TOWARD MORE REALISTIC SIMULATION MODELS USING OBJECT-CENTRIC PROCESS MINING

Wil M.P. van der Aalst
Process and Data Science (PADS)
RWTH Aachen University
D-52056 Aachen, Germany
Email: wvdaalst@pads.rwth-aachen.de

KEYWORDS

Process mining; Discrete event simulation; Colored Petri nets; Object-centric process mining

ABSTRACT

Discrete-event simulation has been around for over half a century with applications in production, healthcare, logistics, transportation, etc. However, it is still challenging to create a reliable simulation model that mimics the actual process well and allows for "what-if" questions. Process mining allows for the automated discovery of stochastic process models using event data extracted from information systems. This technology is one of the key enablers for creating digital shadows and digital twins of operational processes. However, traditional process mining focuses on individual cases (e.g., an order, a patient, or a train) with events just referring to a single object (the case). Therefore, the discipline is moving to Object-Centric Process Mining (OCPM), where events can refer to any number of objects. Based on research on OCPM and the prototypes developed, now also commercial software vendors are embracing OCPM, as illustrated by Celonis Process Sphere, which allows for the discovery and analysis of object-centric process models. We believe that OCPM will help to create much more realistic simulation models. Whereas process discovery is backward-looking, with object-centric simulation models, we can also support forward-looking forms of process mining. Although such techniques still need to be developed, they provide a unique opportunity to create more realistic digital twins of organizations and their processes.

INTRODUCTION

Discrete-Event Simulation (DES) is a standard tool in the toolbox of anyone trying to improve operational processes [3], [15], [31], [32]. Although DES software has been around for a long time and is part of the standard curriculum of industrial engineers, its application in practical settings is limited. The reason is that it is time-consuming to create simulation models and it is very difficult to create simulation models that are close to reality, especially when humans are involved. Process mining techniques allow for the automatic generation of simulation models based on event data [2]. However, traditional process-mining techniques adopt a single-case notion and do not support process models involving *different types of objects*, and events involving *multiple objects*. As a result, also these models have problems automatically creating simulation models that correspond to reality. Therefore, we advocate the use of *object-centric process mining* [4], [8], [10] which overcomes some of the limitations of traditional process mining.

We will use the term *simulation* to refer to DES (excluding approaches such as System Dynamics [26], [41]). In the 1960-ties, the first mainstream simulation tools emerged with SIMULA (SIMulation LAnguage) as a notable example [24]. Other early examples of simulation tools include GPSS, SLAM, and SIMAN. Today, there are many mature simulation tools available, e.g., AnyLogic, Simul8, and Arena. Most of the recent examples provide a graphical user interface to drag-and-drop simulation components such as create, queue, machine, conveyor, etc. However, these parameterized components are often tool or application specific. Petri-net-based tools such as *CPN Tools* are less ad-hoc, starting from generic concepts [28]. Colored Petri Nets (CPNs) are an extension of classical Petri nets [25], [34] where tokens can have arbitrarily complex values and are timed. Tokens may get a delay sampled from probability distributions and also transitions may take time. This way it is possible to model and simulate arbitrarily complex discrete-event systems [15]. However, in all of the above cases, it tends to be time-consuming to create simulation models and it is extremely hard to produce models that behave just like the processes and systems observed in reality. For example, the speed of people may depend on their workload (cf. the Yerkes-Dodson Law of Arousal) [35] and people may be involved in multiple processes [1], [14].

Next to mainstream DES approaches, there are also simulation approaches that use a higher-abstraction level. A well-know example is the *System Dynamics* (SD) approach [26], [41] which deliberately ignores the details of a system, such as the properties and behaviors of people, products, or events. The higher abstraction level makes SD suitable for long-term, strategic modeling and simulation. These approaches do not model and simulate individual events stored in information systems. Instead, SD considers quantities, rates, etc. For example, instead of simulating events related to individual customer orders (place order, send invoice, ship, receive payment, etc.) SD simulates aggregated variables related to, for example, the number of new orders placed, the number of orders in the pipeline, the number of resources, etc. Hence, there is a gap between the steps in the simulation model and the actual recorded events that happen in reality. Therefore, we created techniques to automatically convert event data into SD models [38], [36], [37]. These provide an interesting alternative compared to mainstream DES-based approaches. However, the relation between the simulated steps in the SD models and the actual events in the operational processes remains indirect. Therefore, we *focus on simulation at the level of events* in operational processes (as is common in DES).

In the remainder of this keynote paper, we focus on (1) the *interplay* between process mining and simulation, (2) *object-centric* process mining, which allows us to discover models that are not limited to a single-case notion, (3) the challenges related to automatically *creating object-centric simulation models*, and (4) the relation between object-centric simulation models and *digital twins* [5], [7], [13], [17], [30]. The goal is to relate the different topics and introduce novel, innovative process mining techniques for an audience familiar with simulation.

ON THE INTERPLAY BETWEEN PROCESS MINING AND SIMULATION

Figure 1 puts process mining and simulation in the broader perspective of process science and data science.

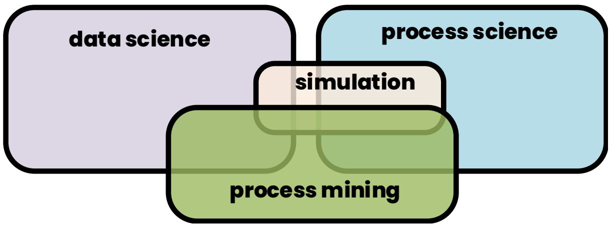

Fig. 1. Relationships between the different disciplines.

Process Mining can be viewed as the *bridge* between process science and data science. In [2] *process science* was defined as "the broader discipline that combines knowledge from information technology and knowledge from management sciences to improve and run operational processes". In [21] it was defined as "the interdisciplinary study of continuous change. By process, we mean a coherent series of changes that unfold over time and occur at multiple levels." The ultimate goal of process science is to improve processes. In [2] *data science* was defined as "an interdisciplinary field aiming to turn data into real value. Data may be structured or unstructured, big or small, static or streaming. Value may be provided in the form of predictions, automated decisions, models learned from data, or any type of data visualization delivering insights. Data science includes data extraction, data preparation, data exploration, data transformation, storage and retrieval, computing infrastructures, various types of mining and learning, presentation of explanations and predictions, and the exploitation of results taking into account ethical, social, legal, and business aspects." Whereas process science tends to start from models, e.g., a Business Process Model and Notation (BPMN) model, data science starts from data instead of models. Due to the many breakthroughs in Machine Learning (ML) in recent years, data science techniques are in focus. However, for process-related topics, modeling still plays a major role.

Process mining starts from *event data* extracted from information systems [12]. In the traditional setting, each event refers to a *case*, an *activity* and a *timestamp*. In process mining terminology, the case is a unique identifier assigned to a specific instance of a process. It is based on the notion of a *single* object type involved in a business process (e.g. sales order item, delivery, etc.) and it groups all events associated with a case to reconstruct a particular process instance for analysis. For example, to create an event log, one needs to decide on which object type to use for grouping events. Once the case notion is fixed is defined, events can be grouped per case. Should an event refer to multiple cases, it needs to be replicated. Based on the grouping, each case refers to sequence of activities executed for that case. The ordering is based on the timestamps of the corresponding events. There may be many other case and event attributes. However, case, activity and timestamp are the mandatory event attributes used for discovering the backbone of process models. This means that any event log can be abstracted into a multiset of traces where each trace is a sequence of activities corresponding to a case. Since different cases may have the same sequence of activities, we need to use multisets.

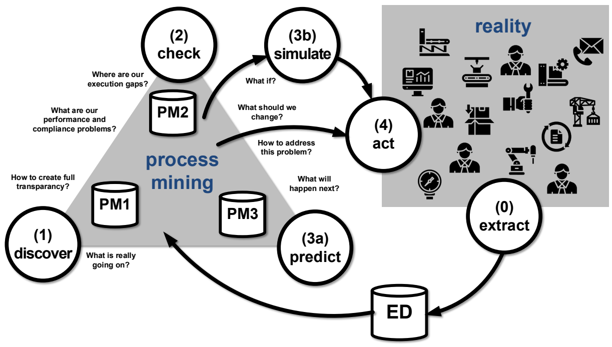

Fig. 2. Overview of the different process mining tasks and the relation to the actual processes and systems.

Figure 2 shows an overview of the main process mining tasks.

- **Extract (0)**: Process mining starts with the extraction of event data (**ED** in Figure 2). This is often a time consuming and iterative process and specific for the selected case notion. There are often multiple systems from different vendors involved (e.g., ERP, CRM and SCM systems), and even within the system of a single vendor, you may find dozens or thousands of different tables.
- **Discover (1)**: Based on the event data, one can discover process models (**PM** in Figure 2). There are many mature process discovery techniques supporting this step [2],

[6], [18], [33], [42], [19]. These may produce Directly-Follows Graphs (DFG) or models that allow for concurrency, e.g., Petri nets, process trees, BPMN, and UML activity diagrams. The discovered process models show what is really going on, thus providing transparency.

- **Check (2)**: For this task we use as input, event data and process models. Discovered process models can be modified from "as is" models into "to be" models. It is also possible to model the expected or normative model from scratch. Event data are replayed on these models. This reveals all discrepancies between data and model [2], [9], [23]. Moreover, process models can be further annotated with frequency and time information. By replaying reality on process models and annotating these, it is possible to identify compliance and performance problems. These are often referred to as "execution gaps" and can be visualized to reveal problems.
- **Predict (3a)**: The tasks just described are *backward-looking*. This is valuable because it exposes opportunities for improvement. However, to realize these opportunities, we need *forward-looking* forms of process mining. One possibility is to use machine-learning techniques to predict performance or compliance problems. The so-called *situation tables* play an essential role in this and provide the interface between process mining and machine learning. A situation table is a two-dimensional table. Each row is an instance. Each column is a feature (also called attribute or variable). There may be a split into a target feature and other features if one wants to use supervised learning. However, situation tables are also used for unsupervised learning (e.g., clustering resources or cases). An instance (i.e., row) could be a case, and the target feature could be overall flow time. Other features could be the resources that worked on the case, the number of deviations, and the total number of cases in the pipeline. An event-based situation table could be used to predict a choice in the process (e.g., a decision to accept or reject). Each time a particular choice needs to be made, a new instance is created, just like for decision-point analysis. A resource-based situation table could show how often resources perform activities and use this to cluster resources and find typical roles. These examples show that process mining *enables* machine learning.
- **Simulate(3b)**: Simulation provides another form of *forward-looking* analysis allowing for "What if?" analysis. In the context of Figure 2, we limit ourselves to simulation models generated by backward-looking process-mining techniques. This requires that, next to control-flow aspects, also organizational, case, and time aspects are added to the discovered control-flow model using attributes from the event log. This is far from trivial. The goal is to first create a simulation model that generates simulation logs close to the event logs of the real process. After this, one can answer "What if?" by changing parts of the model. If it is impossible to mimic the existing processes, then one cannot use simulation to evaluate alternatives.
- **Act (4)**: The ultimate goal of process mining is to improve operational processes. All of the mentioned analysis techniques may provide diagnostics to take action. Ideally, the feedback loop is automated. For example, for known *execution gaps* so-called *action flows* are generated that directly intervene in the running processes (e.g., blocking suppliers, adding resources, rejecting new cases, or triggering stakeholders). Both automatic and human interventions are possible.

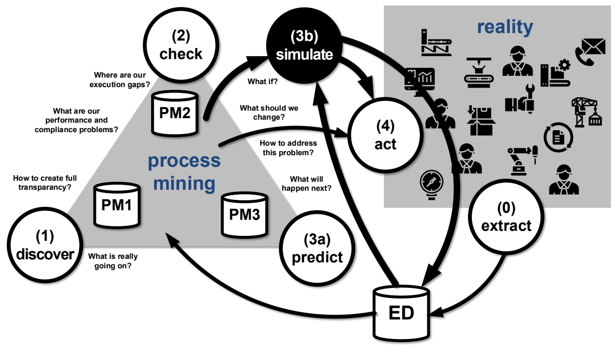

Fig. 3. Overview zooming in on the role of simulation in process mining. Both real events logs and artificial event logs produced in a simulation study can be analyzed in the same way. This provides a unified view. Also simulation models can be discovered automatically using process mining techniques. Finally, the tight coupling between events and the model allows for short-term simulation.

Traditional process mining tools and approaches focus on process discovery and conformance checking, i.e., backward-looking techniques. Hence, it is very natural to combine process mining and simulation [1], [3]. There are four connections that provide synergistic effects (cf. Figure 3).

- **Generating synthetic event logs**: In process mining research, simulation models are often generated to *create event logs* with a know ground truth. In this way simulation is often used to train and evaluate process mining approaches.
- **Generating simulation models**: It is possible to *automatically learn simulation models* using process mining. Several process mining tools have been extended with capabilities to do so [22], [27], [39]. ProM 5 already supported the discovery of simulation models with control-flow, data, resources, and time in 2008 [39]. Today, also commercial tools like Celonis provide means to discover simulation models from event data.
- **Unified views**: Simulation logs can be *analyzed using process mining* as if they are normal event logs. This helps to diagnose simulation results. Most simulation tools focus on aggregate statics and do not provide detailed views like for example dotted charts, process patterns, and alignments. Moreover, it is very powerful to *unify event logs and simulation logs*. Being able to view the real process and the simulated process through the same lens is very advantageous. One can, for example, use *comparative process mining* [11] and directly compare the "as is" process with one of the possible "to be" processes. Because of such a unification, the difference between interpreting simulated behavior and real behavior is fading. The opens up new perspectives.

- **Short-term simulation**: If it is possible to continuously extract event data from information systems, then it is also possible to reconstruct the *current state* of the process at any point in time. This means that *short-term simulation* comes into reach. The key idea of short-term simulation is to start all simulation runs from the current state (which is known precisely) and focus on the *transient* short-term behavior [3], [40]. For transient analysis, the focus is on the initial part of future behavior, i.e., starting from the initial state, the "near future" is explored. While for steady-state analysis the initial state is irrelevant and the simulation can be started without any cases in progress, this type of simulation relies on state information and a tight coupling between the real event data and the simulation model. In [40], we demonstrated that this is possible using a process mining tool (ProM) and a workflow management system (YAWL). The result is a "fast forward button" to explore the future of a process [3].

The above shows shows that many interesting crossovers between process mining and simulation are possible. However, this stand or falls with the ability of models adequately capturing reality. This is where object-centricity comes in.

OBJECT-CENTRIC PROCESS MINING

Combining process mining and simulation provides powerful new analysis approaches using a mix of backward-looking and forward-looking techniques. Normally, process mining does not allow for "What if?" questions. However, simulation only makes sense if the simulation model indeed adequately describes reality. Anyone that has applied simulation in real-life settings knows this is extremely difficult. One of the reasons is that processes simulated in isolation often depend on other processes (e.g., because resources are shared). A full-time employee spending only 10% of her time on a particular process may still cause bottlenecks in this process due to external factors [1]. When using traditional process mining to discover simulation models, one needs to pick a particular case notion. Traditional process mining only allows you to view a process from the perspective of a single object (e.g., sales order or invoice), rather than the relationship between all objects involved. The focus is on a single object, and the model describes the lifecycle of one instance of that object, "a case", in isolation. Picking a case identifier corresponds to projecting reality onto a single object type for analysis (e.g., orders, customers, items, suppliers, deliveries, etc.). *Object-Centric Process Mining* (OCPM) aims to address this problem by extending the notion of an event to overcome some of the limitations of traditional process mining [4], [8], [10]. The core idea is that an event can refer to *any number of objects* and there is no longer the need to pick a case notion.

Object-Centric Event Data (OCED) connect events and objects. Each *object* has precisely one *object type*, but many objects may have the same type. Example object types are product, container, patient, customer, supplier, machine, order, treatment, claim, payment, complaint, request, etc. Objects are instances of these types. For example, a particular container or a particular supplier. Each *event* has an *event type*, also called *activity*. Many events can have the same type, but each event has precisely one type. Often the terms event type and activity are used interchangeably. Example event types are load container, make decision, record payment, store item, etc. Events are instances of these types. We assume that events are *atomic*. Therefore, each event has precisely one timestamp. An event may refer to *any number of objects*. There is a qualified many-to-many relationship between events and objects. In traditional process mining, there would be just one object type *case* and each event would refer to precisely one object of that type (i.e., a case). The so-called *Event-to-Object* (E2O) relations generalize this to a qualified many-to-many relationship. Objects can be related using *Object-to-Object* (O2O) relationships. O2O relations are *static* and the E2O relations are *dynamic*. Both objects and events can have any number of *attributes* with corresponding *values*. For objects, these values may change over time.

After explaining OCED it is easier to explain the challenges of traditional process model using a single-case notion. Assume that we take OCED as input and try to apply a traditional process mining tool. Because we need to pick one case identifier for each event (rather than any number of objects), we need to transform the data using the followings steps:

- Pick an *object type* to serve as the *case notion*.
- Remove all *objects* of a *different type*. The remaining objects are called *cases*.
- Only keep *object attribute values* corresponding to these cases, and, if there are multiple case attribute values for a case and case attribute combination, then keep only the last one.
- Remove all *events* that do not have an O2E relation to at least one case. The remaining events refer to one or more cases.
- If an event refers to multiple cases, then *replicate the event once for each case*. By replicating events, we can relate each event to a single case.

The resulting event data are called *flattened event data* and can be loaded into any process mining tool. Obviously, events in the original event log that have no corresponding events in the flattened event log disappear from the input data. This is called *deficiency*. More interesting are the *convergence* and *divergence* problems. Events referring to multiple objects of the selected type are replicated, possibly leading to *unintentional duplication*. The replication of events can lead to misleading diagnostics and is called convergence. There may also be multiple events that refer to the same case and activity differing with respect to one of the *not selected* object types. These events are referring to different objects of a type not selected as the case notion and thus become indistinguishable looking only at the case and activity (i.e., event type). This is the divergence problem. Next to the problem of concurrency, this is one of the reasons why discovered process models (especially DFGs) are often Spaghetti-like.

Fortunately, it is possible to modify many existing case process mining techniques to handle OCED [4], [8], [10]. How these techniques work precisely is out of the scope of this paper. Instead, we refer to [10]. The general idea is to use the flattened event data to create intermediate process models and then merge and correct them to make sure that the right frequencies and times are shown.

There are over 40 process mining vendors, see for example the listing on www.processmining.org. Process mining tools form a new category of software, as is reflected by the corresponding Gartner Magic Quadrant [29]. All tools support the discovery of Directly-Follows Graphs (DFGs) with frequencies and times, and most of them (but not all) support some form of conformance checking and BPMN visualization. However, very few support Object-Centric Process Mining (OCPM). Celonis was the first commercial vendor fully supporting OCPM with the release of *Process Sphere* in 2022 [8]. Before, there were several non-commercial open-source tools supporting object-centricity, e.g., the "OCELStandard" package in ProM (promtools.org), the OC-PM tool (ocpm.info) [20], and Object-Centric Process Insights (ocpi.ai) [16]. A detailed explanation of these tools and the algorithms used is outside the scope of this paper. However, we show a few screenshots of Celonis Process Sphere while analyzing a data set with six object types and 16 activities (event types). Figures 4, 5, and 6 provide an impression of the advantages of using multiple objects.

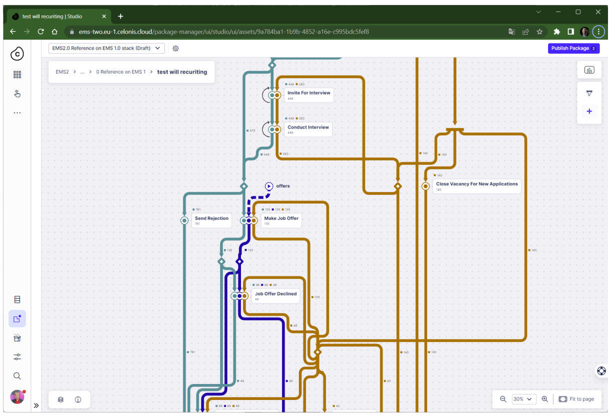

Fig. 4. Screenshot of Celonis Process Sphere while analyzing the process of handling applications. The different colors refer to different object types.

In this small data set there are 288 applicants, 916 applications, 135 offers, 140 vacancies, 20 recruiters, and 6 managers. Figure 5 shows that applications, offers and vacancies are selected. It is possible to select any combination of object types and event types. This means that it is possible to view the process from any angle. It is also possible to select objects having specific properties, e.g., rejected applications or vacancies that could not be filled.

Figure 6 shows an analysis of the flow time. It is possible to select any pair of activities and an object type, and then analyze how many objects moved from the first to the second activity and how long this took. In the selected view, we analyze the flow time from activity "Submit Application" to activity "Make Job Offer" with respect to object type "applications". Process Sphere shows that of the 916 applications, 135 got an offer, and the average time was 35 days.

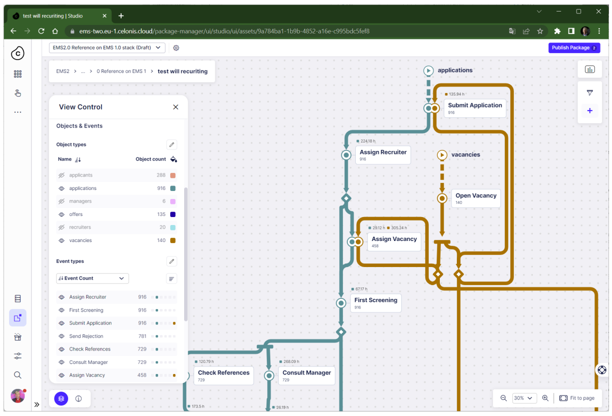

Fig. 5. It is possible to select both object types and activities (event types). This defines the scope of analysis.

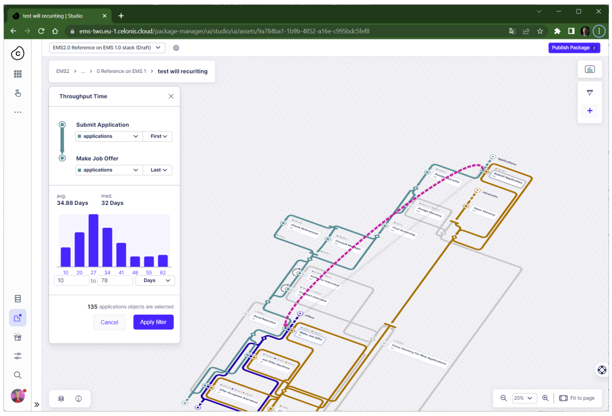

Fig. 6. It is possible to select any pair of activities and an object type, and perform a detailed analysis of the flow of objects between these two activities.

Note that one applicant may apply for many vacancies, for one vacancy there may be many applications, for one vacancy there may be multiple offers in case applicants decline, etc. Also one can view the process from the viewpoint of the applicants, applications, offers, vacancies, recruiters, and managers. Depending on the selected view, the process may look completely different. Using OCPM, one can view the process from multiple angles without distortions [4], [8], [10].

OBJECT-CENTRIC SIMULATION MODELS

Traditional process mining can be used to automatically generate simulation models. However, (1) extraction is time consuming and needs to be repeated when new questions emerge, (2) interactions between objects are not captured and objects are analyzed in isolation, and (3) a three-dimensional reality with multiple object types needs to be squeezed into two-dimensional event logs and models focusing on individual cases [4], [8], [10]. OCPM addresses these problems. For example, event data are extracted only once and interactions between objects are captured and analyzed. Hence, traditional 2D process mining is like taking an X-ray, and OCPM is like taking an MRI creating a three-dimensional image that can be viewed from any angle. However, thus far, there are no techniques to automatically discover *object-centric* *simulation models*. This is an important direction of future research because faithful simulation models need to be based on data and cover multiple objects. Many performance-related problems involve multiple types of objects. Note that resources like people and machines are also objects and this is typically covered well in existing simulation tools. However, these tools also tend to focus on one type of cases competing for shared resources and the object-centricity in OCPM goes far beyond this. Consider, for example, an organization that has procurement problems leading to smaller batch sizes in production. The smaller batch sizes lead to multiple shipments for a single sales order. Imagine a customer that orders 50 items. Instead of receiving all 50 items in one shipment, the 50 items end up in three separate shipments leading to extra costs and emissions. However, the root cause of the problem is not in transportation or production. Using OCPM, one can see the ripple effect of procurement problems leading to higher shipping costs. Actually, there is untapped improvement potential in most situations where objects (orders, items, suppliers, machines, customers, shipments, etc.) depend on each other. Obviously, *simulation models need to incorporate these dependencies to describe reality faithfully*.

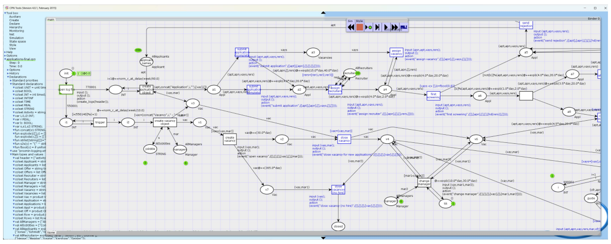

Fig. 7. A CPN Tools simulation model for the process involving applicants, applications, offers, vacancies, recruiters, and managers.

To understand the challenge of creating object-centric simulation models, let us take a look at two existing process-model notations.

We start with *Colored Petri Nets* (CPNs) as a process-model notation [15], [28]. Figure 7 shows a screenshot of *CPN Tools* (cpntools.org). The CPN models the hiring process involving six types of objects (applicants, applications, offers, vacancies, recruiters, and managers) mentioned before. The figure is not intended to be readable, but it gives an impression of CPN Tools. In a CPN, places are typed (called color sets). One can have any number of types. Tokens in a place should have a value (i.e., color) of the corresponding type. CPNs can be hierarchical and timed. Transitions can have guards and arcs have inscriptions. An arc inscription evaluates to a multiset of token values or a single value. These inscriptions are related through shared free variables. This is used to determine whether a transition is enabled and what tokens should be consumed and produced. This way, one can describe arbitrary complex systems. Process mining tools like ProM can generate CPNs based on event data [39]. However, the class of CPNs generated is very restricted using a single-case notion. There is no tool or approach to generate arbitrary CPNs from event data. This is also impossible, because CPNs are as powerful as a programming language. It is simply infeasible to discover an arbitrary CPN from an (object-centric) event log. Instead, one needs to use a restricted representational bias to be able to discover CPNs, i.e., the search space needs to be limited to a subclass of CPNs.

The second process-model notation is more restricted and tailored toward process discovery. *Object-Centric Petri Nets* (OCPNs), as defined in [10], extend labeled Petri nets with place typing and variable arcs. Places types correspond to the object types in OCED and tokens correspond to objects in OCED. Transition labels in the OCPN correspond to the event types (i.e., activities) in OCED and transition occurrences correspond to events in OCED. Variable arcs are used to specify that multiple objects of the same type can be involved in a transition occurrence. Arcs can also be equipped with cardinality expressions (e.g., at least one or at most one object of the given type). Note that the behavior of an OCPN is deliberately "underspecified" to allow for process discovery and conformance checking. There are only typing and cardinality constraints. Hence, objects of different types are unrelated from the OCPN point of view. Of course, they are related in the data. In OCED, objects are related through O2O relations and pairs of O2E relations (i.e., two objects involved in the same event). As a result, the discovered OCPNs allow for too much behavior, i.e., they are underfitting. Therefore, it is impossible to directly simulate OCPNs without further bounding behavior.

In the traditional setting, it is often easy to model the arrival process of new cases. One can, for example, assume that interarrival times are sampled from a negative-exponential distribution (i.e., a Poisson process) and that the parameter λ depends on the time of the day. For object-centric simulation models this is much more difficult because there are different types of objects and these are related. How they are related also influences the dynamic behavior. Consider for example orders consisting of multiple items. For the arrival process of new objects one cannot randomly create orders and objects independent of one another. Also the dynamic behavior depends on this. The Petri net needs to know non-local dependencies and these need to be learned from the example data. More research is needed to solve this satisfactorily. Thus far only heuristics have been developed.

ENABLING DIGITAL TWINS

The notion of a *digital twin* has been around for quite a while. Originally, the term was used for a *virtual* model designed to accurately reflect a *physical* object (e.g., an engine or wing). However, since Gartner introduced the term "Digital Twin of an Organization" (DTO), it is also used for digital models of organizations and their processes [30]. Obviously, object-centric simulation models can be seen as digital twins.

The author prefers to distinguish between a digital model, a digital shadow, and a digital twin as illustrated using Figure 8 [5], [13]. A *digital model* is a reflection of reality that is created manually and functions in an offline manner, i.e., the model does not change when reality changes. A *digital shadow* goes one step further. The model is now automatically

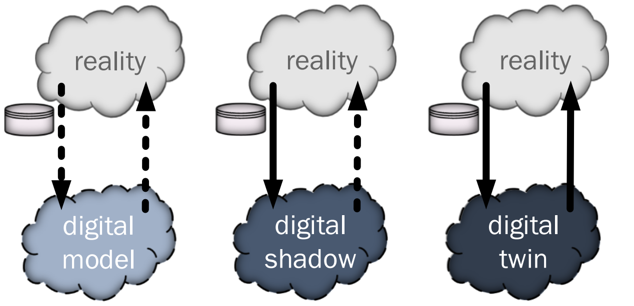

Fig. 8. The difference between a digital model, a digital shadow, and a digital twin. The solid arcs are realized using hardware and software in a real-time setting. The dashed arcs are translations done by humans offline.

derived and changes when reality changes. The digital shadow can also be used to reason about reality and answer what-if questions (e.g., using simulation). Although the digital shadow is based on data tapped from reality, there is no automated real-time feedback loop [13]. A *digital twin* also includes an automated real-time feedback loop. Results produced by the digital twin may directly impact reality. For example, when the simulation model predicts a delay, the production process is reconfigured automatically. Currently, digital twins only exist in very specific settings and the term is often misused when referring to a digital model. Most process mining applications result in digital shadows instead of digital twins. Therefore, we advocate the use of *Hybrid Intelligence* (HI), combining *human intelligence* (flexible, creative, emphatic, instinctive, and commonsensical) and *machine intelligence* (fast, efficient, cheap, scalable, and consistent) [13].

Object-Centric Process Mining (OCPM) is an important enabler in improving the quality of digital shadows and object-centric simulation models could help to lift these digital shadows to digital twins [5], [7].

The development toward digital twins for organizations and their processes will be a gradual development that can be compared to the development of autonomous driving. In [7], the author defined "Six Levels of Autonomous Process Execution Management" inspired by the six levels defined by the Society of Automotive Engineers (SAE). SAE identified levels ranging from no driving automation (Level 0) to full driving automation (Level 5). In 2022, Mercedes-Benz was the world's first automaker to obtain international approval for a car operating at Level 3. This is still under very specific conditions: during daytime, on highways, and at speeds below 60 kilometers per hour. This shows the gradual development of the field. Table I shows the SAE levels side-by-side with the six levels of autonomous process execution management defined in [7]. In the table, the term Process Execution Management System (PEMS) is used as an umbrella term for different capabilities to support processes. Obviously, these levels are related to the notions of digital shadows and digital twins.

Table I and Figure 8 are very high-level and visionary. However, it is clear that such visions can only be realized using data-driven techniques that consider multiple object types and their dependencies. Therefore, object-centric process mining will need to play a key role.

CONCLUSION

This paper discussed the relationship between simulation and process mining. It is vital to extend process mining with forward-looking techniques like simulation. However, this requires more realistic process models. Process mining and most business process modeling notation focus on one object type and describe the handling of cases in isolation. Object-centric process mining will help to create process models incorporating multiple objects of different types and their interactions. Most performance problems involve multiple objects that live at intersections of organizational units. Therefore, we need object-centric simulation models to answer "What if?" questions in these more complex settings. Simulation languages like CPN Tools allow for the modeling of such processes. However, these and other simulation models still need to be created by hand. This is time-consuming and error-prone. Therefore, we need more systematic ways to create object-centric simulation models from event data.

ACKNOWLEDGEMENTS

The author thanks the Alexander von Humboldt (AvH) Stiftung for supporting his research. The research is also funded by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy, Internet of Production (390621612).

REFERENCES

[1] W.M.P. van der Aalst. Business Process Simulation Survival Guide. In J. vom Brocke and M. Rosemann, editors, *Handbook on Business Process Management 1*, International Handbooks on Information Systems, pages 337–370. Springer-Verlag, Berlin, 2015.

[2] W.M.P. van der Aalst. *Process Mining: Data Science in Action*. Springer-Verlag, Berlin, 2016.

[3] W.M.P. van der Aalst. Process Mining and Simulation: A Match Made in Heaven! In A. D'Ambrogio and G. Zacharewicz, editors, *Computer Simulation Conference (SummerSim 2018)*, pages 1–12. ACM Press, 2018.

[4] W.M.P. van der Aalst. Object-Centric Process Mining: Dealing With Divergence and Convergence in Event Data. In P.C. Ölveczky and G. Salaün, editors, *Software Engineering and Formal Methods (SEFM 2019)*, volume 11724 of *Lecture Notes in Computer Science*, pages 3–25. Springer-Verlag, Berlin, 2019.

[5] W.M.P. van der Aalst. Concurrency and Objects Matter! Disentangling the Fabric of Real Operational Processes to Create Digital Twins. In A. Cerone and P. Olveczky, editors, *International Colloquium on Theoretical Aspects of Computing (ICTAC 2021)*, volume 12819 of *Lecture Notes in Computer Science*, pages 3–17. Springer-Verlag, Berlin, 2021.

[6] W.M.P. van der Aalst. Foundations of Process Discovery. In W.M.P. van der Aalst and J. Carmona, editors, *Process Mining Handbook*, volume 448 of *Lecture Notes in Business Information Processing*, pages 37–75. Springer-Verlag, Berlin, 2022.

[7] W.M.P. van der Aalst. Six Levels of Autonomous Process Execution Management (APEM). *Computing Research Repository (CoRR) in arXiv*, abs/2204.11328, 2022.

[8] W.M.P. van der Aalst. Object-Centric Process Mining: The Next Frontier in Business Performance. celon.is/OCPM-Whitepaper, 2023.

[9] W.M.P. van der Aalst, A. Adriansyah, and B. van Dongen. Replaying History on Process Models for Conformance Checking and Performance Analysis. *WIREs Data Mining and Knowledge Discovery*, 2(2):182–192, 2012.

[10] W.M.P. van der Aalst and A. Berti. Discovering Object-Centric Petri Nets. *Fundamenta Informaticae*, 175(1-4):1–40, 2020.

	SAE levels for autonomous driving	Levels of autonomous process execution management
Level 0	A human is driving, and features are limited to breaking assistance, blind-spot warning, lane departure warning, etc.	There is no PEMS. All orchestration and management are done by humans. Features are limited to dashboards, reporting, key performance indicators, hard-coded workflows, and manually created simulations to conduct what-if analysis.
Level 1	A human is driving, but the car provides steering or brake/ acceleration support, e.g., lane centering or adaptive cruise control.	The PEMS is able to detect and quantify known and unknown performance and compliance problems. Features include process discovery and conformance checking. The PEMS may create alerts. However, humans need to interpret the diagnostics and, if needed, select appropriate actions.
Level 2	A human is driving, but the car provides steering and brake/ acceleration support. The difference with Level 1 is the combination of systems.	The PEMS is able to detect and quantify known and unknown performance and compliance problems. Moreover, the PEMS is able to recommend actions in case of detected known performance and compliance problems (execution gaps) and support the user in triggering corresponding actions. These actions may be automated, but in-the-end a human decides.
Level 3	Under selected circumstances, the car is driving. However, the driver needs to be alert and ready to take over control at any time.	The PEMS automatically responds to performance and compliance problems by taking appropriate actions. However, this is limited to a subset of problems and humans need to be alert and ready to take over control.
Level 4	Under selected circumstances, the car is driving. If the conditions are not met, the vehicle stops. The driver does not need to constantly monitor the situation.	The PEMS automatically responds to performance and compliance problems by taking appropriate actions. In principle, all management and orchestration decisions are made by the PEMS. Humans do not need to constantly monitor the PEMS, but the system may decide to call on the help of humans in case of diverging or unexpected behaviors.
Level 5	The car can drive itself under all circumstances (comparable to a human driver).	The PEMS functions fully autonomous also under diverging or unexpected circumstances.

TABLE I
THE SIX LEVELS OF AUTONOMOUS DRIVING DEFINED BY THE SOCIETY OF AUTOMOTIVE ENGINEERS (SAE) COMPARED WITH PROPOSED AUTONOMY LEVELS FOR PROCESS EXECUTION MANAGEMENT SYSTEMS (PEMS).

[11] W.M.P. van der Aalst, T. Brockhoff, A. Farhang, M. Pourbafrani, M.S. Uysal, and S. J. van Zelst. Removing Operational Friction Using Process Mining: Challenges Provided by the Internet of Production (IoP). In S. Hammoudi and C. Quix, editors, *Data Management Technologies and Applications*, volume 1446 of *Communications in Computer and Information Science*, pages 1–31. Springer-Verlag, Berlin, 2021.

[12] W.M.P. van der Aalst and J. Carmona, editors. *Process Mining Handbook*, volume 448 of *Lecture Notes in Business Information Processing*. Springer-Verlag, Berlin, 2022.

[13] W.M.P. van der Aalst, O. Hinz, and C. Weinhardt. Resilient Digital Twins. *Business and Information Systems Engineering*, 63(6):615–619, 2021.

[14] W.M.P. van der Aalst, J. Nakatumba, A. Rozinat, and N. Russell. Business Process Simulation: How to Get it Right? In J. vom Brocke and M. Rosemann, editors, *Handbook on Business Process Management*, International Handbooks on Information Systems, pages 313–338. Springer-Verlag, Berlin, 2010.

[15] W.M.P. van der Aalst and C. Stahl. *Modeling Business Processes: A Petri Net Oriented Approach*. MIT Press, Cambridge, MA, 2011.

[16] J.N. Adams and W.M.P. van der Aalst. OCpi: Object-Centric Process Insights. In L. Bernardinello and L. Petrucci, editors, *Application and Theory of Petri Nets and Concurrency (Petri Nets 2022)*, volume 13288 of *Lecture Notes in Computer Science*, pages 139–150, 2022.

[17] K. Agalianos, S.T. Ponis, E. Aretoulaki, G. Plakas, and O. Efthymiou. Discrete Event Simulation and Digital Twins: Review and Challenges for Logistics. *Procedia Manufacturing*, 51:1636–1641, 2020. 30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021).

[18] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, F.M. Maggi, A. Marrella, M. Mecella, and A. Soo. Automated Discovery of Process Models from Event Logs: Review and Benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 31(4):686–705, 2019.

[19] A. Augusto, R. Conforti, M. Marlon, M. La Rosa, and A. Polyvyanyy. Split Miner: Automated Discovery of Accurate and Simple Business Process Models from Event Logs. *Knowledge Information Systems*, 59(2):251–284, 2019.

[20] A. Berti and W.M.P. van der Aalst. OC-PM: Analyzing Object-Centric Event Logs and Process Models. *Computing Research Repository (CoRR) in arXiv*, abs/2209.09725, 2022.

[21] J. vom Brocke, W.M.P van der Aalst, T. Grisold, W. Kremser, J. Mendling, B. Pentland, J. Recker, M. Roeglinger, M. Rosemann, and B. Weber. Process Science: The Interdisciplinary Study of Continuous Change. Available via SSRN: http://ssrn.com/abstract=3916817, 2021.

[22] M. Camargo, M. Duma, and O. González-Rojas. Automated Discovery of Business Process Simulation Models From Event Logs. *Decision Support Systems*, 134:113284, 2020.

[23] J. Carmona, B. van Dongen, A. Solti, and M. Weidlich. *Conformance Checking: Relating Processes and Models*. Springer-Verlag, Berlin, 2018.

[24] O.J. Dahl and K. Nygaard. SIMULA: An ALGOL Based Simulation Language. *Communications of the ACM*, 1:671–678, Sept 1966.

[25] J. Desel and J. Esparza. *Free Choice Petri Nets*, volume 40 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 1995.

[26] J.W. Forrester. *System Dynamics and the Lessons of 35 Years*, pages 199–240. Springer-Verlag, Berlin, 1993.

[27] F. Gottschalk, W.M.P. van der Aalst, M.H. Jansen-Vullers, and H.M.W. Verbeek. Protos2CPN: Using Colored Petri Nets for Configuring and Testing Business Processes. *International Journal on Software Tools for Technology Transfer*, 10(1):95–111, 2008.

[28] K. Jensen. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*. EATCS monographs on Theoretical Computer Science. Springer-Verlag, Berlin, 1992.

[29] M. Kerremans, K. Iijima, A. Sachelarescu, N. Duffy, and D. Sugden. Magic Quadrant for Process Mining Tools, Gartner Research Note GG00774746. www.gartner.com, 2013.

[30] M. Kerremans and J. Kopcho. Create a Digital Twin of Your Organization to Optimize Your Digital Transformation Program, Research Note G00379226. www.gartner.com, 2019.

[31] J. Kleijnen and W. van Groenendaal. *Simulation: A Statistical Perspective*. John Wiley and Sons, New York, 1992.

[32] A.M. Law and D.W. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, 1982.

[33] S.J.J. Leemans, D. Fahland, and W.M.P. van der Aalst. Scalable Process Discovery and Conformance Checking. *Software and Systems Modeling*, 17(2):599–631, 2018.

[34] T. Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.

[35] J. Nakatumba and W.M.P. van der Aalst. Analyzing Resource Behavior Using Process Mining. In S. Rinderle-Ma, S. Sadiq, and F. Leymann, editors, *BPM 2009 Workshops, Proceedings of the Fifth Workshop on Business Process Intelligence (BPI'09)*, volume 43 of *Lecture Notes in Business Information Processing*, pages 69–80. Springer-Verlag, Berlin, 2010.

[36] M. Pourbafrani and W.M.P. van der Aalst. Extracting Process Features from Event Logs to Learn Coarse-Grained Simulation Models. In M. La Rosa, S. Sadiq, and E. Teniente, editors, *International Conference on Advanced Information Systems Engineering (CAiSE 2021)*, volume

12751 of *Lecture Notes in Computer Science*, pages 125–140. Springer-Verlag, Berlin, 2021.

[37] M. Pourbafrani and W.M.P. van der Aalst. Discovering System Dynamics Simulation Models Using Process Mining. *IEEE Access*, 10:78527–78547, 2022.

[38] M. Pourbafrani, S.J. van Zelst, and W.M.P. van der Aalst. Supporting Automatic System Dynamics Model Generation for Simulation in the Context of Process Mining. In W. Abramowicz and G. Klein, editors, *International Conference on Business Information Systems (BIS 2020)*, volume 389 of *Lecture Notes in Business Information Processing*, pages 249–263. Springer-Verlag, Berlin, 2020.

[39] A. Rozinat, R.S. Mans, M. Song, and W.M.P. van der Aalst. Discovering Simulation Models. *Information Systems*, 34(3):305–327, 2009.

[40] A. Rozinat, M. Wynn, W.M.P. van der Aalst, A.H.M. ter Hofstede, and C. Fidge. Workflow Simulation for Operational Decision Support. *Data and Knowledge Engineering*, 68(9):834–850, 2009.

[41] J. Sterman. System Dynamics: Systems Thinking and Modeling for a Complex World. Technical report, Massachusetts Institute of Technology, 2002. ESD-WP-2003-01.13.

[42] J.M.E.M. van der Werf, B.F. van Dongen, C.A.J. Hurkens, and A. Serebrenik. Process Discovery using Integer Linear Programming. *Fundamenta Informaticae*, 94:387–412, 2010.

Wil van der Aalst is a full professor at RWTH Aachen University, leading the Process and Data Science (PADS) group. He is also the Chief Scientist at Celonis, part-time affiliated with the Fraunhofer FIT, and a member of the Board of Governors of Tilburg University. His research interests include process mining, Petri nets, business process management, workflow management, process modeling, and process analysis. Wil van der Aalst has published over 900 articles and books and is typically considered to be in the top-15 of most cited computer scientists with an H-index of 173 and more than 133.000 citations. Van der Aalst is an IFIP Fellow, IEEE Fellow, ACM Fellow, and received honorary degrees from the Moscow Higher School of Economics (Prof. h.c.), Tsinghua University, and Hasselt University (Dr. h.c.). He is also an elected member of the Royal Netherlands Academy of Arts and Sciences, the Royal Holland Society of Sciences and Humanities, the Academy of Europe, the North Rhine-Westphalian Academy of Sciences, Humanities and the Arts, and the German Academy of Science and Engineering. In 2018, he was awarded an Alexander-von-Humboldt Professorship.