Object-Centric Simulation: Learning the True Fabric of Operational Processes

Wil M.P. van der Aalst[0000−0002−0955−6940]

If you cannot discover it, it does not exist.

Abstract Simulation is a well-established discipline, but simulation models have traditionally been constructed manually and therefore rely heavily on expert assumptions. The growing availability of event data, combined with advances in process mining and machine learning, now makes it possible to learn simulation models directly from observed operational behavior. For case-centric processes, event logs can already be used to discover simulation models that complement process mining and answer "What if?" questions. Yet operational reality is rarely case-centric. Customers, orders, items, packages, deliveries, resources, and other objects evolve together, interact through shared events, and continuously change the network that connects them. Capturing the true fabric of such operational processes requires moving from case-centric to *object-centric simulation*, building on Object-Centric Process Mining (OCPM). This transition is fundamental rather than incremental. A faithful object-centric simulation must capture not only behavior, timing, and resource usage, but also evolving object populations, changing relations, variable cardinalities, synchronization, splits, merges, and dependencies between interacting life-cycles. Reconstructing the complete generative mechanism underlying such behavior may therefore be unrealistic – and may not even be necessary. The goal is not to reproduce the past, but to generate and evaluate plausible alternative futures. This paper outlines three complementary directions beyond the conventional discover-then-simulate paradigm. *Scenario-based simulation* follows the principle: model less, reuse more (e.g., by sampling, recombining, and reweighting observed object-centric structures). *Short-term simulation* starts from the current operational state and fast-forwards it into the near future. *High-level simulation* abstracts from individual events and objects to aggregate flows and time series. Together, these directions outline a research agenda for learning object-centric simulation models that are closer to the true fabric of operational processes.

Wil M.P. van der Aalst
RWTH Aachen University, Process and Data Science, D-52074 Aachen, Germany.
Celonis, Theresienstraße 6, 80333 München. e-mail: wvdaalst@pads.rwth-aachen.de

Keywords Discrete Event Simulation · Process Mining · Object-Centric Event Data

1 Generative Event-Driven Process Intelligence

For four decades, the *European Conference on Modeling and Simulation* (ECMS) has served as a major forum for advances in modeling and simulation, bringing together researchers from engineering, computer science, and the social sciences to develop new simulation techniques and applications. Since its inception in the late 1980s, ECMS has helped define the state of the art in modeling and simulation and has fostered interdisciplinary collaboration across domains [31].

During this period, simulation has matured into a powerful methodology for analyzing complex systems and evaluating alternative designs [29, 30, 32, 36, 46, 52]. However, traditional simulation approaches often rely on manually constructed models and assumptions that may not accurately reflect the real behavior of operational processes. At the same time, organizations increasingly generate detailed event data through enterprise information systems, sensors, and digital platforms.

Process mining and the broader field of process intelligence provide a bridge between real-life operations and simulation models [3, 11]. By extracting process models, performance characteristics, and behavioral patterns directly from event data, process mining enables the automatic discovery, validation, and continuous calibration of simulation models. This data-driven perspective transforms simulation from a largely model-centric activity into a closed loop connecting data, models, analysis, and improvement.

Process mining can be used to diagnose the “as-is” situation. Using *action-oriented* process mining techniques [45] it is possible to react to performance and compliance problems. If there are enough data and the process is stable, then it is even possible to *predict* performance and compliance problems. However, process mining is less suitable for exploring “to-be” situations. This requires the generation of alternatives and evaluating these. This is the field where traditional *Discrete Event Simulation* (DES) excels. Therefore, process mining and simulation have been combined in commercial tools (e.g., Apromore, ARIS, Celonis, iGrafx, and Signavio) and various research groups have studied the combination of both, starting with the work of Anne Rozinat et al. twenty years ago [53, 55].

Figure 1 illustrates how process mining and simulation can complement each other. First of all, it is possible to use *process discovery techniques* covering different perspectives to create an integrated case-centric simulation model [3, 55]. In such models, cases are independent but may compete for the same resources. Given the “as-is” simulation model it is possible to generate alternative “to-be” simulation models to explore different alternatives. By simulating these “to-be” simulation models, it is possible to create a *collection* of case-centric event data sets (typically called event logs). This allows us to use *comparative process mining techniques* [8, 10, 12, 18]. This is very powerful because reality (the “as-is” event log) and possible alternatives (the “to-be” event logs) can be compared using standard process mining techniques. The “lens” to look at reality is the same as the “lens” to look at possible alternative processes. The situation sketched

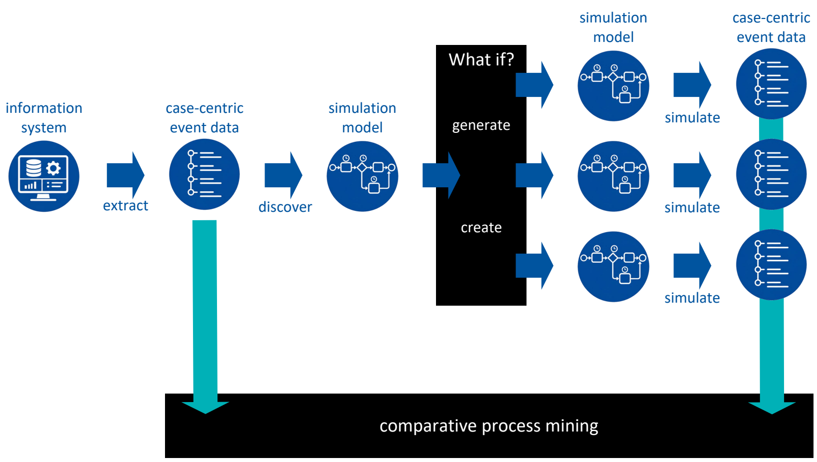


Fig. 1 Connecting classical case-centric process mining to traditional simulation approaches using explicit simulation models. Process mining techniques are used to automatically generate an "as-is" simulation model and to compare the actual process with multiple "to be" processes using the same process-mining-based visualizations.

in Figure 1 is supported by various tools. However, it remains extremely challenging to create a faithful "as-is" model and to support generation of "to-be" models.

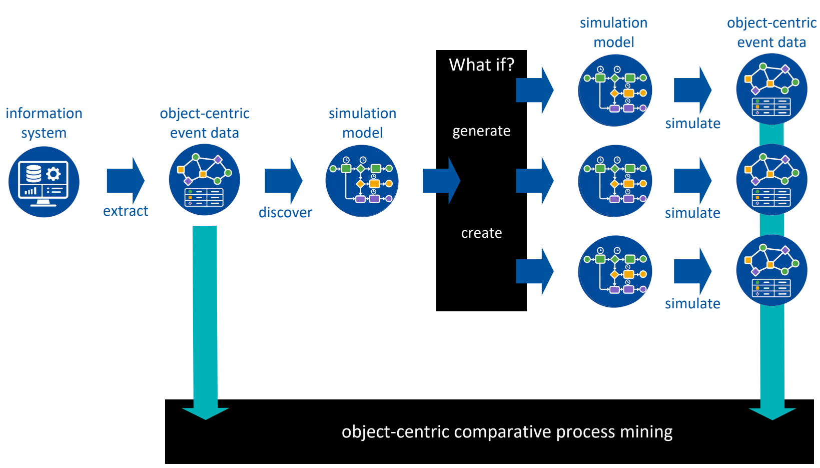


Fig. 2 Moving from case-centric to object-centric: operational reality is not a sequence of isolated cases. It is a network of interacting objects that move, wait, synchronize, and compete for resources.

Figure 2 lifts the idea depicted in Figure 1 from case-centric to object-centric. An object-centric process does not focus on one single object type. Instead, it describes how different kinds of objects (such as orders, items, customers, deliveries, invoices, and machines) develop and interact over time. The same event may involve several objects at once: for example, one delivery may contain items from multiple orders, while one order may be split across several deliveries. Object-centric processes, therefore, capture operations as a network of related objects and shared events, rather than forcing reality into separate, independent sequential cases. On the one hand, the transition from case-centric to object-centric is necessary to create more realistic simulation models. On the other hand, this dramatically increases complexity. The independence assumption used for case-centric processes simplifies the data needed, the models produced, and the techniques used. However, it may also lead to distortions and misleading results. There are a few attempts to automatically discover object-centric simulation models. However, these techniques are still in their infancy.

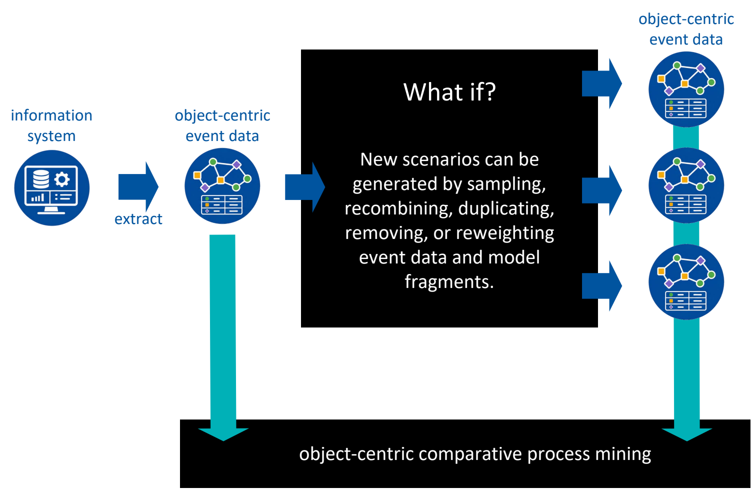


Fig. 3 Towards generative event-driven process intelligence without constructing end-to-end process models.

It is already difficult to learn a faithful object-centric simulation model that generates object-centric event data comparable to the data used to learn the model. In fact, defining what "comparable" means is itself challenging, and there is no consensus on the appropriate notions of similarity in such more complex settings. More importantly, the ultimate goal is not to reproduce the original event data. The real objective is to explore alternative processes and answer "What if?" questions.

Ideally, an approach should therefore also be able to *generate* meaningful process alternatives. The uptake of generative AI creates new possibilities for producing improvement suggestions and alternative process designs. However, such alternatives still need to be evaluated. Figure 3 illustrates our vision of approaches that do *not* require the construction of complete end-to-end simulation models. Instead, simulation and historical data can be combined to generate and evaluate alternatives.

Consider, for example, a setting in which customers may place multiple orders, and each order may contain multiple products. A conventional simulation approach would attempt to model the underlying choices explicitly. It might describe how individual customers decide when to place an order, which products to select, how many products to combine in one order, whether to cancel an order, and how their decisions depend on prices, previous purchases, delivery times, promotions, or other contextual factors. Constructing such a behavioral model is notoriously difficult because many of these decision mechanisms are only partially observable in the event data. Moreover, faithfully modeling heterogeneous customers may require detailed assumptions about latent customer types and their individual preferences.

An alternative is to *avoid modeling these choices explicitly*. Suppose that a large collection of observed customer, order, and product variants have been recorded over a longer period. These variants can be reused as realistic building blocks. Entire observed behavioral patterns, or fragments thereof, can be sampled and recombined to construct new hypothetical populations of customers. For example, one may create a scenario with more customers who place many small orders, more customers buying particular combinations of products, or fewer customers exhibiting rare and operationally expensive behavior. The resulting customers are hypothetical, but their behavior is grounded in patterns that actually occurred.

This is fundamentally different from modeling how customers make choices. The approach does not attempt to explain or reproduce the decision process that led a customer to select product P_1 rather than P_2, or to place three orders rather than one. Instead, it treats observed patterns of choices as *reusable empirical material*. The emphasis shifts from modeling the internal logic of each customer to controlling the composition of the population being simulated. This makes it possible to change the mix of hypothetical customers without modeling them in detail. For instance, one can increase the proportion of high-frequency customers, customers ordering complex product bundles, customers associated with long delivery routes, or customers exhibiting combinations of characteristics that are rare in the original data. Such changes can be used to stress-test the process and evaluate proposed alternatives under different demand profiles. Hence, historical data provides realistic behavioral building blocks, while simulation is used selectively to modify their composition, introduce process changes, and evaluate their consequences.

We believe that process mining and process intelligence can *revitalize classical simulation research*. In earlier work, we demonstrated how event data can be used to discover simulation models, detect deviations between simulated and real behavior, generate realistic parameter distributions, and enable continuous model updating. The resulting synergy supports new forms of digital twins, predictive and prescriptive analytics, and decision-centric simulation environments. By integrating data-driven process discovery with established simulation techniques, the modeling and simulation community can move toward a new paradigm in which models are not static artifacts but living representations of real systems. This perspective reconnects simulation with the operational reality of organizations and opens new opportunities for research at the intersection of simulation modeling, process mining, process management, and artificial intelligence. However, as discussed above, this is far from trivial. This paper can be viewed as a research agenda raising important topics.

The remainder of this paper is organized as follows. Section 2 presents related work. *Object-Centric Process Intelligence* (OCPI) is introduced in Section 3 by showing how case-centric approaches can be lifted to object-centric and provide the context for AI.

Section 4 sketches how case-centric simulation models can be learned, and Section 5 explains that it is far from trivial to extend this towards object-centric simulation models. We then discuss interesting perspectives aiming to deal with these challenges. Section 6 introduces *scenario-based simulation*, which reuses observed variants rather than reconstructing complete process models. Instead of simulating all aspects from scratch, more of the actual data is reused to explore different scenarios. Section 7 discusses *short-term simulation*, which fast-forwards the current operational state into the near future. Section 8 provides pointers to work on *high-level simulation*, which abstracts from individual events to coarse-grained dynamics and time series. By lifting the abstraction level, it is easier to model the collective behavior without assuming independence. Section 9 concludes the paper.

2 Related Work

Simulation is a mature discipline with well-established foundations for modeling, experimentation, and statistical analysis. Classical textbooks and monographs cover simulation methodology and analysis [32, 36, 46, 52], discrete-event simulation environments [30], and more specialized formalisms such as stochastic Petri nets [29] and queuing networks [28, 40].

Despite this rich methodological foundation, constructing a simulation model has traditionally been a predominantly manual activity. Analysts determine the model structure, identify relevant entities and resources, specify routing decisions, and estimate arrival, service-time, and waiting-time distributions. This is particularly problematic for business and operational processes, where the observed behavior may differ substantially from normative process descriptions. Early work on business process simulation therefore emphasized both its potential and its practical pitfalls [1, 2, 14]. A recurring conclusion is that a simulation model is useful only to the extent that its structure and parameters faithfully represent the system being studied.

The increasing availability of event data offers an alternative. Process mining extracts behavioral and performance information directly from observed executions [3, 11]. This creates a *natural complementarity*: process mining reveals what *actually* happened, whereas simulation explores what *could* happen under changed conditions. The resulting connection has been described as a particularly natural combination of the two fields [4].

The combination of process mining and simulation predates the recent wave of machine-learning-based approaches. Maruster and van Beest combined process mining and simulation for process redesign [42], while Rozinat et al. introduced one of the first comprehensive approaches for discovering simulation models from event logs [53, 55]. Related work investigated the generation of event logs from simulation models while taking workload-dependent processing speeds into account [44] and the use of workflow simulation for operational decision support [56]. Martin et al. subsequently studied more systematically how process mining can support the construction of business process simulation models [41].

Together, these works established a bidirectional relationship between observed and simulated behavior. A recent review by Sulis et al. confirms the breadth of this research area and identifies substantial remaining opportunities at the intersection of simulation and process mining [58].

More recent work aims to automate the complete discovery of simulation models, thus refining the work of Rozinat et al. [53, 55]. Camargo et al. developed an approach for automatically discovering business process simulation models from event logs [22]. Subsequent work combined process mining with deep learning to capture aspects of process behavior that are difficult to represent using traditional parameter estimation techniques [21]. Estrada-Torres et al. extended simulation model discovery to settings with multitasking and resource availability constraints [25], while Lopez-Pintado and Dumas investigated the importance of differentiating resources in business process simulation models [38].

These studies show that learning a simulation model requires substantially more than just discovering a control-flow model. Resource behavior, calendars, batching, multitasking, temporal dependencies, and routing probabilities may all influence the behavior of the simulated system.

This line of research also raises the question of whether a discovered simulation model can be trusted. Chapela-Campa et al. proposed techniques for measuring the quality of business process simulation models by comparing generated and observed behavior [23]. Tool support has matured accordingly. SIMOD, for example, provides an integrated implementation for the automated discovery of business process simulation models [24]. Recent work additionally considers online discovery, allowing simulation models to evolve together with changing business processes [59]. Hence, the field is gradually moving from one-off model construction toward continuously discovered, evaluated, and updated simulation models.

Nevertheless, almost all of these approaches remain case-centric. They assume that events can be partitioned into process instances and that each event belongs to one case. This assumption makes model discovery tractable, but it abstracts from many of the interactions that constitute the actual operational system. *Object-Centric Process Mining* (OCPM) was developed to overcome the limitations of forcing events into a single case notion [5, 6, 7]. Instead of representing a process as a collection of isolated traces, object-centric event data represent events together with the multiple objects to which they relate. OCPM uses such data to reveal interactions between orders, items, packages, machines, employees, customers, and other operational entities [7].

OCEL 2.0 [17] is the de facto standard to represent *Object-Centric Event Data* (OCED) [26]. The research community is moving toward OCPM, lifting existing approaches for process discovery, conformance checking, and predictive analytics to the next level [7, 6]. Moreover, commercial systems started to implement OCPM. Celonis has been pioneering the field by developing a platform based on a meta model close to the original OCEL 2.0 meta model and supporting object-centric process discovery and conformance checking (e.g., OC-BPMN discovery and object-centric alignments).

Moving from case-centric to object-centric simulation is a natural but nontrivial next step. Initial work argued that object-centric process mining can support the construction of more realistic simulation models [8]. Knopp et al. subsequently presented an approach for discovering object-centric process simulation models [33]. These contributions demonstrate the the theoretical feasibility of the idea, but they also expose the fundamental difficulties discussed in this paper. Despite the uptake of OCPM, progress in object-centric simulation-model discovery has been limited.

An object-centric simulation model must reproduce not only activity sequences and durations, but also the creation and disappearance of objects, relations between objects, cardinalities, synchronization, resource sharing, and interactions between concurrently evolving object populations. This difference is fundamental. In a case-centric setting, a simulated case is typically generated, routed through a model, and completed. In an object-centric setting, there is no single process instance that can be simulated independently. Objects may already exist when other objects are created, a single event may affect multiple objects, and future behavior may depend on a changing network of relations.

The problem is therefore not merely to generalize a conventional simulation model to multiple case types. The challenge is to learn the evolving fabric that connects all relevant objects.

Most work connecting process mining and simulation assumes that a detailed end-to-end process model should first be discovered and then simulated. However, several strands of prior research suggest alternatives to this *discover-then-simulate* paradigm.

First, simulation does not necessarily require a fully detailed process model. Pourbafrani and van der Aalst showed how process features can be extracted from event logs to learn coarse-grained simulation models [47]. Related work supports the automatic generation of system dynamics models in a process mining context [50] and discovers such models directly from event data [48]. Hybrid approaches combine detailed process simulation with high-level simulation models, allowing coarse-grained dynamics to update or steer more detailed models [49].

This body of work provides the foundation for the *high-level simulation* direction discussed later: when individual events and objects are too detailed, it may be preferable to learn aggregate flows, populations, delays, and time-dependent dynamics.

Second, simulation can be used for near-term operational decision support rather than only for strategic, long-term experimentation. The notion of short-term simulation was introduced to bridge operational control and strategic decision making [51]. Instead of repeatedly simulating an artificial system from an empty initial state, the simulation starts from the observed current state and “fast-forwards” the system into the near future. Workflow simulation for operational decision support explored a closely related objective [56]. Object-centric event data make this direction particularly compelling because they provide a much richer representation of the current state, including the objects currently present and the relations between them.

Third, the increasing complexity of automatically discovered simulation models raises the question of whether an explicit generative process model is always necessary. The work on simulation-model discovery demonstrates the growing sophistication required to learn routing, timing, resource behavior, and other dependencies [21, 22, 23, 24, 25, 38, 53, 55, 59].

This motivates the *scenario-based simulation* direction explored in this paper. Rather than reconstructing the entire generative mechanism, one may reuse, combine, and adapt observed behavioral variants, thereby preserving dependencies that would otherwise need to be learned explicitly. For object-centric processes, where a “variant” is no longer a simple sequence but a connected structure of events and objects, this opens a largely unexplored research direction.

In summary, existing research has progressed from manually constructed simulation models [1, 2, 14, 29, 30, 32, 36, 46, 52], through the integration of process mining and simulation [4, 41, 42, 44, 55, 56, 58], to increasingly automated discovery and continuous updating of simulation models [22, 21, 25, 38, 23, 24, 59].

In parallel, object-centric process mining has made it possible to represent and analyze the interactions that are lost in conventional event logs [7], leading to first steps toward object-centric simulation [8, 33].

3 Object-Centric Process Intelligence

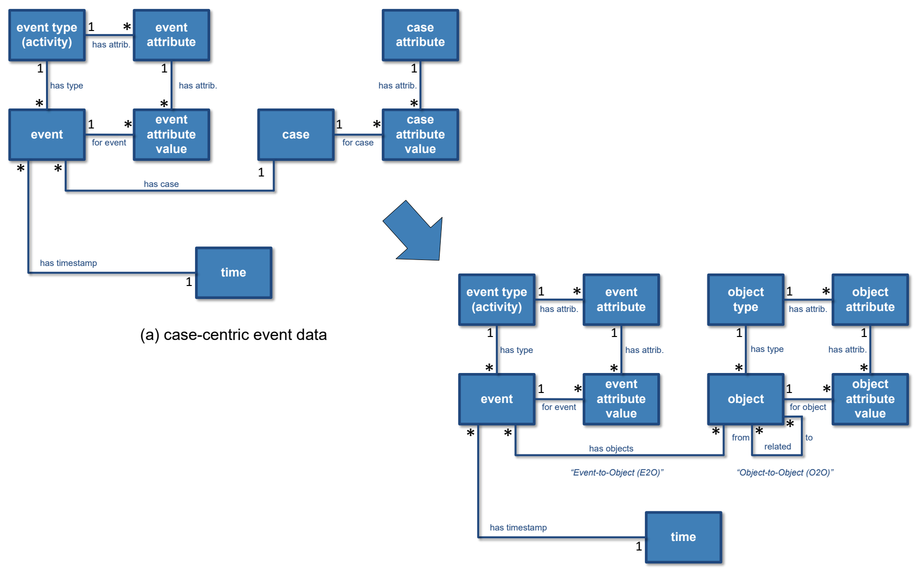
Object-Centric Process Intelligence (OCPI) aims to describe, analyze, predict, and improve operational processes without imposing an artificial case notion. The starting point is the observation that operational reality consists of interacting objects. Customers place orders, orders contain items, items are grouped into deliveries, deliveries use vehicles, invoices refer to orders, and employees and machines work on many such objects over time. Events change this evolving network. In this section, we briefly describe Object-Centric Event Data (OCED) and Object-Centric Process Mining (OCPM) techniques using such data. Then we combine these with Artificial Intelligence (AI) and Machine Learning (ML). We refer to the amalgamation of these ingredients as OCPI.

3.1 Object-Centric Event Data

Figure 4 shows the conceptual shift from case-centric event data to object-centric event data. The key idea is simple: a case-centric event log forces every event to be linked to precisely one case. Object-centric event data allows an event to refer directly to multiple interacting objects that may be related independent of events.

As shown in Figure 4(a), traditional case-centric event data assume a single case notion. Each event refers to (1) one *activity type* called activity, (2) one *timestamp*, and (3) precisely one *case*. Events and cases can have attributes e.g., an event of type "approve claim" may have duration and refer to a location or resource. Case attributes are invariant, e.g., the customer that initiated the claim. The problem is that reality is usually richer. The "create invoice" event may involve an order, an invoice, a customer, and several items. The case-centric representation requires us to select one of these as the case and somehow flatten the others into attributes, duplicate events across logs, or ignore some relationships.

Figure 4(b) shows that in *Object-Centric Event Data* (OCED) events still refer to one activity type and one timestamp. However, an event may refer to any number of objects. The *Event-to-Object* (E2O) relation is no longer a functional relationship. Objects are also typed. Instead of a single case-notion there may be objects referring to orders, invoices, customers, suppliers, items, resources, teams, etc. Also, objects may



(a) case-centric event data

(b) object-centric event data

Fig. 4 The transition from case-centric event data to simple object-centric event data.

be related explicitly through the *Object-to-Object* (O2O) relation. O2O relations are important when querying event data. For example, to answer the question "Give me all the orders with more than 10 items that were not delivered on time". Object-Centric Process Querying (OCPQ) is a dedicated language and tool to query such data [35].

The meta model in Figure 4(b) is a simplification of the OCEL 2.0 meta model [34, 26]. In the full OCEL 2.0 meta model, the E2O and O2O relation can be qualified and attribute updates can be timed. However, to explain the main concept of object-centricity, Figure 4(b) suffices.

Despite the simplicity of Figure 4(b), the result is amazingly powerful. Object-centric event data are able to capture the true fabric of operational processes: multiple object types evolving together through shared events, rather than an artificial collection of isolated case sequences.

3.2 Object-Centric Process Mining

Object-Centric Process Mining (OCPM) builds directly on object-centric event data (OCED). Whereas OCED describe what happened, when it happened, and which objects were involved, OCPM uses these events, objects, and relations to understand the behavior of the underlying process.

Object-centric process discovery starts from the observation that each object has its own history, while many events involve several objects at the same time. Consider a process involving customers, orders, and items. A customer may place several orders,

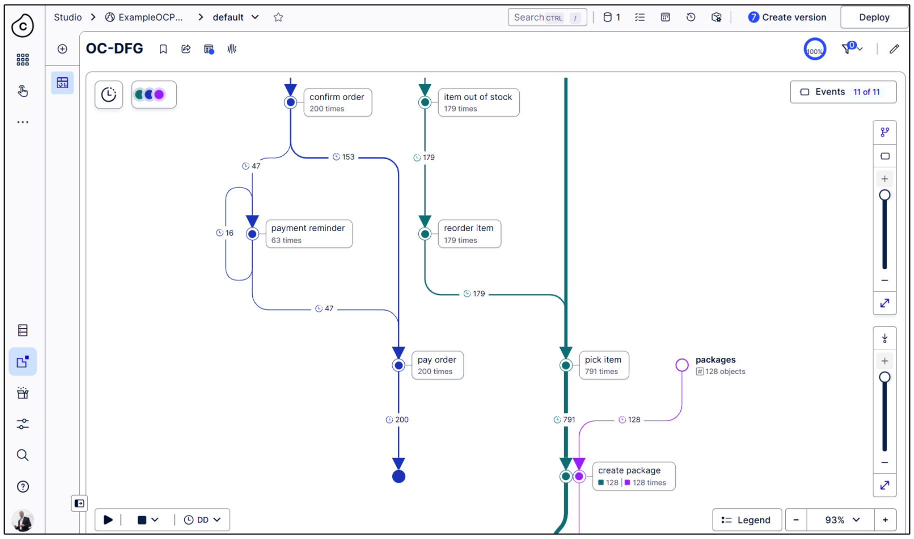


Fig. 5 Screenshot of Celonis while discovering an Object-Centric Directly-Follows Graph (OC-DFG). There are three object types: orders (blue), items (green), and packages (purple).

each order may contain several items, and an event such as placing an order may simultaneously involve one customer, one order, and multiple items. Later, a packing event may involve several items, perhaps belonging to the same order or even to different orders, while a payment may cover multiple orders of the same customer. There is therefore no single natural trace that captures the full process. Instead, discovery has to identify the life-cycle of each object type and, at the same time, the synchronizations between these life-cycles.

Multiplicity is also an important part of this behavior. One order may contain many items, several items may be combined into one package, and one payment may cover several orders. These variable numbers of participating objects are part of the process itself and should not be lost by reducing everything to a single case notion. An object-centric process model, therefore, represents the joint behavior of interacting object types rather than a collection of unrelated case-centric models. This multiplicity is reflected by so-called *variable arcs* in object-centric process models such as Object-Centric Directly-Follows Graphs (OC-DFGs), Object-Centric Petri Nets (OC-PNs), and Object-Centric Business Process Model and Notation (OC-BPMN) [7, 9]. Figure 5 shows a discovered OC-DFG for a small data set.

Object-centric conformance checking starts from such an object-centric process model and asks how well the observed behavior in the OCED fits it. The model may have been discovered from historical data, created manually, or intended as a normative description of how the process should operate. Figure 6 shows conformance diagnostics obtained using object-centric alignments. The BPMN model shown is the normative process showing only the "happy paths". Several deviations are detected, e.g., failed deliveries, payment reminders, etc. Figure 6 zooms in on the problem of items being out of stock leading to significant delays.

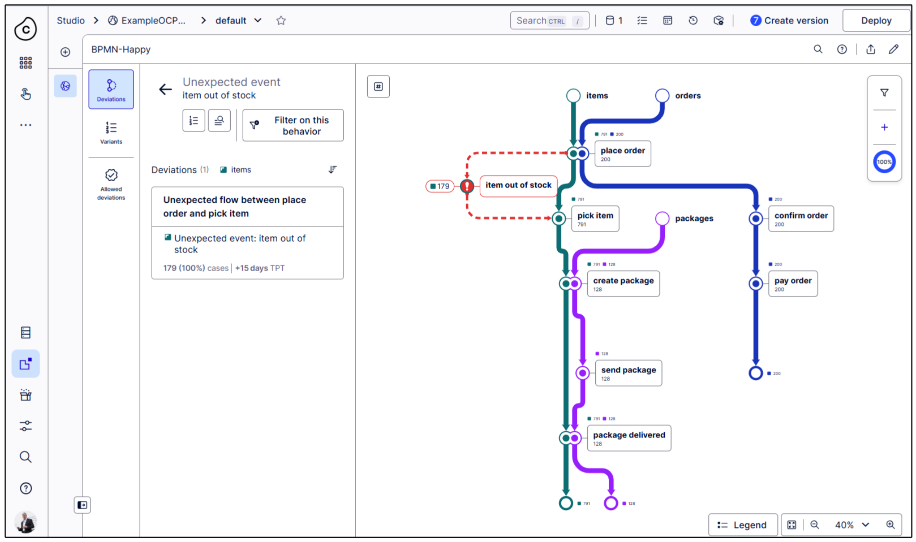


Fig. 6 Screenshot of Celonis while checking the conformance of a small data set with respect to a simple OC-BPMN. Again, there are three object types: orders (blue), items (green), and packages (purple). One of the most frequent deviations is selected: 179 times, an item was out of stock when ordered. This happened in-between the activities place order and pick item. This deviation added on average 15 days to the flow time.

Figures 5 and 6 illustrate that Celonis natively supports OCPM. As indicated in the Gartner Magic Quadrant for Process Intelligence Platforms [57], OCPM is a key capability and many vendors are in the process of implementing this.

3.3 Enabling AI and ML

OCPM provides a powerful foundation for applying AI and machine learning to operational performance and compliance problems. Object-centric event data and process models serve as a lens through which such problems can be formulated in a process-aware manner, preserving the relevant context of interacting objects, events, and dependencies.

For each concrete problem, one can construct a dedicated *situation table* in which every row represents a relevant situation and the columns capture the process context needed for prediction, classification, root-cause analysis, or decision support. In this way, OCPM systematically transforms rich operational data into meaningful input for machine-learning techniques, rather than applying AI to isolated records without understanding the surrounding process. For example, the problem identified in Figure 6 can be used to generate a situation table. Based on this table, one is able to diagnose and even predict when items are out of stock.

Generative AI further extends these possibilities by enabling users to interact with process data and models in a more natural way, formulate questions in ordinary language, generate explanations and improvement suggestions, and connect quantitative process insights to contextual knowledge. *OCPM thus provides the grounding and structure that AI needs to reason about operational reality, while AI and ML add predictive, prescriptive, and conversational capabilities on top of the process perspective.*

4 Discovering Case-Centric Simulation Models

Compared to the object-centric setting discussed in the next section, discovering a case-centric simulation model from event data is conceptually relatively straightforward. A conventional event log provides a collection of cases, where each case consists of a sequence of timestamped events. Assuming that events also contain resource attributes, for example identifying the person, machine, or system executing an activity, a surprisingly large part of a discrete-event simulation model can be learned directly from the observed data.

The most obvious element to discover is the *set of activities*. Each distinct activity type occurring in the event data corresponds to a potential step in the simulation model. Process discovery techniques can then be used to learn the routing structure between these activities. Depending on the desired modeling language, this may result in a Directly-Follows Graph (DFG), a Petri net, a BPMN model, or another executable process representation. The discovered structure determines which activities may follow one another, which activities can occur concurrently, and where choices, loops, and synchronization points occur.

The event data can also be used to estimate the *routing behavior of cases* through this structure. For a choice between alternative paths, the observed frequencies can be translated into routing probabilities. More sophisticated approaches can make these probabilities dependent on case attributes and the history of the case. For example, the probability that an insurance claim is sent to an additional review may depend on the claim amount, customer type, or activities already performed. Hence, discovery does not need to be restricted to fixed branching probabilities: data-driven decision rules or learned predictive models may determine the routing of individual simulated cases.

Timestamps provide the basis for learning the *temporal perspective*. When both start and completion timestamps are available, activity durations can be observed directly and suitable probability distributions can be fitted. When only completion timestamps are present, durations and waiting times are harder to disentangle and additional assumptions are needed. Nevertheless, the data can be used to estimate inter-arrival times, processing times, waiting times, time-dependent arrival rates, and other temporal characteristics. Rather than assuming simple textbook distributions, one may use empirical distributions or more expressive models that preserve dependencies on the activity, case attributes, resource, workload, time of day, or day of the week.

Resource information attached to events makes it possible to discover the organizational perspective. The observed resources indicate who or what is able to execute particular activities. Resources can be grouped into roles or pools based on their observed behavior, and resource-activity relationships can be learned directly from the log. The timestamps of resource executions can also reveal working calendars, shifts, breaks, and periods of unavailability. If the data are sufficiently rich, one can additionally learn differentiated processing speeds, resource-selection rules, workload-dependent behavior, multitasking patterns, and the tendency of particular resources to perform certain types of work. For example, in [43] we explored the effect of workload on service times and indeed found evidence for the "Yerkes-Dodson Law of Arousal" [60]. Thus, the event log provides information not only about what happens and in which order, but also about who performs the work, when resources are available, and even the effect of workload on processing speeds.

A case-centric simulation model also requires a model for the *creation of new cases*, i.e., the *arrival process*. This can be learned from case start times. The resulting arrival process may be stationary, but in many real settings arrival rates depend on the hour of the day, the day of the week, the season, or external conditions. Case attributes can be sampled from their empirical distributions and may also be used to preserve important dependencies. For example, urgent cases may arrive according to a different temporal pattern, follow different routes, and require different processing times than regular cases. A faithful simulation model should therefore avoid learning all model components independently when the event data reveal strong correlations between them.

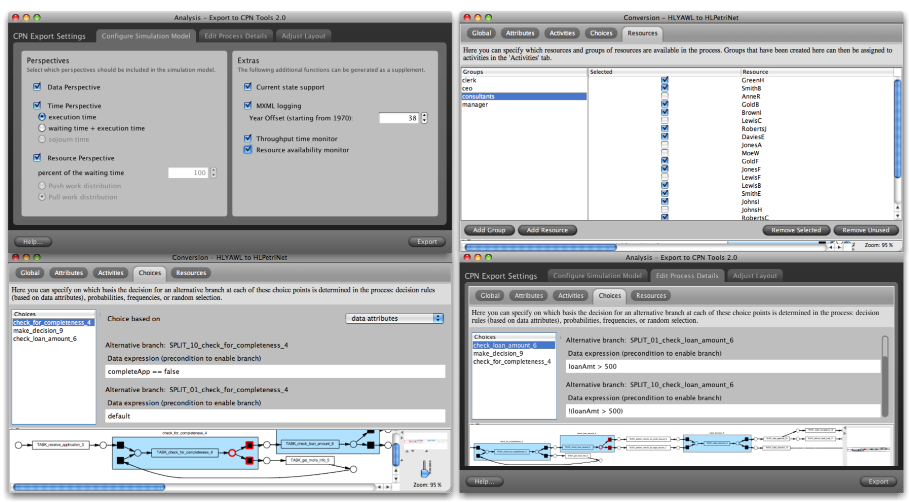


Fig. 7 A few screenshots from *ProM 4.2* illustrating the ability to discover the different perspectives from event data and merging them into an integrated simulation model (https://processmining.org/old-version/simulation.html).

The approach to process mining to learn simulation models was first described and implemented in [53, 54, 55] twenty years ago (see Figure 7). The process mining framework *ProM* was used to discover and merge the different perspectives: control-flow, time, probabilities, data-driven decision points, resource availability, etc. Based on such an integrated process model, *ProM* could automatically generate a simulation models in *CPN Tools*. This was used to conduct "What if" analyses in Dutch municipalities. Simulation tools such as *ExSpect* and *CPN Tools* were integrated with *Protos* of Pallas Athena and *ProM* [27]. This shows that the desire to connect process mining, simulation, workflow automation, and process management has been around for decades.

Although many approaches exist (see Section 2), even case-centric simulation model discovery is not a solved problem. Event data record what happened, whereas a simulation model must specify what could happen under conditions not yet observed. Logs may be incomplete, timestamps may have different semantics, resource attributes may be missing or unreliable, and observed waiting times may be caused by hidden priorities or dependencies. Nevertheless, the key simplification in the case-centric setting is that each case can be treated as a distinct process instance. Cases may compete for shared resources, but their identities, histories, and boundaries are given in the event data. It is precisely this assumption that makes the discovery of case-centric simulation models manageable. However, this assumption no longer holds when moving to the true object-centric fabric of operational processes.

5 What Makes the Discovery of Object-Centric Simulation Models Much Harder?

The transition from case-centric to object-centric simulation is not simply a matter of adding more case types. In a case-centric simulation, a new case is created, follows a route, and eventually completes. Cases may compete for resources, but their identities and boundaries are clear. In an object-centric process, customers, orders, items, products, packages, and deliveries are intertwined. Their lifecycles overlap, and events continuously create and modify the network connecting them. The simulation model must therefore generate an evolving fabric of objects and events rather than a collection of independent traces. This evolving fabric of objects and events is precisely the structure represented by the E2O and O2O relations introduced in Figure 4(b).

Consider customers, orders, products, and deliveries. There is no simple function that takes a customer and determines which products this customer will buy, how these products will be distributed over orders, and through which deliveries they will eventually arrive. These outcomes emerge from many interacting mechanisms that are only partially visible in the event data. Generating customers, orders, products, and deliveries independently and connecting them afterwards would destroy important dependencies.

The dynamics influence both E2O and O2O relations. Placing an order connects a customer to a new order and the order to products. Packing may connect items to a package, while consolidation may connect one delivery to items from several orders. The model therefore has to learn not only which event happens next, but also which objects participate and how their relations change.

This creates a *circular dependency*. The current object network determines which events can occur, while events modify that same network. For example, whether a delivery can be created may depend on item availability, order composition, destinations, and consolidation rules. Once created, the delivery changes the object network and influences future behavior.

A faithful simulation must also preserve object identity and connectivity over time. If a customer places several orders and items from these orders are later combined in one delivery, all subsequent events must retain the correct links to the original customer, orders, and items. Locally plausible events are not enough if these connections are lost. *Splits, merges, and changing cardinalities must remain globally consistent.*

This is why separately discovering a life-cycle model for each object type is insufficient. The main difficulty lies in the *synchronization between object life-cycles*: an order may wait for an unavailable item, a delivery may depend on several orders, and a customer may create new orders while earlier ones are still active. The discovery problem is therefore not merely to learn control flow, timing, and resources, but to learn how a dynamically changing network of related objects evolves.

This is what makes object-centric simulation fundamentally harder. The model must generate not only realistic object behavior, but also the evolving connections that make these objects part of one operational reality. This motivates the alternative directions discussed next: reusing observed connected structures, simulating only the near future from the current object network, or moving to a higher level of abstraction.

6 Scenario-Based Simulation: Model Less, Reuse More

Scenario-based simulation starts from the observation that, in an object-centric setting, much of the behavior that is difficult to model explicitly is already present in the historical data. Rather than discovering a complete generative model for the creation, evolution, and interaction of all objects, one can reuse observed connected structures of events and objects as *building blocks* for simulation. These structures may correspond to complete historical populations, connected fragments, or recurring object-centric variants involving, for example, customers, orders, items, packages, and deliveries.

New scenarios can then be constructed by sampling, recombining, duplicating, removing, or reweighting such structures. For example, one may increase the proportion of customers placing many small orders, introduce more orders containing difficult product combinations, or create a larger share of deliveries that consolidate items from several orders. The key idea is to preserve complex empirical dependencies by reusing what actually happened and to simulate only those aspects that need to change. The challenge is to determine which parts of observed object-centric behavior can be safely reused, how structures may be recombined without creating inconsistencies, and how hypothetical changes can be introduced while retaining sufficient realism. In this sense, scenario-based simulation follows the principle “model less, reuse more” and shifts the emphasis from reconstructing the full process-generating mechanism to constructing plausible alternative worlds.

To explain the idea, consider a simple case-centric event log abstracting from time, resources, data, etc. Such an event log can be described as a multiset of traces. This can be turned into a stochastic language [13, 37] assigning a probability to each trace. A simple example is an event log $[\langle a, b, c, e\rangle^{500}, \langle a, c, b, e\rangle^{300}, \langle a, d, e\rangle^{200}]$ with 1000 cases that is transformed into a stochastic language $[\langle a, b, c, e\rangle^{0.5}, \langle a, c, b, e\rangle^{0.3}, \langle a, d, e\rangle^{0.2}]$, e.g., trace $\langle a, d, e\rangle$ has a probability of 20%. In this example, it is easy to create a simulation model that generates a sample from the same stochastic language. After a there is a 20% probability to do d and a 80% probability to do both b and c. If both b and c need to happen, there is a higher likelihood that b will happen first (62.5%). Existing techniques are able to discover a stochastic model having precisely this behavior [20, 37]. For example, a Generalized Stochastic Petri Net (GSPN) [13, 15] assigning weight 0.5 to transition b, weight 0.3 to transition c, and weight 0.2 to transition d generates the same stochastic language. The well-known Earth Movers' Distance (EMD) can be used to compare observed and modeled behavior [37].

However, now consider the stochastic language $[\langle a, b, c, e\rangle^{0.4}, \langle a, c, b, f\rangle^{0.2}, \langle a, b, c, f\rangle^{0.1}, \langle a, c, b, e\rangle^{0.1}, \langle a, d, e\rangle^{0.2}]$. Suddenly, things get much more difficult. The initial behavior involving a, b, c, and d is still the same. However, at the end the choice between e and f depends on the choice between d and b and c, and even the ordering of b and c. If b happens first, there is higher likelihood that later e follows. If c happens first, there is higher likelihood that later f follows. In general it is impossible to capture such behavior in a stochastic Petri nets where each activity corresponds to a single transition. However, it is easy to simply sample the different variants without creating a traditional model

Note that we did not consider time, resources, data, etc. and already have problems producing the "same" behavior. The reason is that things are not independent. Also, it is easy to define Earth Movers' Distance (EMD) for stochastic languages, but it gets more involved when time and other perspectives play a role [19].

When lifting simulation from case-centric processes to object-centric processes, things becomes *even more challenging* due to the dependencies explained before. A package delivered to a customer needs to contain items ordered by that customer. However, due to (un)availability of items, items in one order may end up in different deliveries and items of different orders may end up in the same delivery. This quickly becomes intractable to model. Therefore, it may be better to sample, recombine, duplicate, remove, or reweigh elements of the input data to explore alternative scenarios. Possible terms to describe such ideas are: scenario-based simulation, generative what-if analysis, what-if process exploration, and process-scenario editing. This is related to the "improve your history" idea presented in [39] where the event data are split into two parts: a fixed part and a variable part. The fixed part of the event data is kept the same (e.g., the activities performed and their durations, and the arrival times of process instances) while other elements within the event data (e.g., resource allocations and start times of activities) are adjusted to explore different execution scenarios. The "improve your history" approach then searches for alternative execution scenarios by perturbing the variable part.

7 Short-Term Simulation: Fast Forward Into The Future

Short-term simulation avoids another difficult aspect of conventional simulation: generating a realistic system from an artificial empty initial state. Instead, it starts from the current operational state observed in the object-centric event data and fast-forwards this state into the near future. At any point in time, the OCED contains customers with open orders, partially completed items, packages waiting for transport, deliveries already in progress, resources with current workloads, and many relations connecting these objects. This rich current state provides a much stronger starting point than a collection of independent unfinished cases. The task is therefore not to reproduce the entire long-run dynamics of the system, but to predict and simulate what is likely to happen next over a limited horizon. Such simulations can support operational questions such as which deliveries will be late, where congestion will emerge, which resources will become overloaded, and what the consequences are of reallocating work or changing priorities. Because the prediction horizon is short, more of the current object network can be treated as given and fewer assumptions about long-term object creation and population dynamics are needed. Nevertheless, the approach must still capture how ongoing object life-cycles synchronize, how newly created objects interact with existing ones, and how uncertainty grows as the simulation moves further into the future. Short-term object-centric simulation therefore provides a promising middle ground between pure prediction and full end-to-end simulation.

The idea to use short-term simulation as a “fast forward button” into the future was first proposed in 1999 using a combination of tools (*Protos*, *COSA*, *FLOWer*, and *ExSpect*) [51]. A more mature prototypical environment involving *YAWL*, *ProM*, and *CPN Tools* was described in [56].

With the uptake of OCPM and the realization that AI needs “context” (structured information about related objects and events when making decisions), the topic is more relevant than ever before.

8 High-Level Simulation: From Events To Time Series

A third direction to handle complexity is to deliberately abandon the attempt to simulate every individual event and object. When the interactions between large object populations are too complex to reproduce faithfully at the microscopic level, it may be more useful to model aggregate dynamics.

Object-centric event data can be *transformed into time-dependent features* describing, for example, the numbers of active orders, waiting items, available resources, delayed deliveries, newly created customers, or objects moving between coarse-grained process states. These features form multivariate time series that capture the collective behavior of the operational system. High-level simulation models, including system dynamics [48] and other coarse-grained approaches [16], can then be learned from these data and used to study feedback loops, bottlenecks, delays, and changes in population composition. The advantage is that dependencies between many interacting objects need not be reconstructed individually: their combined effect may become visible through aggregate flows and stocks. This is particularly attractive for strategic “What if?” questions concerning substantial changes in demand, capacity, or policy. At the same time, the choice of abstraction is crucial, because aggregation may hide precisely the interactions that cause performance or compliance problems. An important research direction is therefore to connect levels of abstraction, allowing high-level simulations to steer more detailed models or to zoom in selectively when aggregate behavior indicates that a particular part of the process requires closer analysis.

9 Conclusion

Process mining and simulation are highly *complementary*. Process mining starts from observed event data and reveals what actually happened, whereas simulation starts from models and explores what could happen under changed conditions. Process mining is therefore particularly strong in diagnosing the “as-is” situation, detecting performance and compliance problems, and learning from operational reality. Simulation, in contrast, is particularly strong in exploring “to-be” situations and answering “What if?” questions. Connecting the two creates a natural closed loop: *event data are used to learn and calibrate simulation models, simulation generates possible alternative futures, and process mining provides a common lens for comparing these alternatives with reality*. This complementarity has motivated research for more than two decades and is becoming increasingly relevant as organizations generate richer event data and demand more data-driven support for operational decision making.

This connection also fits naturally with the history of the *European Conference on Modeling and Simulation* (ECMS). For four decades, ECMS has provided a forum for developing simulation methods and applying them to increasingly complex systems. Much of the traditional work in modeling and simulation necessarily started from manually constructed models, expert knowledge, and assumptions about system behavior. This was not a limitation of simulation as a discipline, but a consequence of the data that were available. Today, the situation is fundamentally different. Operational systems continuously produce detailed event data describing activities, objects, resources, and interactions. Process mining and process intelligence make it possible to use these data to discover, validate, update, and challenge simulation models. Hence, the future of simulation can build on the methodological foundations developed by the ECMS community while becoming much more tightly connected to the operational reality from which the models originate.

The *transition from case-centric to object-centric simulation* is an important part of this development. Operational reality is rarely a collection of independent cases. Customers, orders, items, products, packages, deliveries, resources, and many other objects evolve together and interact through shared events. Moving from case-centric event logs to OCED and OCPM makes this “fabric” visible, but it also exposes the limits of the traditional *discover-then-simulate* paradigm.

This paper, therefore, outlined three complementary directions. *Scenario-based simulation* reuses observed connected structures and modifies their composition rather than reconstructing all underlying choices. *Short-term simulation* starts from the current object-centric state and fast-forwards it into the near future. *High-level simulation* abstracts from individual events and objects to population dynamics and time series. These directions are not mutually exclusive. Future simulation environments may combine them: historical structures may be reused where realistic data are abundant, selected mechanisms may be simulated explicitly where interventions are introduced, the immediate future may be explored from the current operational state, and long-term feedback effects may be represented at a higher level of abstraction.

The broader research agenda is therefore not simply to discover ever more detailed simulation models. The more important question is what should be modeled, what can be learned, what can be reused, and what can be abstracted away in order to evaluate meaningful alternatives. *Generative AI* makes this agenda even more urgent. It can increasingly propose process changes, alternative operating policies, and new organizational designs, but generating alternatives is only useful when their consequences can be evaluated in a way that remains *grounded in operational reality*. Process mining provides this grounding, whereas simulation enables moving beyond the observed past.

References

1. W.M.P. van der Aalst. Business Process Simulation Revisited. In J. Barjis, editor, *Enterprise and Organizational Modeling and Simulation*, volume 63 of *Lecture Notes in Business Information Processing*, pages 1–14. Springer-Verlag, Berlin, 2010.
2. 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.
3. W.M.P. van der Aalst. *Process Mining: Data Science in Action*. Springer-Verlag, Berlin, 2016.
4. 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.
5. 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.
6. W.M.P. van der Aalst. Object-Centric Process Mining: An Introduction. In A. Cerone, editor, *Formal Methods for an Informal World, ICTAC 2021 Summer School*, volume 13490 of *Lecture Notes in Computer Science*, pages 73–105. Springer-Verlag, Berlin, 2023.
7. W.M.P. van der Aalst. Object-Centric Process Mining: Unraveling the Fabric of Real Processes. *Mathematics*, 11(12):2691, 2023.
8. W.M.P. van der Aalst. Toward More Realistic Simulation Models Using Object-Centric Process Mining. In E. Vicario, R. Bandinelli, V. Fani, and M. Mastroianni, editors, *Proceedings of the 37th ECMS International Conference on Modelling and Simulation, (ECMS 2023)*, pages 5–13. European Council for Modeling and Simulation, 2023.
9. W.M.P. van der Aalst and A. Berti. Discovering Object-Centric Petri Nets. *Fundamenta Informaticae*, 175(1-4):1–40, 2020.
10. 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.

11. 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.
12. W.M.P. van der Aalst, S. Guo, and P. Gorissen. Comparative Process Mining in Education: An Approach Based on Process Cubes. In P. Ceravolo, R. Accorsi, and P. Cudre-Mauroux, editors, *IFIP International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA 2013)*, volume 203 of *Lecture Notes in Business Information Processing*, pages 110–135. Springer-Verlag, Berlin, 2015.
13. W.M.P. van der Aalst and S.J.J. Leemans. Learning Generalized Stochastic Petri Nets From Event Data. In N. Jansen, S. Junges, B.L. Kaminski, C. Matheja, T. Noll, T. Quatmann, M. Stoelinga, and M. Volk, editors, *Principles of Verification: Cycling the Probabilistic Landscape - Essays Dedicated to Joost-Pieter Katoen on the Occasion of His 60th Birthday, Part III*, volume 15262 of *Lecture Notes in Computer Science*, pages 3–17. Springer-Verlag, Berlin, 2024.
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. M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. John Wiley and Sons, 1995.
16. B. Bakullari and W.M.P. van der Aalst. High-Level Event Mining: Overview and Future Work. *Computing Research Repository (CoRR) in arXiv*, abs/2405.14435, 2024.
17. A. Berti, I. Koren, J.N. Adams, G. Park, B. Knopp, N. Graves, M. Rafiei, L. Liß, L. Tacke Genannt Unterberg, Y. Zhang, C. Schwanen, M. Pegoraro, and W.M.P. van der Aalst. OCEL (Object-Centric Event Log) 2.0 Specification. www.ocel-standard.org, 2023.
18. A. Bolt, M. de Leoni, and W.M.P. van der Aalst. A Visual Approach to Spot Statistically-Significant Differences in Event Logs Based on Process Metrics. In S. Nurcan, P. Soffer, M. Bajec, and J. Eder, editors, *International Conference on Advanced Information Systems Engineering (Caise 2016)*, volume 9694 of *Lecture Notes in Computer Science*, pages 151–166. Springer-Verlag, Berlin, 2016.
19. T. Brockhoff, M.S. Uysal, and W.M.P. van der Aalst. Time-aware Concept Drift Detection Using the Earth Mover's Distance. In B.F. van Dongen, M. Montali, and M. Wynn, editors, *International Conference on Process Mining (ICPM 2020)*, pages 33–40. IEEE Computer Society, 2020.
20. A.T. Burke, S.J.J. Leemans, M.T. Wynn, W.M.P. van der Aalst, and A.H.M. ter Hofstede. Stochastic Process Model-Log Quality Dimensions: An Experimental Study. In A. Burattin, A. Polyvyanyy, and B. Weber, editors, *International Conference on Process Mining (ICPM 2022)*, pages 80–87. IEEE, 2022.
21. M. Camargo, D. Baron, M. Dumas, and O. Gonzalez-Rojas. Learning business process simulation models: A Hybrid process mining and deep learning approach. *Information Systems*, 117:102248, 2023.
22. M. Camargo, M. Dumas, and O. González-Rojas. Automated Discovery of Business Process Simulation Models From Event Logs. *Decision Support Systems*, 134:113284, 2020.
23. D. Chapela-Campa, I. Benchekroun, O. Baron, M. Dumas, D. Krass, and A. Senderovich. Can I Trust My Simulation Model? Measuring the Quality of Business Process Simulation Models. In C. Di Francescomarino, A. Burattin, C. Janiesch, and Sh. Sadiq, editors, *Business Process Management (BPM 2023)*, volume 14159 of *Lecture Notes in Computer Science*, pages 20–37. Springer-Verlag, Berlin, 2023.
24. D. Chapela-Campa, O. Lopez-Pintado, I. Suvorau, and M. Dumas. SIMOD: Automated Discovery of Business Process Simulation Models. *SoftwareX*, 30:102157, 2025.
25. B. Estrada-Torres, M. Camargo, M. Dumas, OL. Banuelos, I. Mahdy, and M. Yerokhin. Discovering Business Process Simulation Models in the Presence of Multitasking and Availability Constraints. *Data and Knowledge Engineering*, 134:101897, 2021.
26. D. Fahland, M. Montali, J. Lebherz, W.M.P. van der Aalst, M. van Asseldonk, P. Blank, L. Bosmans, M. Brenscheidt, C. Di Ciccio, A. Delgado, D. Calegari, J. Peeperkorn, E. Verbeek, L. Vugs, and M.T. Wynn. Towards a Simple and Extensible Standard for Object-Centric Event Data (OCED): Core Model, Design Space, and Lessons Learned. *Computing Research Repository (CoRR) in arXiv*, abs/2410.14495, 2024.

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. D. Gross and C.M. Harris. *Fundamentals of queueing theory*. Wiley, London, 1985.
29. P.J. Haas. *Stochastic Petri Nets: Modelling, Stability, Simulation*. Springer Series in Operations Research. Springer-Verlag, Berlin, 2002.
30. D.W. Kelton, R. Sadowski, and D. Sturrock. *Simulation with Arena*. McGraw-Hill, New York, 2003.
31. E.J.H. Kerckhoffs. ECMS: Its Position in the European Simulation History. In *Seminal Contributions to Modelling and Simulation, Simulation Foundations, Methods and Applications*, pages 1–5. Springer-Verlag, Berlin, 2016.
32. J. Kleijnen and W. van Groenendaal. *Simulation: A Statistical Perspective*. John Wiley and Sons, New York, 1992.
33. B. Knopp, M. Pourbafrani, and W.M.P. van der Aalst. Discovering Object-Centric Process Simulation Models. In S. Rinderle-Ma, J. Munoz-Gama, and A. Senderovich, editors, *International Conference on Process Mining (ICPM 2023)*, pages 81–88. IEEE Computer Society, 2023.
34. I. Koren, J.N. Adams, A. Berti, and W.M.P. van der Aalst. OCEL 2.0 Resources - www.ocel-standard.org. In J.M. van der Werf, C. Cabanillas, F. Leotta, and L. Genga, editors, *Doctoral Consortium and Demo Track 2023 at the International Conference on Process Mining 2023*, volume 3648 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2023.
35. A. Küsters and W.M.P. van der Aalst. OCPQ: Object-Centric Process Querying and Constraints. In J. Grabis, T. Vos, M.J. Escalona, and O. Pastor, editors, *International Conference on Research Challenges in Information Science (RCIS 2025)*, volume 547 of *Lecture Notes in Business Information Processing*, pages 383–400. Springer-Verlag, Berlin, 2025.
36. A.M. Law and D.W. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, 1982.
37. S.J.J. Leemans, W.M.P. van der Aalst, T. Brockhoff, and A. Polyvyanyy. Stochastic Process Mining: Earth Movers' Stochastic Conformance. *Information Systems*, 102:101724, 2021.
38. O. Lopez-Pintado and M. Dumas. Business Process Simulation with Differentiated Resources: Does it Make a Difference? In C. Di Ciccio, R.M. Dijkman, A. del Rio-Ortega, and S. Rinderle-Ma, editors, *International Conference on Business Process Management (BPM 2022)*, volume 13420 of *Lecture Notes in Computer Science*, pages 361—-378. Springer-Verlag, Berlin, 2022.
39. W.Z. Low, S.K.L.M. vanden Broucke, M.T. Wynn, A.H.M. ter Hofstede, J. De Weerdt, and W.M.P. van der Aalst. Revising History for Cost-Informed Process Improvement. *Computing*, 98(9):895–921, 2016.
40. A. Mandelbaum and S. Zeltyn. Erlang-R: A Time-Varying Queue with Reentrant Customers, in Support of Healthcare Staffing. *Queueing Systems*, 16(2):283–299, 2014.
41. N. Martin, B. Depaire, and A. Caris. The Use of Process Mining in Business Process Simulation Model Construction. *Business and Information Systems Engineering*, 58:73–87, 2016.
42. L. Maruster and N.R.T.P. van Beest. Redesigning Business Processes: A Methodology Based on Simulation and Process Mining Techniques. *Knowledge and Information Systems*, 21(7):267–297, 2009.
43. 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.
44. J. Nakatumba, M. Westergaard, and W.M.P. van der Aalst. Generating Event Logs with Workload-Dependent Speeds from Simulation Models. In M. Bajec and J. Eder, editors, *Advanced Information Systems Engineering Workshops*, volume 112 of *Lecture Notes in Business Information Processing*, pages 383–397. Springer-Verlag, Berlin, 2012.
45. G. Park and W.M.P. van der Aalst. Action-Oriented Process Mining: Bridging The Gap Between Insights and Actions. *Progress in Artificial Intelligence*, pages 1–22, July 2022.
46. M. Pidd. *Computer Modelling for Discrete Simulation*. John Wiley and Sons, New York, 1989.
47. 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.

48. M. Pourbafrani and W.M.P. van der Aalst. Discovering System Dynamics Simulation Models Using Process Mining. *IEEE Access*, 10:78527–78547, 2022.
49. M. Pourbafrani and W.M.P. van der Aalst. Hybrid Business Process Simulation: Updating Detailed Process Simulation Models Using High-Level Simulations. In R.S.S. Guizzardi, J. Ralyté, and X. Franch, editors, *International Conference on Research Challenges in Information Science (RCIS 2022)*, volume 446 of *Lecture Notes in Business Information Processing*, pages 177–194. Springer-Verlag, Berlin, 2022.
50. 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.
51. H.A. Reijers and W.M.P. van der Aalst. Short-Term Simulation: Bridging the Gap between Operational Control and Strategic Decision Making. In M.H. Hamza, editor, *Proceedings of the IASTED International Conference on Modelling and Simulation*, pages 417–421. IASTED/Acta Press, Anaheim, USA, 1999.
52. S.M. Ross. *A Course in Simulation*. Macmillan, New York, 1990.
53. A. Rozinat, R.S. Mans, and W.M.P. van der Aalst. Mining CPN Models: Discovering Process Models with Data from Event Logs. In K. Jensen, editor, *Proceedings of the Seventh Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2006)*, volume 579 of *DAIMI*, pages 57–76, Aarhus, Denmark, October 2006. University of Aarhus.
54. A. Rozinat, R.S. Mans, M. Song, and W.M.P. van der Aalst. Discovering Colored Petri Nets From Event Logs. *International Journal on Software Tools for Technology Transfer*, 10(1):57–74, 2008.
55. A. Rozinat, R.S. Mans, M. Song, and W.M.P. van der Aalst. Discovering Simulation Models. *Information Systems*, 34(3):305–327, 2009.
56. 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.
57. T. Srivastava, D. Sugden, and M. Kerremans. Magic Quadrant for Process Intelligence Platforms, Gartner Research Note G00840852. www.gartner.com, 2026.
58. E. Sulis, L. Genga, and N. Martin. Simulation and Process Mining: Review and Outlook. *International Journal of Data Science and Analytics*, 22(198):1–24, 2024.
59. F. Vinci, G. Park, W.M.P. van der Aalst, and M. de Leoni. Online Discovery of Simulation Models for Evolving Business Processes. In A. Senderovich, C. Cabanillas, I. Vanderfeesten, and H. Reijers, editors, *International Conference on Business Process Management (BPM 2025)*, volume 16044 of *Lecture Notes in Computer Science*, pages 451–468. Springer-Verlag, Berlin, 2025.
60. R.M. Yerkes and J.D. Dodson. The Relation of Strength of Stimulus to Rapidity of Habit-Formation. *Journal of Comparative Neurology and Psychology*, 18:459–482, 1908.