of this paper. Considerations on this matter will become relevant when future work will tackle the identified research challenges. Note that each subsection title also includes a reference to the BPM use case it is most related to. This is either use case DiscM, referring to the automated discovery of e.g. a control-flow or organizational model from an event log, or PerfED, reflecting the combined use of models and timed event data (van der Aalst 2013a).

Table 3: Overview of relevant references

Model component	**Modeling tasks**	**Relevant process mining references**	**State of the art on the use of process mining in a BPS context**
Entities (3.1)	Entity attributes (3.1.1)	-	-
	Entity type (3.1.2)	Trace clustering: * Greco et al (2006) * de Medeiros et al (2008) * Bose and van der Aalst (2009) * Song et al (2009) * Bose and van der Aalst (2010) * Veiga and Ferreira (2011) * De Weerdt et al (2013) Process cubes: * van der Aalst (2013c)	-
	Entity arrival rate (3.1.3)	Song and van der Aalst (2007)	Rozinat et al (2009) Martin et al (in press a) Martin et al (in press b)
Activities (3.2)	Activity definition (3.2.1)	* Günther et al (2010) * Szimanski et al (2013) * Baier et al (2014)	-
	Duration (3.2.2)	Activity duration: * van der Aalst et al (2011) * van der Aalst et al (2012b) * Nakatumba (2013) * Wombacher and Iacob (2013) * Rogge-Solti et al (2014)	Activity duration: * Rozinat et al (2009) * Pospíšil and Hruška (2012) * Pospíšil et al (2013) Workload-dependent processing speed: * Nakatumba and van der Aalst (2010)

			* Nakatumba et al (2012)
	Resource requirements (3.2.3)	Staff assignment rules: * Ly et al (2006) * Liu et al (2008) * Huang et al (2011) * Senderovich et al (2014a)	-
	Queue discipline (3.2.4)	Q-log: * Senderovich et al (2014b) Queue length: * Senderovich et al (2015) * Senderovich et al (in press)	-
	Queue abandonment condition (3.2.5)	Q-log: * Senderovich et al (2014b) Entity patience: * Senderovich et al (2015)	-
	Interruptibility (3.2.6)	Interruptions in service logs: * Senderovich et al (2014a) Outlier detection: * Pika et al (2013) * Rogge-Solti and Kasneci (2014)	-
	Unexpected interruptions (3.2.6)	Interruptions in service logs: * Senderovich et al (2014a) Outlier detection: * Pika et al (2013) * Rogge-Solti and Kasneci (2014)	-
Control-flow (3.3)	Control-flow definition (3.3.1)	Control-flow discovery: * van der Aalst (2011) [overview] * De Weerdt et al (2012) [overview]	* Rozinat et al (2008a) [alpha] * Măruşter and van Beest (2009) [heuristics miner] * Rozinat et al (2009) [alpha] * van Beest and Măruşter (2009) [fuzzy mining] * Aguirre et al (2013) [alpha]
	Gateway routing logic (3.3.2)	* Rozinat and van der Aalst (2006a) * Rozinat and van der Aalst (2006b) * de Leoni et al (2013)	* Rozinat et al (2008a) * Rozinat et al (2009) * Pospíšil and Hruška (2012)
Resources (3.4)	Resource roles (3.4.1)	* Song and van der Aalst (2008) * Ferreira and Alves (2012) * Burattin et al (2013)	* Rozinat et al (2009)
	Resource schedule (3.4.2)	* van der Aalst et al (2010) * Wombacher et al (2011) * Liu et al (2012) * Senderovich et al (2014a)	-
	Unavailability handling procedure (3.4.3)	-	-
	Entity handling procedure (3.4.4)	-	-

3.1. Entities

Entities are objects that move through the system and on which activities are executed (Kelton et al 2010). Associated modeling efforts include the definition of the entity attributes, entity types and the entity arrival rate.

3.1.1. Entity attributes (DiscM)

Entities are characterized by attributes, e.g. the number of items in an order, and their entity-specific attribute values (Kelton et al 2010). Modeling attribute value assignment includes a deterministic part, either a constant or an expression conditional on e.g. other entity attributes or the system state. A stochastic element can optionally be added to account for random variation.

To model entity attributes, event log attributes can be used, which can be defined at the event or case level. Consider the running example: when an order is a case, the administrative clerk executing 'Register order' is an event attribute and customer shipping information is a case attribute. When entities and cases correspond, case attributes are direct candidates for entity attributes. However, entities and cases do not have to be defined at the same level of detail by definition. For instance: the BPS model might consider an order as an entity, while a case in the event log is a single order line with an order number as a case attribute. Under these conditions, case attributes need to be aggregated across order lines to obtain entity attributes, i.e. order attributes. Even when entity and case definitions coincide, aggregation is required when event attributes are used as entity attributes. Consider e.g. the summation of attribute values over the events of a particular case. When the appropriate mapping of case and event attributes to entity attributes is determined, logs can be analyzed to assign attribute values to entities.

From the previous, it follows that the ability to retrieve entity attributes from an event log depends on the presence of case and event attributes. Hence, this has to be considered during event log creation, e.g. based on a database as discussed by van der Aalst (2015). However, even when an extensive set of case and event attributes is available, the retrieval of a set of relevant entity attributes, i.e. attributes that influence process execution such as entity routing and activity durations, is far from trivial. Consequently, a method to retrieve relevant attributes from a set of event log attributes is required, which remains unchartered territory in PM. Feature selection techniques from the machine learning field aiming at the selection of a subset of attributes can be a starting point in this respect (Guyon and Elisseeff 2003).

3.1.2. Entity types (PerfED)

Entity types are a set of entity profiles, describing entities with similar attribute values (Kelton et al 2010), e.g. small domestic orders, large domestic orders and international orders. This optional simplification, especially relevant in more complex processes, can be beneficial as one only needs to specify BPS model behavior for each entity type instead of modeling the impact of each attribute on the process. This makes the model's process logic easier to understand and allows performance measures to be expressed on the entity type level. Moreover, the model becomes less sensitive to extreme attribute values of individual entities.

Event logs can be helpful to support the specification of entity types. Even though event logs do not contain direct entity type information, case and event attributes can be helpful as attribute value convergence suggests entity type existence. Other valuable information sources in the log are an entity's trace and possibly its activity durations.

Despite its potential, no research efforts to support entity type modeling using event logs are identified. The principles of trace clustering are a valuable starting point as similar cases need to be grouped. Trace clustering tends to focus on control-flow similarity to create clusters. To this end, each trace is transformed into a vector of control-flow related features, e.g. the directly follows frequency of two particular activities. A clustering algorithm such as hierarchical clustering is applied to these vectors and a separate control-flow model is mined for each cluster (Greco et al 2006; de Medeiros et al 2008; Bose and van der Aalst 2010). Song et al (2009) also include resource-related characteristics in the feature set, which is useful as entity types do not have to differ solely on the control-flow level. Even though the vector-based clustering approach is dominant, other approaches using the generic edit distance to express similarity (Bose and van der Aalst 2009) and Markov model based clusters (Veiga and Ferreira 2011) have been proposed. The aforementioned references cluster traces and afterwards mine a process model for each cluster. Another technique is presented in De Weerdt et al (2013), where traces are grouped in order to maximize the accuracy of the resulting set of process models.

Even though trace clustering provides a useful starting point, two key observations highlight the need for further research. Firstly, the obtained results might support entity type definition at e.g. the control-flow level. Linking these entity types to entity attributes remains a challenge and the modeler might obtain entity types that influence the process flow, but cannot be profiled at the attribute level. This hinders the analysis of simulation results as one cannot describe an entity type and also renders it impossible to assign new entities to the appropriate entity type. Even when all relevant process features and entity attributes are combined in the analysis vector, cluster analysis does not guarantee to result in clusters that differ in terms of both process features and entity attributes. One could e.g. end up with clusters that are different based on entity attributes, but not in terms of control-flow, rendering the entity typology useless for BPS purposes. Secondly, the aforementioned algorithms aim to create distinct process models for each cluster. To use this knowledge in a BPS model, differences between the models for each cluster need to be mapped to a single BPS model.

Besides trace clustering, the notion of process cubes, i.e. structures in which events are organized according to several dimensions (van der Aalst 2013c), can also pose a starting point for entity type modeling on the control-flow level. Dimensions might correspond to distinguishing characteristics of potential entity types. Hence, cube cells can present candidate entity types. Future research could extend these efforts to identify entity types that influence process features in a broader context than only the control-flow.

3.1.3. Entity arrival rate (PerfED)

Accurately modeling entity arrival is crucial given its major influence on process performance, e.g. the average queue length. Entity arrival can be expressed by a constant or context-dependent expression, e.g. the order arrival rate can be higher on weekdays than during the weekend. Optionally, stochasticity can be added by including a probability distribution, e.g. an exponential distribution of interarrival times (IATs) (van der Aalst 2013b).

Research interest on the use of timestamp analysis to support entity arrival rate modeling is limited. A dotted chart, representing the events of all cases by dots (Song and van der Aalst 2007), can provide preliminary insight in the arrival rate. Within the scarce literature on PM in BPS, only Rozinat et al (2009) briefly mention arrival rate modeling. The authors a priori assume an exponential distribution of IATs and focus on the first activity start timestamp to define its parameter.

Both aforementioned approaches implicitly assume a correspondence between the first recorded event and case arrival. However, if the first activity has a non-zero duration and limited resources, queue formation can lead to a deviation between arrival time and the first recorded timestamp. As will be outlined in section 4, queues are often not included in PM research given its predominant focus on independent process instances and not the simultaneous presence of multiple cases in the process. Recently, the authors of this work included the notion of queues when mining the entity arrival rate, using the proportion of entities that queued upon arrival in the event log as a guiding metric. This latter value is approximated by iteratively adjusting the parameters of a probability distribution, which results in an estimate for the distribution of IATs (Martin et al, in press a). The algorithm's sensitivity for the initial parameter values and the event log size is evaluated in Martin et al (in press b). The proposed approach e.g. a priori assumes a gamma distribution of IATs and uses both start and end events, necessitating further research to extend these efforts.

3.2. Activities

Activities provide service to entities (Tumay 1996). Hence, determining which activities should be included in the BPS model is a starting point in activity modeling. Once an activity is defined, several parameters need to be specified, including its duration, resource requirements, queue discipline, queue abandonment condition, interruptibility and unexpected interruptions.

3.2.1. Activity definition (DiscM)

Event logs can be helpful when defining BPS model activities as activities record events in the log. However, assuming a matching log and simulation model granularity can be inappropriate as real-life event logs might register information on a more detailed level (Baier et al 2014). Even in the latter case, log analysis can be useful as activities that are e.g. executed in the same order by similar resources without intermediate queues might be modelled as a single simulation model activity.

The level of detail at which business experts describe activities should also be taken into account during activity specification (Ferreira et al 2013) as deviating definitions might render it complex for experts to e.g. verify the correctness of data analysis results. When the business experts' granularity should be maintained in the BPS model, Szimanski et al (2013) propose a method to link low-level events to these high-level activities using a hierarchical Markov model. The appropriate abstraction level for business experts can also be taken into account during event log creation or, if an event log is given, to alter its granularity. In the latter case, activity mining techniques can be useful as they alter the log's level of detail to create a process model suitable for

business experts (Günther et al 2010), potentially including their knowledge in the process (Baier et al 2014). However, crucial simulation-related notions such as queues should also be taken into account during activity definition, marking opportunities for extending existing methods. Note that the modeler can also build a BPS model on a lower level of detail and aggregate activities solely for discussions with business experts.

3.2.2. Duration (PerfED)

Activity duration reflects its execution time and can be modeled deterministically, either fixed or conditional on entity attributes, resource attributes, queue length or the system state. For instance: the duration of 'Pick order' can be influenced by the number of items in an order, the experience of the order picker, etc. An optional stochastic distribution can be added, using e.g. a beta (van der Aalst 2013b) or gamma (Law 2007) distribution.

Duration observations can be retrieved from an event log, where the observation accuracy depends on the recorded event types. When start and end events are recorded when an activity is executed and interruptibility is disregarded, duration is the difference between the timestamps of both events. When only start events are recorded and no batch processing occurs, the start timestamp of the next processed entity can be a proxy for the associated end timestamp. However, this information is only relevant when queueing entities are present and can be biased when waiting for resource allocation. Alternatively, the start timestamp of the next event in the trace can be analyzed, but the resulting duration can include waiting time for the next activity. No research has been done on this topic.

Conversely, literature does provide suggestions to derive the activity duration when only end events are recorded. Suggested proxies for the associated start time are the end time of the previous activity in a trace and the time at which the resource performing the activity finished its prior task (Nakatumba 2013; Wombacher and Iacob 2013). The suggested approaches can be helpful to obtain rough duration estimates, but will generate a maximal duration value rather than the actual duration. As a consequence, the potential inaccuracy of these estimates should be taken into account.

So far, the discussion focused on how insights in activity duration can be gathered, solely using the event log. However, in PM literature, activity duration is often considered after event log replay (van der Aalst et al 2011) or using the alignment notion (van der Aalst et al 2012b; Rogge-Solti et al 2014), both requiring the presence of a model.

Activity duration is the only activity parameter for which PM is applied in BPS. Rozinat et al (2009) use log replay, assume the presence of both start and end events and hypothesize a normal distribution, using the log to determine parameter values. Future research can extend this approach as it considers activity duration in isolation from e.g. entity and resource attributes and the system state. This will not always hold as e.g. the time required to pack ordered products is not independent of the number of ordered products. Moreover, Rogge-Solti et al (2014) state that log alignment techniques are more robust for event log noise than log replay, but do not explicitly

consider the BPS context. In an online simulation setting, Pospíšil et al (2013) suggest using classification techniques or association rule mining to determine activity duration, but solely focus on case attributes. Hence, these efforts mark a valuable starting point for future work which e.g. also takes the system state into account.

Given the wide range of activity duration determinants (Pospíšil and Hruška 2012), adequately modeling it is complex. For instance: the workload is likely to influence activity duration. To retrieve this relationship from logs, Nakatumba and van der Aalst (2010) and Nakatumba et al (2012) estimate a linear regression for each individual employee. Building on this work, future efforts should (i) fit an alternative model that is more consistent with the hypothesized parabolic relationship between workload and processing speed and (ii) provide opportunities to generalize the results to new employees, which is a common analysis scenario in BPS.

3.2.3. Resource requirements (DiscM)

The resource type and quantity required for activity execution needs to be specified. If event logs contain resource information, as is the case in the example event log in table 1, PM can be useful. Some research efforts highlight the activity-resource relationship when mining resource assignment rules (Ly et al 2006; Liu et al 2008; Huang et al 2011; Senderovich et al 2014a).

Resource assignment rules aim to recommend a single resource for the execution of an activity on a particular entity. To make those efforts applicable in a BPS context, more profound insights are required in resource assignment to an activity when an entity with particular characteristics requests service, potentially taking into account the system state. Moreover, in simulation, resource requirements are often expressed on a resource role level. In that case, the obtained conclusions need to be linked to the allocation of resources to resource roles, as will be discussed in section 3.4.1.

3.2.4. Queue discipline (PerfED)

When the required resources cannot be allocated, entities are placed in a queue. The processing order of queuing entities is determined by the queue discipline. van der Aalst (2013b) marks the limited attention on this matter as a major limitation of contemporary simulation. Classical queue disciplines include first-in-first-out (FIFO), i.e. the first arriving entity is served first, and last-in-first-out (LIFO), i.e. the last arrived entity is processed first. Alternatively, priority rules can be defined to promote entities based on e.g. their attributes, type or the system state (van der Aalst 1998). For instance: express orders might be prioritized in ‘Pick order’, while FIFO holds for other orders.

To support queue discipline modeling, log analysis should identify entities that are in the queue at a particular moment, their characteristics, the system state properties, etc. The observed processing order of entities suggests the queue discipline. Situations in which a queue is empty should be disregarded as arriving entities are processed immediately, independent of their priority profile.

Despite the PM potential, literature does not provide clear starting points on this topic. This can be caused by the tendency of PM algorithms to consider independent process instances and not the interaction between cases simultaneously present in the process, as will be detailed in section 4. Queues in the event log can be detected by e.g. replaying an event log on a CPN model. How this queue is handled should be linked to entity and system state characteristics, as described above. To this end, the notion of Q-logs, introduced in Senderovich et al (2014b), can facilitate the identification of entities residing in the queue at a particular moment as the entrance of an entity in the queue is registered. Senderovich et al (2015) use a Q-log to mine, among others, the evolution in the number of queueing entities. However, requiring the presence of such a log will limit the applicability of the developed techniques. Consequently, future research should also present an algorithm to retrieve queue discipline insights from event logs that do not contain queue-related events. In this respect, recent efforts by Senderovich et al (in press) present a valuable starting point. They approximate queue length when e.g. queue entrance is not logged using either an approach based on K-means clustering and Bayes' theorem or a Markov-chain Monte-Carlo technique.

3.2.5. Queue abandonment condition (PerfED)

Besides the queue discipline, a queue abandonment condition can be specified to express conditions under which entities prematurely leave the queue.

A distinction can be made between reneging and jockeying, respectively expressing entities leaving the queue and the process before being serviced and entities leaving the queue to enter a similar one (Chung 2004). The abandonment condition expresses when such behavior occurs and can be modeled conditionally on e.g. the queue length with an optional stochastic element. Balking, where an entity leaves the system before entering the queue (Chung 2004), is not included as it can be considered as an XOR-gateway, determining whether an entity joins the queue or leaves the system.

As a typical event log only registers events related to activity execution, it is not trivial to determine in which queues a case resided before actual processing. This observation complicates the identification of jockeying behavior. Concerning reneging, detecting incomplete traces can be useful. However, determining the exact time at which these cases terminated to gain insight in e.g. the system state is cumbersome when only start and end events are recorded. Deriving the queue abandonment condition is facilitated when a Q-log is available as it also contains queue entrance and abandonment events (Senderovich et al 2014b). Even when such events are recorded, mining the queue abandonment condition is not trivial as it can depend upon the system state. A starting point for future work is Senderovich et al (2015), where abandonment events are used to model entity patience, which is assumed to follow an exponential distribution. However, hypothesizing the presence of queue-related events is a strong assumption. Consequently, supporting this modeling task in the absence of a Q-log poses a challenging research question.

3.2.6. Interruptibility and unexpected interruptions (PerfED)

When activities might be interrupted during their execution, two parameters need to be defined: the activity's interruptibility and possible unexpected interruptions.

The interruptibility parameter indicates if an activity can be interrupted mid-execution when a time slot is reached where a resource becomes unavailable, e.g. when a break starts while picking an order. Rules can make interruptibility contingent on for instance the system state and entity or resource attributes.

When unforeseen interruptions due to e.g. a packing machine breakdown occur, activity interruptibility is irrelevant. For unexpected interruptions, both the occurrence frequency and duration need to be modeled. Firstly, the occurrence can be modeled e.g. based on a counter expressing the number of entities processed without interruption, with an optional stochastic component to include uncertainty regarding the occurrence of unexpected interruptions. Secondly, the duration of an interruption can be modeled analogously to the activity duration. When a probability distribution is used, a gamma distribution can be considered (Law 2007).

Interruptions during a working day can directly be mined from a service log as e.g. the start of a break is recorded (Senderovich et al 2014a). However, assuming the presence of a service log limits the applicability of the developed techniques. When interruption events are not logged, interruption time is included in the activity duration observations. Hence, outlier analysis, combined with log-based resource schedules, can support the identification of interruptible activities. In contrast, resource schedules will not include unexpected interruptions, leaving activity duration outliers as the only event log information to rely on. Nonetheless, efforts to develop techniques to extract e.g. machine breakdown insights from logs are valuable as their infrequent occurrence makes it difficult to use observations to collect information (Robinson 2004).

Regarding duration outlier analysis in event logs, Pika et al (2013) assume a lognormal distribution of activity duration and define an outlier as a value that is at least two standard deviations higher than the mean. Rogge-Solti and Kasneci (2014) suggest assuming a normal distribution and position the threshold at three standard deviations from the mean.

Despite these efforts, several challenges are present to use outlier analysis to support this modeling task. When the common assumption in outlier detection that start and end timestamps are known is not fulfilled, alternative outlier definitions might be required. Moreover, not every large activity duration observation will be caused by an interruption. Even when outliers can be identified and correctly classified, they need to be related to features that might predict if it relates to a foreseeable or unexpected interruption, e.g. entity attributes or the system state.

3.3. Control-flow

To convert the set of atomic activities, defined using section 3.2, into an executable process model, relationships between activities need to be specified. This involves the addition of sequence flows

and gateways, i.e. a simulation model's control-flow. Besides defining the control-flow, the routing logic also needs to be modeled.

3.3.1. Control-flow definition (DiscM)

By means of sequence flows and gateways, the control-flow defines the routes an entity can follow through the process, e.g. 'Register order' – 'Pick order' – 'Manual packing' – 'Prepare shipping'. The sequential, choice or parallel relationship between activities can be discovered by analyzing the traces in the log. The same holds for the gateway type.

Of all modeling tasks discussed in this paper, PM provides the most extensive support for control-flow discovery. A multitude of algorithms have been developed, for which an overview is presented in e.g. van der Aalst (2011) and De Weerdt et al (2012). These overviews can be extended by including recently developed algorithms such as the inductive miner (Leemans et al 2013).

The alpha-algorithm (Rozinat et al 2008a; Rozinat et al 2009; Aguirre et al 2013), heuristic mining (Măruşter and van Beest 2009) and fuzzy mining (van Beest and Măruşter 2007) have been applied in a BPS-context. These publications consider fairly simple processes, reflecting their proof-of-concept nature. When applied to real-life event logs, the application of these algorithms might lead to incomprehensible process models, requiring the application of e.g. log filtering techniques or the selection of an alternative discovery algorithm (van der Aalst 2011). Consequently, obtaining both a clear and accurate process model from real-life event logs remains challenging.

Control-flow discovery algorithms might not depict gateways with dedicated symbols, but a gateway can be perceived as any splitting point in the model (Rozinat et al 2009). Two general observations regarding gateway discovery need to be taken into account in future work. Firstly, an AND-gateway can be suggested even when not all activity orders are present in the log, implicitly assuming that all interleavings are possible. Secondly, as discovery algorithms tend to be Petri net based, OR-gateways cannot be directly discovered, necessitating further processing.

3.3.2. Gateway routing logic (PerfED)

Routing logic needs to be specified for XOR- and OR-gateways. It consists of rules with a deterministic part, either fixed or conditional on attributes of BPS model components or the system state, and an optional stochastic component to add random variation. In the running example, the choice at the XOR-split is determined by the fragility of the ordered goods.

Event logs can support routing logic modeling by analyzing activity execution circumstances. At a minimum, a log contains frequency information on activity execution. Moreover, event logs often explicitly or implicitly contain information on e.g. entities and past events that allow the discovery of more complex routing models.

Rozinat and van der Aalst (2006a; 2006b) apply a decision tree algorithm to learn the logic in pre-discovered decision points. This approach is also applied in a BPS context (Rozinat et al 2008a;

Rozinat et al 2009). Other related work that acts as a starting point for further research originates from de Leoni et al (2013) which allow for rules that are linear equations of multiple variables and the work of Pospíšil and Hruška (2012) who conceptually suggest adding a stochastic element to such rules in a simulation setting.

The aforementioned work can be extended in two ways. Firstly, non-linear classification rules can also be considered. Secondly, the decision variable scope can be broadened to incorporate e.g. queue length or resource availability instead of only case attributes.

3.4. Resources

Resources execute the activities in the simulation model (Tumay 1996). When resource information is logged, it is trivial to mine a list of resources that are active within the process. This information can be used to support resource modeling tasks, i.e. the definition of resource roles, a resource schedule, the unavailability handling procedure and an entity handling procedure.

3.4.1. Resource roles (DiscM)

Resources performing similar activities can be grouped in a resource role. Resource role specification requires the assignment of both resources and activities to roles. As performed activities typically correspond to organizational functions, resource roles can correspond to e.g. manager and administrative clerk. Consequently, it seems reasonable to assume that a resource belongs to a single resource role and activities can be added to multiple roles with different priorities. Rules can be defined to allow or prohibit deviations from standard priorities. Besides activity priorities, activity permissions need to be specified when a role can only perform an activity under certain conditions, e.g. when the queue length exceeds a threshold.

When event logs contain resource information, it can support resource role identification. To determine activity priorities, an overview of pending role requests and the choices made is required, where only situations where there are several pending requests are relevant. Insights in activity permissions can be gained by analyzing the system state when a resource performs a particular activity.

Song and van der Aalst (2008) use PM to group resources and e.g. use a resource-activity frequency matrix to cluster resources; an approach applied in a BPS context by Rozinat et al (2009). Afterwards, an activity is assigned to a cluster when a cluster's member performs this activity. For simulation purposes, future work can also take e.g. entity attributes into account when defining resource roles, instead of only activity execution frequencies. Moreover, the outlined assignment rule will only lead to satisfactory results when activity division among employees is very rigid. Otherwise, an activity might be linked to multiple roles even when only one of its members performed it once. Future research might take into account e.g. the percentage share of activity execution. Other related work originates from Burattin et al (2013), focusing on handover relationships instead of atomic activities to define roles, and Ferreira and Alves (2012), using hierarchical clustering based on the number of cases resources jointly worked on. However, the

former assumes that an activity can only be assigned to a single role at a particular moment and the latter does not focus on activities. These observations limit their usefulness given the resource role definition of this paper.

3.4.2. Resource schedule (PerfED)

A schedule reflects a resource's presence for the process. When the BPS project studies the addition of resources, two types of schedules need to be specified: one for the current resources such as administrative clerk Sue and a baseline resource role schedule for new resources, e.g. when a new clerk is hired. Deviation rules can be used to model e.g. the occurrence of overtime.

Event logs containing resource information transmit knowledge on a resource's actions in a process. Conversely, schedules provided by e.g. the HR-department might reflect the resource's presence for multiple processes. Moreover, linking the presence of a resource to e.g. the workload can enable the inclusion of schedule deviation rules. Hence, event log analysis can be helpful, but is complex as e.g. periods in which a resource is waiting for entities will not leave a trail in the log.

Mining directly implementable BPS resource schedules resource schedules is an open research question. Related work is limited to mining resource availability, reflecting the fraction of time a resource is executing process activities (van der Aalst et al 2010; Liu et al 2012), and the approximate retrieval of the start and end of a working day (Wombacher et al 2011). Further research is required as schedule definition requires insight in the exact timeframes in which a resource is available, taking into account intermediate interruptions such as a break. As indicated in section 3.2.6, interruptions are logged in service logs (Senderovich et al 2014a), but assuming its presence limits the applicability of developed techniques.

3.4.3. Unavailability handling procedure (PerfED)

The unavailability handling procedure specifies how unavailability periods are dealt with when starting during the execution of a non-interruptible activity, e.g. when a break starts during order packing. It defines what should happen with the elapsed time between the start of the current time slot and the time the resource finishes its activity. The elapsed period can be deduced from the unavailability time or, alternatively, the entire unavailability period can be postponed until the activity is finalized. Note that, in the former case, the resource might not be able to become unavailable. As with schedules, deviation rules and a baseline procedure might be required.

Specifying the unavailability handling procedure is closely related to resource schedule discovery. Consequently, research attention should initially be attributed to the discovery of resource schedules. Afterwards, the length of the identified unavailability periods and the resource's allocation to non-interruptible activities around this period can be investigated. This can provide insight in the applied unavailability handling procedure, which might be conditional on e.g. entity attributes.

3.4.4. Entity handling procedure (PerfED)

The entity handling procedure specifies the number of entities on which a resource performs an activity simultaneously or successively. The importance of taking the tendency to let similar work items accumulate, also known as the batch organization of work, is shown in van der Aalst et al (2010) by analyzing its effect on flow time. As with resource schedules, a baseline procedure and deviation rules might have to be defined. For example: administrative clerk Sue might let orders requiring registration accumulate, unless the backlog for activity 'Pick order' becomes too low.

Modeling the entity handling procedure using event logs requires an analysis of the activities performed by the resource. In case of the simultaneous batch processing, multiple start events for the same activity with an identical or quasi-identical timestamp will be discovered. When sequential batch processing occurs, a particular pattern in queue length evolution should be discovered. Both analyses are complex as the entity handling procedure can depend on the entity type, activity or the system state. Moreover, the absence of start timestamps in real-life logs can be an additional limiting factor. Consequently, given the absence of related work, mining the entity handling procedure is a direction for future research.

4. Discussion

This paper shows that, despite the potential of PM to support BPS model construction, significant research challenges are still ahead to fundamentally integrate both fields. Rozinat et al (2009) provide the most comprehensive support by, firstly, outlining a stepwise method to mine a simulation model from an event log and, secondly, by suggesting suitable plugins within the ProM framework (Verbeek et al 2010) to implement the suggested method. However, simplifying assumptions were made such as using the first activity start timestamp as a proxy for entity arrival and excluding topics such as the queue discipline. The number of required assumptions and their scope will systematically be reduced as more research challenges identified in this paper are tackled. Consequently, event logs will become a more powerful information source to support the construction of a BPS model with its complex internal structure, as outlined in this paper.

Based on an evaluation of the state of the art in literature on this topic, a series of research challenges are identified for all aggregated BPS model building blocks. Key challenges on entity modeling include (i) identifying of relevant entity attributes at an appropriate level of abstraction, (ii) detecting entity types that influence process execution and can be profiled using entity attributes and (iii) retrieving the entity arrival rate when queues are formed. Related to activities, promising directions for future research are (i) defining BPS model activities taking into account simulation-related concepts such as queues, (ii) mining the activity duration given its wide range of determinants and the limited timestamp information that might be available, (iii) specifying resource requirements conditional on e.g. resource attributes and system state variables, (iv) identifying the queue discipline and queue abandonment condition from logs with or without queue-related events and (v) retrieving insights on activity interruptibility and unexpected interruptions using event logs containing or not containing interruption events. Regarding control-

flow modeling, the most extensive PM support is available. However, remaining challenges are (i) discovering a comprehensible process model with appropriate gateways from a real-life event log and (ii) extending the variable scope when specifying the gateway routing logic. With respect to resources, the final aggregated building block, research attention should focus on (i) defining resource roles taking into account e.g. entity attributes and using an appropriate activity assignment rule, (ii) retrieving directly implementable resource schedules and (iii) defining the unavailability and entity handling procedure.

Future research should develop and implement algorithms to tackle the research challenges identified in this paper. The information contained in the event log influences the complexity of retrieving useful insights on a particular BPS modeling task and, hence, the complexity of these algorithms. When, for instance, activity interruptions are logged, determining activity interruptibility is more straightforward than when it is not recorded as, in the latter case, the interruption period is included in the activity duration.

Besides identifying research challenges, starting points for future research originating from PM literature are, if present, also outlined. Despite the steady growth of this field in the last decade, few PM algorithms are directly applicable to support BPS model construction due to differences between the underlying paradigms in PM and simulation. Many PM techniques, such as control-flow discovery algorithms, are designed to gain insights from event logs by focusing on a series of independent cases. Consequently, the interaction between cases that are simultaneously present in the process at a particular moment is often not relevant and, hence, neglected. In contrast, BPS aims to mimic the behavior of an operational business process in which multiple entities tend to be present at the same time. This complicates the use of many existing PM techniques to support BPS model construction, especially for modeling tasks where this interaction influences process behavior such as the queue discipline. This discrepancy between an independent-case paradigm in PM and a correlated-case one in simulation necessitates further efforts to extract BPS-relevant knowledge from event logs. Bridging this gap will require the development of new PM techniques that consider a correlated-case context, in which cases are related due to their simultaneous presence in the system.

5. Conclusion

This paper provided a broad and structured overview on the use of PM to support BPS model construction. Given the potential of PM to improve BPS models and its recognition as a key challenge for PM research in the Process Mining Manifesto (van der Aalst et al 2012), further research is required. Existing research efforts on this topic tend to make simplifying assumptions due to their proof-of-concept nature. Consequently, the literature base needs to be extended to provide PM support that recognizes the complex internal structure of a real-life BPS model.

The observed contrast between the potential of PM in BPS modeling and the state of the art in literature led to the identification of a multitude of research challenges. If present, PM references that can form a starting point for future research are identified. However, the amount of directly

applicable PM algorithms is limited, which can be attributed to differences in the underlying paradigms in both domains.

From the previous, it follows that extensive research is required to fundamentally bridge the gap between PM and BPS. Having a clear understanding of the required BPS modeling efforts and issues that need to be considered is a prerequisite for future work. As a consequence, this paper is a key starting point to structurally integrate PM in simulation model construction.

References

Aguirre S, Parra C, Alvardo J (2013) Combination of process mining and simulation techniques for business process redesign: a methodological approach. Lect Notes Bus Inf 162:24-43. doi: 10.1007/978-3-642-40919-6_2

Baier T, Mendling J, Weske M (2014) Bridging abstraction layers in process mining. Inform Syst 46:123-139. doi:10.1007/978-3-642-40176-3_4

Bose RPJC, van der Aalst WMP (2009) Context aware trace clustering: towards improving process mining results. Proceedings of the Ninth SIAM International Conference on Data Mining 401-412. doi: 10.1137/1.9781611972795.35

Bose RPJC, van der Aalst, WMP (2010) Trace clustering based on conserved patterns: towards achieving better process models. Lect Notes Bus Inf 43:170-181. doi:10.1007/978-3-642-12186-9_16

Burattin A, Sperduti A, Veluscek M (2013) Business models enhancement through discovery of roles. Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining 103-110. doi:10.1109/CIDM.2013.6597224

Chung AC (2004) Simulation modeling handbook: a practical approach. CRC press, Boca Raton

de Leoni M, Dumas M, García-Bañuelos L. (2013) Discovering branching conditions from business process execution logs. Lect Notes Comp Sc 7793:114-129. doi:10.1007/978-3-642-37057-1_9

de Medeiros AKA, Guzzo A, Greco G, van der Aalst WMP, Weijters AJMM, Van Dongen BF, Saccà D (2008) Process mining based on clustering: a quest for precision. Lect Notes Comp Sc 4928:17-29. doi:10.1007/978-3-540-78238-4_4

De Weerdt J, De Backer M, Vanthienen J, Baesens B (2012) A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. Inform Syst 37:654-676. doi:10.1016/j.is.2012.02.004

De Weerdt J, Vanthienen J, Baesens B (2013) Active trace clustering for improved process discovery. IEEE T Knowl Data En 25:2708-2720. doi:10.1109/TKDE.2013.64

Ferreira DR, Alves C (2012) Discovering user communities in large event logs. Lect Notes Bus Inf 99:123-134. doi:10.1007/978-3-642-28108-2_11

Ferreira DR, Szimanski F, Ralha, CG (2013) Mining the low-level behavior of agents in high-level business processes. Int J Business Integration and Management 6:146-166. doi:10.1504/ijbpim.2013.054678

Greco G, Guzzo A, Ponieri L, Sacca D (2006) Discovering expressive process models by clustering log traces. IEEE T Knowl Data En 18:1010-1027. doi:10.1109/TKDE.2006.123

Günther CW, Rozinat A, van der Aalst WMP (2010) Activity mining by global trace segmentation. Lect Notes Bus Inf 43:128-139. doi:10.1007/978-3-642-12186-9_13

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157-1182. doi:10.1162/153244303322753616

Huang Z, Lu X, Duan H (2011) Mining association rules to support resource allocation in business process management. Expert Syst Appl 38:9483-9490. doi:10.1016/j.eswa.2011.01.146

Johnson BT, Eagly AH (2000) Quantitative synthesis of social psychological research. In Reis T, Judd CM (eds) Handbook of research methods in social and personality psychology. Cambridge University Press, Cambridge

Kelton WD, Sadowski P, Swets NB (2010) Simulation with Arena. McGraw-Hill, New York

Law AM (2007) Simulation modeling and analysis. McGraw-Hill, New York

Leemans SJ, Fahland D, van der Aalst WMP (2013) Discovering block-structured process models from event logs-a constructive approach. Lect Notes Comp Sc 7927:311-329. doi:10.1007/978-3-642-38697-8_17

Liu Y, Wang J, Yang Y, Sun J (2008) A semi-automatic approach for workflow staff assignment. Comput Ind 59:463-476. doi:10.1016/j.compind.2007.12.002

Liu Y, Zhang H, Li C, Jiao RJ (2012) Workflow simulation for operational decision support using event graph through process mining. Decis Support Syst 52:685-697. doi:10.1016/j.dss.2011.11.003

Ly TL, Rinderle S, Dadam P, Reichert M (2006) Mining staff assignment rules from event-based data. Lect Notes Comp Sc 3812:177-190. doi:10.1007/11678564_16

Martin N, Depaire B, Caris A (2014a) Event log knowledge as a complementary simulation model construction input. Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications 456-462. doi:10.5220/0005100404560462

Martin N, Depaire B, Caris A (2014b) The use of process mining in a business process simulation context: overview and challenges. Proceedings of the 2014 IEEE Symposium on Computational Intelligence and Data Mining 381-388. doi:10.1109/CIDM.2014.7008693

Martin N, Depaire B, Caris A (in press a) Using event logs to model interarrival times in business process simulation.

Martin N, Depaire B, Caris A (in press b) Using process mining to model interarrival times: investigating the sensitivity of the ARPRA framework.

Măruşter L, van Beest NRTP (2009) Redesigning business processes: a methodology based on simulation and process mining techniques. Knowl Inf Syst 21:267-297. doi:10.1007/s10115-009-0224-0

Melão N, Pidd M (2003) Use of business process simulation: a survey of practitioners. J Oper Res Soc 54:2-10. doi:10.1057/palgrave.jors.2601477

Nakatumba J (2013) Resource-aware business process management: analysis and support. Dissertation, Eindhoven University of Technology.

Nakatumba J, van der Aalst WMP (2010) Analyzing resource behavior using process mining. Lect Notes Bus Inf 43:69-80. doi:10.1007/978-3-642-12186-9_8

Nakatumba J, Westergaard M, van der Aalst WMP (2012) Generating event logs with workload-dependent speeds from simulation models. Lect Notes Bus Inf 112:383-397. doi:10.1007/978-3-642-31069-0_31

Pika A, van der Aalst WMP, Fidge CJ, ter Hofstede AHM, Wynn MT (2013) Predicting deadline transgressions using event logs. Lect Notes Bus Inf 132:211-216. doi:10.1007/978-3-642-36285-9_22

Pospíšil M, Hruška T (2012) Business process simulation for predictions. BUSTECH 2012, The Second International Conference on Business Intelligence and Technology 14-18

Pospíšil M, Mates V, Hruška T, Bartik V (2013) Process mining in a manufacturing company for predictions and planning. International Journal on Advances in Software 6:293-297

Robinson S (2004) Simulation. John Wiley & Sons, Chichester

Rogge-Solti A, Kasneci G (2014) Temporal anomaly detection in business processes. Lect Notes Comp Sc 8659:234-249. doi:10.1007/978-3-319-10172-9_15

Rogge-Solti A, van der Aalst WMP, Weske M (2014) Discovering stochastic Petri nets with arbitrary delay distributions from event logs. Lect Notes Bus Inf 171:15-27. doi: 10.1007/978-3-319-06257-0_2

Rozinat A, Mans RS, Song M, van der Aalst WMP (2008a) Discovering colored Petri nets from event logs. International Journal on Software Tools for Technology Transfer 10:57-74. doi:10.1007/s10009-007-0051-0

Rozinat A, Mans RS, Song M, van der Aalst WMP (2009) Discovering simulation models. Inform Syst 34:305–327. doi:10.1016/j.is.2008.09.002

Rozinat A, van der Aalst WMP (2006a) Decision mining in ProM. Lect Notes Comp Sc 4102:420-425. doi:10.1007/11841760_33

Rozinat A, van der Aalst WMP (2006b) Decision mining in business processes. BPM Center Report BPM-06-10

Rozinat A, Wynn MT, van der Aalst WMP, ter Hofstede A, Fidge CJ (2008b) Workflow simulation for operational decision support using design, historic and state information. Lect Notes Comp Sc 5240:196-211. doi:10.1007/978-3-540-85758-7_16

Senderovich A, Leemans S, Harel S, Gal A, Mandelbaum A, van der Aalst W (in press). Discovering queues from event logs with varying levels of information.

Senderovich A, Weidlich M, Gal A, Mandelbaum A (2014a) Mining resource-scheduling protocols. Lect Notes Comp Sc 8659:200-216. doi:10.1007/978-3-319-10172-9_13

Senderovich A, Weidlich M, Gal A, Mandelbaum A (2014b). Queue mining – predicting delays in service processes. Lect Notes Comp Sc 8484:42-57. doi:10.1007/978-3-319-07881-6_4

Senderovich A, Weidlich M, Gal A, Mandelbaum A (2015) Queue mining for delay prediction in multi-class service processes. Inf Syst 53:278-295. doi:10.1016/j.is.2015.03.010

Song M, Günther CW, van der Aalst WMP (2009) Trace clustering in process mining. Lect Notes Bus Inf 17:109-120. doi:10.1007/978-3-642-00328-8_11

Song M, van der Aalst WMP (2007) Supporting process mining by showing events at a glance. Proceedings of 17th Annual Workshop on Information Technologies and Systems 139-145

Song M, van der Aalst WMP (2008) Towards comprehensive support for organizational mining. Decis Support Syst 46:300-317. doi:10.1016/j.dss.2008.07.002

Szimanski F, Ralha CG, Wagner G, Ferreira DR (2013) Improving business process models with agent-based simulation and process mining. Lect Notes Bus Inf 147:124-138. doi: 10.1007/978-3-642-38484-4_10

Tumay K (1996) Business process simulation. Proceedings of the 1996 Winter Simulation Conference 55-60. doi:10.1109/WSC.1995.478705

Van Beest NRTP, Măruşter L (2007) A process mining approach to redesign business processes – a case study in gas industry. Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing 541-548. doi:10.1109/SYNASC.2007.50

van der Aalst WMP (1998) The application of Petri nets to workflow management. J Circuit Syst Comp 8:21-66. doi:10.1142/S0218126698000043

van der Aalst WMP (2011) Process mining: discovery, conformance and enhancement of business processes. Springer-Verlag, Heidelberg

van der Aalst WMP (2013a) Business process management: a comprehensive survey. ISRN Software Engineering 2013:37p. doi:10.1155/2013/507984

van der Aalst WMP (2013b) Business process simulation survival guide. BPM Center Report BPM-13-11

van der Aalst WMP (2013c) Process cubes: slicing, dicing, rolling up and drilling down event data for process mining. Lect Notes Bus Inf 159:1-22. doi:10.1007/978-3-319-02922-1_1

van der Aalst WMP (2015) Extracting event data from databases to unleash process mining. In vom Brocke J, Schmiedel T (eds) BPM – driving innovation in a digital world. Springer-Verlag, Heidelberg

van der Aalst WMP, Adriansyah A, de Medeiros AKA, Arcieri F, … , Westergaard M, Wynn M (2012a) Process mining manifesto. Lect Notes Bus Inf 99:169-194. doi:10.1007/978-3-642-28108-2_19

van der Aalst WMP, Adriansyah A, van Dongen B (2012b) Replaying history on process models for conformance checking and performance analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2:182-192. doi:10.1002/widm.1045

van der Aalst WMP, Nakatumba J, Rozinat A, Russel N (2010) Business process simulation. In vom Brocke J, Rosemann M (eds) Handbook of business process management. Springer-Verlag, Heidelberg

van der Aalst WMP, Schonenberg MH, Song M (2011) Time prediction based on process mining. Inform Syst 36:450-475. doi:10.1016/j.is.2010.09.001

Veiga GM, Ferreira DR (2010) Understanding spaghetti models with sequence clustering in ProM. Lect Notes Bus Inf 43:92-103. doi: 10.1007/978-3-642-12186-9_10

Verbeek HMW, Buijs JCAM, van Dongen BF, van der Aalst WMP (2010) ProM 6: the process mining toolkit. CEUR Workshop Proceedings 615:34-39.

Wombacher A, Iacob M, Haitsma M (2011) Towards a performance estimate in semi-structured processes. Proceedings of the 2011 IEEE International Conference on Service-Oriented Computing and Applications 1-5. doi:10.1109/soca.2011.6166256

Wombacher A, Iacob ME (2013) Start time and duration estimation in semi-structured processes. Proceedings of the 28th Annual ACM Symposium on Applied Computing 1403-1409. doi:10.1145/2480362.2480626