AI-augmented Business Process Management Systems: A Research Manifesto

MARLON DUMAS, University of Tartu, Estonia, and Apromore, Australia
FABIANA FOURNIER and LIOR LIMONAD, IBM Research - Haifa, Israel
ANDREA MARRELLA, Sapienza University of Rome, Italy
MARCO MONTALI, Free University of Bozen-Bolzano, Italy
JANA-REBECCA REHSE, University of Mannheim, Germany
RAFAEL ACCORSI, University of Tartu, Estonia
DIEGO CALVANESE, Free University of Bozen-Bolzano, Italy
GIUSEPPE DE GIACOMO, Sapienza University of Rome, Italy
DIRK FAHLAND, Eindhoven University of Technology, The Netherlands
AVIGDOR GAL, Technion, Israel
MARCELLO LA ROSA, The University of Melbourne, Australia, and Apromore, Australia
HAGEN VÖLZER, IBM Research - Zurich, Switzerland
INGO WEBER, TU Berlin, Germany

AI-augmented Business Process Management Systems (ABPMSs) are an emerging class of process-aware information systems, empowered by trustworthy AI technology. An ABPMS enhances the execution of business processes with the aim of making these processes more adaptable, proactive, explainable, and context-sensitive. This manifesto presents a vision for ABPMSs and discusses research challenges that need to be surmounted to realize this vision. To this end, we define the concept of ABPMS, we outline the lifecycle of processes within an ABPMS, we discuss core characteristics of an ABPMS, and we derive a set of challenges to realize systems with these characteristics.

CCS Concepts: • **Applied computing** → **Business process management**; • **Information systems** → **Middleware business process managers**; *Information systems applications*; • **Software and its engineering** → *Software development process management*; • **Computing methodologies** → **Artificial intelligence**; Modeling and simulation;

Additional Key Words and Phrases: Business process management, augmented business process, business automation, trustworthy AI, explainability

Work supported by the European Research Council via Advanced Grants PIX (834141) and WhiteMech (834228).

Authors' addresses: M. Dumas and R. Accorsi, University of Tartu, Tartu, Estonia; emails: {marlon.dumas, rafael.accorsi}@ut.ee; F. Fournier and L. Limonad, IBM Research - Haifa, Haifa, Israel; emails: {fabiana, liorli}@il.ibm.com; A. Marrella and G. De Giacomo, Sapienza University of Rome, Rome, Italy; emails: {marrella, degiacomo}@diag.uniroma1.it; M. Montali and D. Calvanese, Free University of Bozen-Bolzano, Bolzano, Italy; emails: {montali, calvanese}@inf.unibz.it; J.-R. Rehse, University of Mannheim, Mannheim, Germany; email: rehse@uni-mannheim.de; D. Fahland, Eindhoven University of Technology, Eindhoven, The Netherlands; email: d.fahland@tue.nl; A. Gal, Technion, Haifa, Israel; email: avigal@ie.technion.ac.il; M. La Rosa, The University of Melbourne, Melbourne, Australia; email: marcello.larosa@unimelb.edu.au; H. Völzer, IBM Research - Zurich, Zurich, Switzerland; email: hvo@zurich.ibm.com; I. Weber, TU Berlin, Berlin, Germany; email: ingo.weber@tu-berlin.de.

2158-656X/2023/01-ART11 $15.00
https://doi.org/10.1145/3576047

ACM Reference format:
Marlon Dumas, Fabiana Fournier, Lior Limonad, Andrea Marrella, Marco Montali, Jana-Rebecca Rehse, Rafael Accorsi, Diego Calvanese, Giuseppe De Giacomo, Dirk Fahland, Avigdor Gal, Marcello La Rosa, Hagen Völzer, and Ingo Weber. 2023. AI-augmented Business Process Management Systems: A Research Manifesto. *ACM Trans. Manag. Inform. Syst.* 14, 1, Article 11 (January 2023), 19 pages.
https://doi.org/10.1145/3576047

1 VISION AND MOTIVATION

An increased availability of business process execution data, combined with advances in **Artificial Intelligence (AI)**, has laid the ground for the emergence of information systems where the execution flows are not pre-determined, adaptations do not require explicit changes to software applications, and improvement opportunities are autonomously discovered, validated, and enabled on-the-fly. We call such systems ***AI-Augmented Business Process Management Systems*** **(ABPMSs)**.

An ABPMS is a process-aware information system that relies on trustworthy AI technology to reason and act upon data, within a set of restrictions, with the aim to continuously adapt and improve a set of business processes with respect to one or more performance indicators.

This definition takes as a starting point the postulate that an ABPMS is a type of **Information System (IS)** [93], which maintains some representation, typically digital, of a domain (i.e., a part of the world). Following Alter [2], we view an IS as a "work system whose processes and activities are devoted to [...] capturing, transmitting, storing, retrieving, manipulating, and displaying information." Specifically, we consider an ABPMS as one with the following capabilities, which we illustrate below using a running example of a cold-chain transportation dispatch process for perishable goods:

(1) It supports one or more processes, where a business process is *"a collection of tasks that are executed in a specific sequence to achieve some business goal, such as producing a service or product for customers" [96]*. This property sets an ABPMS apart from systems that use AI to support isolated tasks. For example, an **Optical Character Recognition (OCR)** software can be used to produce a digital version of a paper invoice. This may be an important step in a process, but if the software only supports an isolated task, it is not process-aware.

(2) It tracks the execution of the processes it supports, ensuring that each process conforms to a set of restrictions (e.g., policies), herein called the *process frame*. Following Minsky [65], we take as a starting point that a *"frame is a data structure for representing a stereotyped situation, like being in a certain kind of living room [capturing] several kinds of information [...] about how to use the frame [...] what one can expect to happen next [and] about what to do if these expectations are not confirmed."* In the context of an ABPMS, these "expectations" are restrictions on possible states or actions, which the ABPMS is expected to abide to. In other words, the frame defines the boundaries within which the ABPMS should operate. These boundaries may be tightly or loosely specified. For example, a Petri net specifying the prescribed behavior of a process is an example of a tightly framed execution, while a set of temporal logic constraints that must hold true during the execution of the process constitutes a more loosely framed execution. Potentially, any approach presented in [78] is a candidate for framing the process in an ABPMS.

(3) It orchestrates the activities of the processes it supports, such as starting an activity once its preconditions are fulfilled, routing, or performing activities, within the given frame. In other words, an ABPMS operates largely autonomously, within the boundaries set by the process frames (as it may manage multiple processes), while facilitating human-machine

cooperation and guidance when the restrictions are not or cannot be met. The latter property implies that an ABPMS has a mechanism to detect and respond to situations where the restrictions in a process frame cannot be met. To this end, an ABPMS may use AI technology, such as computer vision, signal processing, **Natural Language Understanding (NLU)**, or knowledge graphs, to identify process-level changes to detect and respond to such situations. In our example, a sensor in a truck indicates a cooling malfunction. The system then needs to find the nearest service station and reroute the truck there. It also needs to decide whether there is enough time to wait for the truck to be fixed or whether it needs to send a replacement truck to ensure the delivery of the goods in time. Each of these steps entails a change at the process level (insertion of activities).

(4) It uses AI technology to attain the goal of a business process within the given frame. Thus, a **Business Process Management System (BPMS)** that does not embed any AI capability is not an ABPMS. On the other hand, a BPMS should not fully realize all the characteristics and lifecycle steps described in this article to be an ABPMS, nor does it need to incorporate all the AI techniques mentioned in this article. On the contrary, any BPM system that incorporates AI technology to drive the execution of business processes fits within the definition of an ABPMS.

Embedding AI into a BPMS makes trust a crucial aspect of acceptance [25, 85]. In a traditional BPMS, each task or decision in a process is driven either by a human actor or by a software application that executes the task according to pre-determined business logic. In contrast, in an ABPMS, the system automatically decides if a task is executed at all, and if so, how it will be executed, e.g., by a human agent or by the system, without pre-defined execution scripts.

One may wonder whether an ABPMS is nothing but a general intelligence system that integrates one or more AI techniques (such as truth maintenance, knowledge graphs, machine learning, and automated reasoning). This is indeed the case, provided that the system monitors and orchestrates the execution of one or more business process(es). Thus, truth maintenance can be used to maintain and update knowledge about the process unfolding (in a broad sense), knowledge graphs can be exploited to store structural information about the objects and relations manipulated by the process, machine learning can be employed to predict and recommend what to do next, and so on and so forth. However, the realization of an ABMPS requires the development and deployment of AI technologies that take into consideration the characteristics of (business) processes, chiefly:

(1) Their discrete temporal nature: a business process consists of collections of activities, events, and decisions that transform one or more objects to attain a certain goal.
(2) The imperative for compliance, i.e., staying within the frame.
(3) The ability to interact with business managers to pursue the goals of the process at the tactical level (e.g., optimizing the process performance indicators).
(4) The ability to coordinate the work of several stakeholders at an operational level and to detect and respond to situations that potentially lead to situations outside the frame.

An orthogonal, key dimension is how "open-ended" the ABPMS must be. This depends on the degree of predictability and repetitiveness of the process at hand, which may indeed affect which augmentation techniques can be applied, and how effective they can be.

In this manifesto, we elaborate on the above definition of ABPMS by outlining a possible lifecycle of a business process that is executed, adapted, and continuously improved via AI technology. Based on this augmented lifecycle, we analyze the characteristics of an ABPMS and contrast these characteristics to those of a conventional BPMS [75].

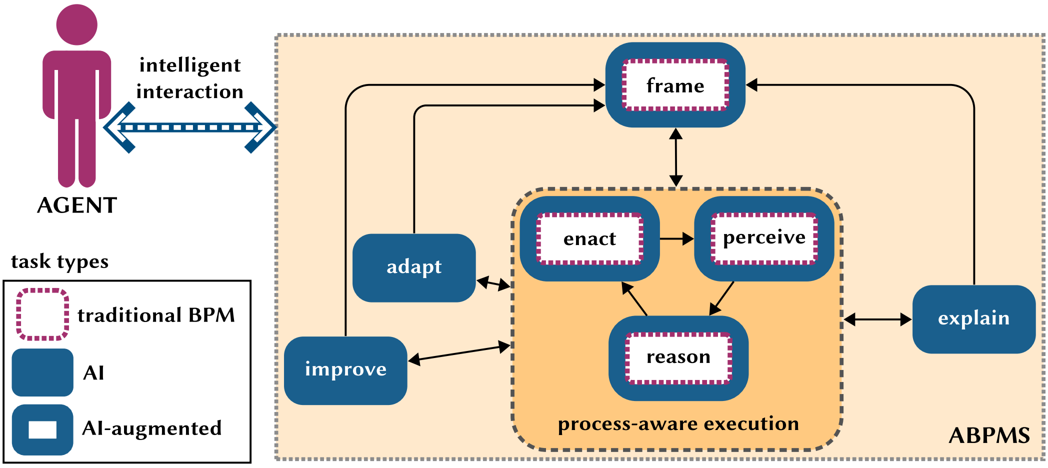

Fig. 1. The ABPMS lifecycle. Rounded rectangles represent lifecycle steps, while arrows denote sequencing.

2 AI-AUGMENTED BPMS: A LIFECYCLE

In this section, we focus on the operational lifecycle of an ABPMS, as shown in Figure 1. We describe the different steps, which in turn require the ABPMS to exhibit certain characteristics, discussed in detail in Section 3. We distinguish between two step types:

- *Basic steps* (frame, enact, perceive, reason). These are steps that can be found also in the lifecycle of a traditional BPMS [20] but that are here augmented with AI technologies.
- *Advanced steps* (explain, adapt, improve). These are specific to ABPMSs and only exist when AI is an integral part of the system.

The lifecycle involves two main actors: the ABPMS itself and one or more *agents* that interact with the ABPMS to achieve their *goals* while complying with *process constraints*. In this context, the term “agent” has to be interpreted as an umbrella term for any autonomous entity entitled to use and control the ABPMS, whether human or digital.

The lifecycle starts when an agent *frames* the ABPMS, that is, equips the ABPMS with initial constraints and goals. Constraints may incorporate different types of knowledge, such as predefined procedures, norms, regulatory constraints, commonsense rules, and best practices. The frame defines the boundaries within which the ABPMS should operate. How the frame is specified (e.g., whether it is interpreted and verified symbolically or used for rewards or other indirect compliance mechanisms) depends on the AI techniques that the ABPMS incorporates. In any case, since an ABPMS seeks to optimize a set of measurable business process goals, these constraints are likely to refer to **Key Performance Indicators (KPIs)**. Such KPIs enable the ABPMS to determine whether the goals are met or to estimate how distant it currently is from a goal state. Note that an ABPMS may have multiple competing goals and may be tasked with finding an optimal tradeoff between them.

In terms of business process specification languages, framing can employ a wide repertoire of different modeling languages, including (i) traditional graphical process modeling languages employed such as BPMN [20]; (ii) flexible process modeling languages [78], including declarative approaches such as Declare [67, 73] and DCR graphs [69] equipped with temporal logic-based semantics [14, 68]; and (iii) “hybrid” combinations of these languages (see, e.g., [16, 81, 86]).

We acknowledge that the concept of “framing,” as presented by Minsky [65], is traditionally conceived as an underlying data stereotyping mechanism or a set of facts about particular objects and event types and arranging the types into a large taxonomic hierarchy or a knowledge graph. In an ABPMS, we build on the core idea of framing as the set of articulations that accounts for process awareness in the system and sets the maximal permissive boundary for its execution. As a general concept, a frame does not have to conform to a single linguistic or symbolic formalism in its realization and may cater to a combination of element specifications such as goals to be achieved, transition rules (e.g., legal and illegal states and transitions), constraints and prohibited execution scenarios, different strategies and rewards for execution, and performance metrics. Overall, a frame is a core characteristic in ABPMSs, defined as the holistic set of epistemic articulations that ensures the system inherently retains (some degree of) process awareness in its execution cycle (namely, “process aware” execution as illustrated in Figure 1). As such, the frame lets the ABPMS determine how it can progress with its executions, whether it is just set to discover a process, to monitor a given process, to alert about process mis-conformances, or to (autonomously) drive the full operations of, say, a certain business unit. We purposely keep the concrete specification of a frame open to give the freedom of its realization to employ any viable means of articulation, ranging from anything between a rigid process model (i.e., traditional BPMS) on the one hand and a general intelligence AI that is trained to maintain process awareness on the other.

The actual process executed by the ABPMS is in general not directly obtained from the provided constraints but instead emerges from the ABPMS itself in later stages of the lifecycle. At the same time, different modes of execution for the ABPMS can be instrumented depending on how the system is framed. At one extreme of framing we have the case where an agent instructs the ABPMS with a very specific goal (such as to minimize time to delivery in our running example, neglecting cost and potential disturbances to the cold chain) and a detailed set of strict constraints yielding an imperative process (such as a BPMN definition of the dispatch process). In this latter scenario, where an imperative modeling notation is used, framing is essentially equivalent to the modeling step in the traditional BPMS lifecycle. At the other extreme of framing, we have the situation where only the goals are given, leaving the ABPMS completely free to decide how to best achieve them. The goal structure embedded in the ABPMS—externally through framing and internally through learning—will then determine its interactions with and implications on the world around it, particularly with regard to the resources it utilizes. If the freight of the cold chain truck is so valuable that the ABPMS is not allowed to take any risks, it will have the cooling repaired immediately, even though it might increase the time to delivery. If the ABPMS’s goal is to minimize time to delivery, it might notice that the truck is close enough to its destination and the ambient temperature is still acceptable, so it continues the delivery and repairs the truck afterward.

Once an initial framing is completed, the lifecycle enters its central stage, namely *process-aware execution*. This consists of the rotation between perceiving, reasoning, and enactment. Such rotation serves as the intrinsic mechanism that accommodates for process awareness, continuously tracking the business process state, and ensuring its execution respects the corresponding framing.

During this stage, the system *autonomously* executes a *process* that emerges from the ABPMS itself depending on (continuously updated) goals and constraints, as well as additional extensional and intentional information obtained by the ABPMS during execution. Internally, this stage is structured following the well-known *sense-think-act* cycle, a widely adopted paradigm to define architectures of dynamic AI systems [80].

Process-aware execution is substantiated by three main steps. First, the ABPMS *perceives* the environment and acquires new data concerning the process and the current state of affairs. Such data may be heterogeneous, with different levels of granularity and degrees of certainty. For example, the ABPMS may acquire event data directly from a workflow management system, from structured legacy systems [9], and/or from unstructured textual and visual sources [47]. Such data may be incomplete and uncertain. For example, when using a video camera to detect relevant event data, it may not be straightforward to distinguish whether a clerk is inspecting a package or loading it into a truck, and the recorded view may be partial.

Although the perceived data often refer to the extensional level, i.e., to facts that the ABPMS perceives to be true about the world, they may also pertain to the intentional level, i.e., the process enacted by the ABPMS. Examples of extensional data of the first kind are the creation of a new order by the customer or the detection of a defect in a package. Examples of intentional data of the second kind are a new inspection procedure for packages that must be acquired by the ABPMS or the refinement of a previously acquired constraint.

Transforming collected data into information, extracting relevant events, pondering about their uncertainty, and combining them with the process, constraints, and goals toward deciding which actions to take next is the purpose of the second step, namely *reasoning*. This pertains to a wide repertoire of relevant AI techniques that may be employed by the ABPMS, such as data integration, event/state recognition, knowledge (graph) construction, planning, simulation, reasoning about actions and goals, verification, estimation, and prediction of the short- and long-term effects and rewards of actions.

Next, the actual execution is carried out in the consequent, *enactment* step, where the ABPMS uses its own actuators to interact with the environment. This interaction may occur directly or indirectly. For example, the ABPMS may autonomously create and send an invoice to a customer if it has direct access to the ERP system of the organization; if it has no access, it may alternatively converse with a human operator from the sales department or with an external software agent to perform the action. The latter scenario typically characterizes physical actions to be directly performed in the world, in case the ABPMS is not equipped with physical actuators.

Once an enactment step is performed, a new perception step is typically carried out, sensing the induced effects of the action and starting a new iteration of the cycle.

At any step during the execution cycle, the ABPMS system may decide to perform one of the advanced steps: provide *explanations* about the past, current, and anticipated states of the system; *adapt* itself to new circumstances and drifts; and *improve* (e.g., optimize) its execution against its goals, the available resources, and the framing constraints. The execution of those tasks could lead the ABPMS to revise and update its internal knowledge both at the extensional level, by adding and revising information about objects, relations, facts, and events, and at the intentional one, by evolving its internal model of the process, as well as its constraints and goals. In other words, the ABPMS may autonomously *reframe* itself on the basis of the newly acquired knowledge and a corresponding adaptation, improvement, or explanation step, while respecting the previous frame (cf. Section 3.1). Importantly, reframing can also be carried out by the agents themselves, depending on the current circumstances and the continuous evolution of goals and requirements.

Implied from the notion of framing, we can thus distinguish an ABPMS from any other form of an AI-empowered system where its core intelligence mechanism is realized without having an explicit component of (process-aware) framing as a key characteristic of an ABPMS. As intelligent and accurate as such a system may be, and whether or not it exploits any of the recent advances in AI technologies (e.g., NLP, DL, RL), we consider such an AI realization to be an ABPMS only if it possesses at least the process execution steps running under the framing to govern for process awareness.

3 CHARACTERISTICS OF AN ABPMS

To realize the ABPMS lifecycle, we postulate that an ABPMS should be (1) *(framed) autonomous* to act independently and proactively, (2) *conversationally actionable* to seamlessly interact with agents whenever necessary, (3) *adaptive* to react to changes in its environment, (4) *(self-)improving* to ensure the optimal achievement of its goals, and (5) *explainable* to ensure the trust and hence the cooperation of the human agents.

3.1 (Framed) Autonomous

One characteristic that is inherent, yet implicit, in the presented lifecycle is that an ABPMS acts *autonomously* within the provided frame. This means that the system performs the lifecycle steps independently and proactively. ABPMSs can leverage AI capabilities to assess the circumstances of a specific situation and then independently select and execute the most appropriate action. This is particularly apparent for the reasoning step, where an ABPMS can draw on either symbolic or sub-symbolic AI capabilities to find its own patterns and make deductions.

The degree of autonomy depends on the framing, which determines the system's actions in two ways. First, framing acts as a maximal permissive boundary for the execution of the ABPMS. It provides the goals and constraints within which the process is executed. As stated in Section 2, the ABPMS may independently reframe itself if it acquires new knowledge during adaption, improvement, or explanation. However, the designer may restrict the ABPMS's autonomy to reframe itself. We refer to this capability as *meta-framing*. Meta-framing gives designers the option to determine which parts of the frame are modifiable and which can only be modified via human instructions. For example, an ABPMS may be allowed to accept deviations in temperature by up to 10% but has to notify a human agent in case of larger changes. Meta-framing allows designers to control the degree of autonomy of the ABPMS, from a system that acts fully autonomously to one that is never allowed to reframe itself. This facilitates a gradual buildup of trust on the part of the human agents.

Seeing the process enacted by the ABPMS as a maximally permissive behavior confined within the boundaries of the current frame is reminiscent of supervisory control [12, 98] and synthesis [23] in discrete-event systems. Our setting is arguably more complex due to the nature of the frame, which includes hard constraints, soft constraints, and different types of goals, in addition to a continuous evolution and adaptation of the frame itself.

The second implication of the framing on the degree of autonomy relates to the abilities of the ABPMS to capture its environment. The potential actions that the ABPMS may take depend on the information that is provided during framing. The more information the ABPMS has about the context of the process and its underlying constraints and assumptions, the more it is able to make its own informed decisions and act accordingly. Ideally, the ABPMS will be *adjustably autonomous*, i.e., able to recognize areas where it needs more information and then ask one of its agents to provide it.

3.2 Conversationally Actionable

An ABPMS is expected to act largely autonomously, reducing the involvement of human agents. However, those agents are still an essential part of the ABPMS lifecycle. Hence, being able to interact with humans effectively is a central characteristic of an ABPMS [70]. Depending on the level of autonomy, the ABPMS may ask for input or guidance to frame the process appropriately. In return, the human agent may ask the ABPMS for some facts. The agent might also provide the ABPMS with new insights about the execution context. These situations require direct interaction between the ABPMS and a human agent.

To keep the human effort at a minimum, we postulate that an ABPMS should be *conversationally actionable*, i.e., able to proactively communicate with human agents about process-related actions, goals, and intentions, using natural language possibly enhanced with richer interfaces. This characteristic ensures that the ABPMS seamlessly integrates into the workday of the human agents. Here, "conversational" means that the ABPMS offers interfaces for both written and spoken natural language to cater to the users' preferences and to facilitate both synchronous and asynchronous communication. In addition, it can adapt its mode of conversation to both the preferences of the user, the type of information that needs to be communicated, and the situation in which the communication happens.

"Actionable" means that the ABPMS makes concrete recommendations to the user and engages in a discussion about their benefits and drawbacks. For example, if a manager enters the office in the morning, they could verbally ask the ABPMS for a recap of the previous day's process performance. The ABPMS could then present the requested information as an interactive dashboard and recommend inspection of some erroneous cases. The manager can interact with the dashboard, ask the ABPMS for potential options to mitigate those problems, and engage in a discussion to eventually find counter-measures.

3.3 Adaptive

Business processes are often embedded in dynamic contexts characterized by uncertainty and change [4]. ABPMSs must be able to operate under these circumstances. For example, a person who was competent to perform an activity yesterday may no longer be certified for its execution today, such that the ABPMS has to reassign the activity without disrupting the process execution. A cooling truck may have broken down, such that the ABPMS has to find a substitute, or the truck has become temporarily unavailable, such that the APBMS has to redirect or slow down process flows to avoid cascading interruptions [88]. The company may acquire new knowledge, such as an expert's opinion, or new tools, such as AI technology, that enable the execution of previously unknown process steps.

An ABPMS must therefore be able to *adapt* itself to the new process context in real time. It needs to notice changes in its environment and react appropriately. In contrast to conventional BPMSs, where ad hoc changes to individual process instances may be performed only via human interventions [13], an ABPMS may trigger adaptations autonomously.

3.4 Self-improving

Whereas the adaptive nature of an ABPMS refers to its capabilities to react to external changes, an ABPMS should also proactively anticipate changes and be constantly *self-improving*. This means that an ABPMS should predict changes and be optimally prepared for those changes. For example, an ABPMS can detect the gradually increasing temperature in a cooling truck and predict that it will have to undergo maintenance operations within the next weeks. It can then schedule those operations for a day when the predicted transportation load is low and reserve capacities in other trucks to take over the concerned orders.

The objective of the ABPMS is to strike optimal tradeoffs between the competing performance measures of its process, such as operational cost and waiting times [53]. Therefore, it autonomously decides what needs to be done to achieve such tradeoffs. These decisions may affect the whole process or individual cases and be temporary or permanent. For our example, the ABPMS can either schedule the maintenance immediately or wait for a day with a low transportation load, weighing a longer delivery time and higher cost against a higher output quality.

3.5 Explainable

Acceptance of an ABPMS requires users to trust that the system will execute the process in a way that they find rational and justified. Thus, *trustworthiness* is a crucial characteristic of an ABPMS. Mayer et al. [63] define trust as "the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party." Based on this definition, we postulate that for an ABPMS to be trustworthy, it should enable users to understand its choices. This requires the ABPMS to give faithful and tactically aligned *explanations* and to allow its operators to ensure it is running all business processes as intended and preserving the ability to recover and retract from undesirable or inconsistent conditions.

Toreini et al. [89] identify four aspects of trustworthiness: **Fairness, Explainability, Auditability, and Safety (FEAS)**. Fairness technologies focus on detection or prevention of discrimination and bias in different demographics; explainability technologies focus on explaining and interpreting the outcome to the stakeholders (including end-users) in a humane manner; auditability technologies focus on enabling third parties and regulators to supervise, challenge, or monitor the operation of the model(s); safety technologies focus on ensuring the operation of the model as intended in the presence of an active or passive malicious attacker. In the scope of an ABPMS, we view explainability in its broader business sense, also governing for fairness and other regulatory concerns (hence, auditability), whereas the safety aspect is addressed in the framing step.

In light of the above, we consider process-aware explainability as an inherent property of an ABPMS. An ABPMS should be equipped with the ability to independently and continuously reason about process enactment outcomes (a "second tier" of reasoning), most often retrospectively. This includes ongoing capturing of key conditions (e.g., historical framing that reflects timely assumptions and beliefs) and the ability to draw inferential associations (dependencies) between such conditions and intermediary process execution results (i.e., reasoning and/or enactment). Such drawing helps to autonomously establish/quantify the situational validity of any derived process output. Constantly tracking such inferential associations could be used to discover invalid or insignificant results and may also be a basis for the ability to retract process executions.

4 RESEARCH CHALLENGES AND OPPORTUNITIES

Enabling ABPMSs requires overcoming a handful of research challenges. Some of the more prominently identified research directions are the following.

4.1 Situation-aware Explainability

The vision about ABPMSs, as opposed to BPMSs, is that such a system possesses intrinsic explainability capabilities (cf. the "explain" step in Figure 1). "Explaining" is used here as a general term, which could unfold itself into a variety of articulated queries, such as "What are the reasons for performing task T? Why was decision X taken? When was it concluded that the business process goal had been attained?" As an inherent step, explainability should be realized to have its output presented to the agent, so that the agent is able to understand and accordingly act upon it and the evolving situation inducing it (e.g., retract to a point in time when the inconsistency could be eliminated). Existing techniques lack in many cases clear semantics of their reasoning and remain detached from a broader process context. For example, consider a patient already awaiting admission to a surgery while a new lab result arrives with some new metrics that reverse the need to operate. Will the system be able to actively stop the process and retract to the point at which surgery may need to be re-assessed?

Providing explanations on why a machine state is reached has been recently investigated, motivated by the opaqueness of so-called "black-box" approaches [39, 40]. Finding an appropriate explanation is not easy, because this requires understanding the context and providing a description of causality and consequences of a given fact. Frequently, explanations cannot be derived from "local" inference (e.g., current undergoing task or decision in a business process) but require explicit bookkeeping that links the current execution frame to its corresponding chain of causal inferential associations, allowing traceability and trackability to the (root) causes that contributed to its deduction. Maintaining this type of historical inference associations is necessary to provide explanations at the process ("global") level. Jan et al. [42] motivate the need for process-aware explanations. Here, process-aware explainability refers to taking advantage of the knowledge of the business process definitions and full runtime process traces for better explanations.

The embedding of AI in the context of BP enables considering richer, time-related, contextual information that relates to application of each AI capability during some process execution. While time might be the most prominent factor affecting a system's state, other factors including location (spatial) and objects grouping (claim process behaves differently for different types of customers), and combinations of these, are also of importance and should be taken into consideration when analyzing different outcomes and behaviors.

Our proposed research direction goes beyond taking into account the sequencing imposed by the process and includes a much broader reasoning context in the lifecycle of an ABPMS that is responsible for capturing the relevant situational factors beyond the mere local context. Hence, our desire in an ABPMS is to enable a broad, *situation-aware explainability*, one that is able to capture traces of reasoning highlights (i.e., combination of "reasoning" parts and corresponding framing) and is able to reflect potential inferential associations that go beyond the local reasoning context. Further to this, situation-aware explainability also entails ongoing tracking of inferential associations between subsequent enactments, as a basis to gaining confidence in its ability to retract upon inconsistencies and handle a large variety of situations.

As with the emergence of Reason Maintenance Systems (e.g., [19]) in relation to knowledge bases back in the 1970s, the consideration of AI in the context of processes presents a unique opportunity for further developments of new explainability techniques and process execution management infrastructure. Complemented by contemporary advances in causal reasoning (e.g., [71, 72]), such infrastructure could be developed to determine plausible justifications for process decisions and results (intermediary or eventual) in real time to allow for valid establishment of reasoning in retrospect. Specifically, *event knowledge graphs* [24], which encode behavioral and causal interdependencies of objects and actors over time in the context of process flows and process knowledge, allow to symbolically represent situations of all kinds for *situation-aware reasoning*. Such techniques may be used to facilitate the (automatic or by humans) tracking of execution consistency for better understanding of process flows and process outcomes and to drive ongoing process improvements (at either design or retraction at runtime).

4.2 Augmented Process Automation

An ABPMS should be able to achieve more complex automation than a traditional BPMS by using AI to minimize human-dependent training and supporting human users to execute complex tasks that entail decision-making. A partial solution to this issue is provided by **Robotic Process Automation (RPA)** technology [92]. RPA is intended to operate on the **user interface (UI)** of software applications by creating software robots that automate mouse and keyboard interactions to perform repetitive tasks previously done by humans on the UI. This minimizes human error due to mental lapses resulting from boredom or exhaustion. A critical component to the usage of RPA is related to the identification of opportunities for automation to add the RPA technology in the right place and maximize its potential [51].

Even though the research in RPA shows promising methods for assessing automation opportunities [1, 6, 50], in practice tasks amenable to be automated are identified manually by subject matter experts. In addition, identifying automatic tasks only solves part of the problem. In fact, many tasks do not reflect routine work, and different users may adopt varying work practices for the same task, making it difficult to realize full process automation, which is not always feasible from a technical point of view.

Differently from RPA, an ABPMS does not aim at fully replacing the human users working on a process, but rather at leveraging AI techniques to augment the humans' capabilities and stimulate their tacit knowledge for executing process tasks in a trustworthy manner. A way to address this challenge is to rely on the concept of *Hybrid Process Intelligence* [90], which postulates that an AI component must interact with human users as a "learning apprentice," adapting itself to the users' work practices (and not vice versa). To realize this vision, a two-sided interaction is required between the ABPMS and the human user. On the one hand, an ABPMS must be capable of recognizing previously unnoticed situations and seamlessly escalate key decisions about individual cases or groups of cases to human decision makers, providing them with contextualized information and scenario assessments to support effective decision-making. On the other hand, human users' can always override a decision taken by the ABPMS to prevent it from making mistakes. Notably, the corrective actions performed by the human should be leveraged by the ABPMS to enhance its internal learning of the process and improve its autonomy.

4.3 Automated Process Adaptation

The management of processes in dynamic contexts requires that an ABPMS provide real-time monitoring and automated adaptation features during process execution to adapt processes to exceptions, exogenous events, and any contextual change that may happen at runtime, by preserving their structure and minimizing any human intervention.

Initial research work addressing the need of adaptive process management can be traced back to the late 1990s and early 2000s [10, 11, 21, 22, 37, 43, 55]. Abstracting from the specific techniques and implementations, a common behavioral pattern can be identified. At design time, a process designer identifies possible exceptions that may occur and specifies suitable exception handlers to catch and fix the exceptions. At runtime, errors, constraint violations, and other events might interrupt the process flow. The exception is detected and thrown, and an exception handler is invoked to catch the exception.

A similar principle has been applied in YAWL [87], a workflow language supported by a traditional BPMS implementation, where for each exception that can be anticipated, it is possible to define an exception handling process, named *exlet*, which includes a number of exception handling primitives (for removing, suspending, continuing, completing, failing, and restarting a work item/case) and one or more compensatory processes in the form of *worklets* (i.e., self-contained YAWL specifications executed as a replacement for a work item or as compensatory processes). Exlets are linked to specifications by defining specific rules in the shape of RippleDown Rules specified as "if condition, then conclusion," where the condition defines the exception triggering condition, and the conclusion defines the exlet. Notably, the BPMS supporting YAWL can be potentially extended to incorporate more advanced adaptation logic toward addressing the requirements of an ABPMS.

In an attempt to increase the level of user support, the ADEPT system and its evolutions [49, 76, 77] rely on the possibility of performing structural ad hoc changes over process instances at runtime. Structural adaptation techniques have been systematized through the identification of adaptation patterns [94], i.e., predefined change operations for adding, deleting, or replacing process activities. Similarly, semi-automated approaches using case-based reasoning have been proposed in [95]. They exploit available knowledge about previously performed changes so that users can retrieve and apply it to adapt a process.

While in the previous work the degree of automation is generally limited to manual ad hoc changes performed by experienced users, an ABPMS is in need of techniques that go beyond hard-coded solutions that put all the burden on IT professionals, who often lack the needed knowledge to model all possible contingencies at the outset, or this knowledge can become obsolete as process instances are executed and evolve, by making useless their initial effort.

In this direction, a number of techniques from the field of AI have been applied to BPMSs with the aim of increasing the degree of automated process adaptation [56, 57]. In [31, 32, 58, 59], if a task failure occurs at runtime and leads to a process goal violation, a new complete process definition that complies with the goal is generated relying on a partial-order AI planner. As a side effect, this often significantly modifies the assignment of tasks to process participants. The work [7] proposes a goal-driven approach to adapt processes to runtime context changes. Process and context changes that prevent goal achievement are specified at design time, and recovery strategies are built at runtime through an adaptation mechanism based on service composition via AI planning. Recently, the SmartPM approach [60, 61] has been proposed. It relies on a planning-based mechanism that requires no predefined handler to build on the fly the recovery procedure required to adapt a running process instance. Specifically, adaptation in SmartPM is seen as reducing the gap between the expected reality, i.e., the (idealized) model of reality that reflects the intended outcome of the task execution, and the physical reality, i.e., the real world with the actual values of conditions and outcomes. If a recovery procedure is needed during process execution (this happens if the two realities are different from each other), SmartPM invokes an AI planner to build a recovery procedure as a plan, which can thereby resolve exceptions that were not designed into the original process.

SmartPM, which can be considered as an implementation of the digital twin paradigm [82], provides an important demonstration of how automated adaption can be incorporated into an ABPMS. However, SmartPM requires the design-time definition of the family of tasks involved in a process, annotated with pre- and post-conditions expressed in terms of data objects and attributes, and when changes to them become relevant. This may specifically require situation-aware descriptions of tasks in the context of complete, partially specified, under-specified process fragments of desired or forbidden behavior over the history and expected future evolution of all involved data objects and actors [28, 29, 38]. Conversely, in an ABPMS, extensions are needed to incorporate new data types and tasks during runtime and to permit richer kinds of data and knowledge in the planning, which can be a daunting task. To ease the achievement of this task, a possible solution is to employ specific design patterns and heuristics at design time [62, 81] that may help to improve the system's ability to deal with unanticipated situations at runtime.

4.4 Perspective Agility

In contrast to BPMSs, where the process model is predefined, ABPMSs should support processes with unknown (or incomplete) structure at design time [28, 29, 38] that may emerge at runtime. To realize this vision, an ABPMS must support a mixture of formalisms (e.g., imperative, declarative, goal-oriented, and actor routines and habits) and interrelated artifact types (e.g., temporal constraints, goal specifications, flowcharts, and data objects) and behaviors to achieve agile process executions.

The challenge of multi-perspective support of processes thereby has to be addressed in two aspects: (1) integrating various formalisms, conceptualizations, and symbolic and sub-symbolic representations of processes and execution flows and (2) integrating behavioral characteristics and constraints of entities interacting in the shared process context.

The first aspect was tackled in the BPM literature mainly in the modelling [48], monitoring [41], and mining [36] phases, even if always targeting the integration of imperative and declarative formalisms only. The concept of perspective agility as a runtime issue is (partly) addressed by the philosophy behind the **Case Management and Modeling Notation (CMMN)** standard [97].

CMMN provides the possibility for ad hoc sequencing that emerges by the case. Another alternative to CMMN could be to rely on the **belief-desire-intention (BDI)** agent perspective [34]. In BDI, an agent is described by its beliefs (i.e., the information an agent has about itself and its environment), its desires (i.e., motivations of the agent that drive its course of action), and its derived intentions (i.e., the short-term plans that the agent wants to execute). In an ABPMS, a process can be executed considering the current goal and the context to determine the next step of the process, and the agent can be seen as an assistant of the user who is responsible for driving a task through the process, whose real structure is discovered only during the process enactment. CMMN and BDI are only two examples. We envision that drawing explicit linkages among formalisms (e.g., through ontology mappings) can help synchronize the specification of different formalisms in a process-oriented way, toward multi-perspective support during process execution.

The second aspect has initially been researched to study the interplay of processes and data objects for automated reasoning. To this end, data-centric process models have been investigated, enriching persistent data models with process-aware dynamics [8, 18]. In addition, declarative and procedural process modeling formalisms have been extended with data inspection and manipulation capabilities. This led to a flourishing series of approaches, ranging from case variables and decisions [5, 15, 30] to process networks that co-evolve multiple objects involved in one-to-many and many-to-many relationships [3, 26, 27, 35, 64, 74], which called for novel process mining techniques [54, 91]. Recently, also the behavioral influence of actors and resources on the processes they perform has been investigated from the angle of performance. This research led to integrating queueing models and process models [84] and the detection of complex performance patterns [17, 44]. This also allowed to increase accuracy in process prediction [45, 83], inferring otherwise unobservable behavior [27] and detecting emergent system-level phenomena [88]. Further, integrating explicit behavioral descriptions of process executions and actors allows to detect complex task execution patterns describing organizational routines and individual habits that involve multiple actors and process executions that evolve over time [46]. All these approaches have in common that they integrate two or more "behavioral dimensions" or explicitly consider a multitude of observations in one dimension, e.g., all cases passing through a queue. What emerges is the need to explicitly distinguish different kinds of objects as well as their "trajectories" over time. We envision that a simple but flexibly extensible graph-based model (e.g., [24]) is capable of encoding the concepts and phenomena so far studied separately in a uniform format.

Any solution to perspective agility thereby fundamentally has to offer the capability to describe and allow inference of the process in any chosen (combination of) perspectives.

4.5 Actionable Conversations

A major challenge to realize the vision behind an ABPMS consists in developing solutions (possibly with a low learning curve) that facilitate the interaction between an ABPMS and the users. Nowadays, there is a strong industry trend toward automating processes using reactive conversational agents (e.g., chatbots) [52, 66], which rely on simple scripts that drive users through a series of predefined questions. Conversely, an ABPMS should provide an interface, which relies on AI to create dynamic conversations, that not only responds to user queries and performs actions on their behalf but also initiates conversations with users in order to inform them of the process progression, alert them of relevant process changes (e.g., changes in demand distribution, customer behavior, or resource performance and availability), and make recommendations for interventions in order to improve performance with respect to relevant performance targets.

To develop such proactive actionable conversations with an ABPMS, we posit that an integrated usage of Natural Language Processing and Machine Learning techniques should be adopted, utilizing their ability to infer conversation meaning from relevant vocabulary and remember previous conversations with users, enabling tailored responses for recurrent users. In this direction, some recent papers touched on the importance of natural language understanding in business process automation. Among them, in [33], the authors present an approach toward less reactive chatbot development that relies on natural language understanding and generative machine learning models, emphasizing the importance of "synthesis from examples." While a significant overhead and domain knowledge are necessary to implement this approach, its predictable and controllable behavior makes it a robust candidate to be adopted for enterprise chatbots. In [79], the authors present a multi-agent framework to develop a conversational assistant supporting the capability of autonomously executing tasks in a business process, although many relevant challenges still need to be tackled, such as scalability, agent overlap (i.e., as the number of agents in the framework increases, some agent functionality and knowledge may overlap), and access control (some agents must not be accessible to specific users).

5 CONCLUSION

The introduction of AI technology into BPMSs creates a range of opportunities to exploit automation in business processes to make them more resourceful, with minimal, yet effective, engagement with human agents during their execution. These opportunities require a significant shift in the way the BPMS operates and interacts with its operators (both human and digital agents). While traditional BPMSs encode pre-defined flows and rules, an ABPMS is able to reason about the current state of the process (or across several processes) to determine a course of action that improves the performance of the process. To fully exploit this capability, the ABPMS needs a degree of autonomy. Naturally, this autonomy needs to be framed by operational assumptions, goals, and environmental constraints. Also, ABPMSs need to engage conversationally with human agents, they need to explain their actions, and they need to recommend adaptations or improvements in the way the process is performed.

This manifesto outlined a number of research challenges that need to be overcome to realize systems that exhibit these characteristics. We cannot bring a proper closure to this manifesto as it stands in contrast with its purpose of being a trigger for future research. The postulated characteristics of an ABPMS, and hence the challenges, are not exhaustive and others may be added. They enable a modular progression from BPMS to ABPMS prioritized by the designer's choice. They should rather be grasped as a first step toward shaping the notion of ABPMSs. Similarly, the stated challenges should be seen as an open call to design, develop, and validate methods and techniques that contribute to achieve the vision of an ABPMS as a system that exploits the capabilities of AI technology to support the continuous improvement of business processes during and as part of their execution.

REFERENCES

[1] Simone Agostinelli, Marco Lupia, Andrea Marrella, and Massimo Mecella. 2020. Automated generation of executable RPA scripts from user interface logs. In *Business Process Management: Blockchain and Robotic Process Automation Forum.* Springer International Publishing, Cham, 116–131.

[2] Steven Alter. 2008. Defining information systems as work systems: Implications for the IS field. *European Journal of Information Systems* 17, 5 (2008), 448–469.

[3] Alessandro Artale, Alisa Kovtunova, Marco Montali, and Wil M. P. van der Aalst. 2019. Modeling and reasoning over declarative data-aware processes with object-centric behavioral constraints. In *Proceedings of the 17th International Conference on Business Process Management (BPM'19) (LNCS)*, Thomas T. Hildebrandt, Boudewijn F. van Dongen, Maximilian Röglinger, and Jan Mendling (Eds.), Vol. 11675. Springer, Cham, 139–156.

[4] Abayomi Baiyere, Hannu Salmela, and Tommi Tapanainen. 2020. Digital transformation and the new logics of business process management. *European Journal of Information Systems* 29, 3 (2020), 238–259.

[5] Kimon Batoulis, Stephan Haarmann, and Mathias Weske. 2017. Various notions of soundness for decision-aware business processes. In *Proceedings of the 36th International Conference on Conceptual Modeling (ER'17) (Lecture Notes in Computer Science)*, Heinrich C. Mayr, Giancarlo Guizzardi, Hui Ma, and Oscar Pastor (Eds.), Vol. 10650. Springer, Cham, 403–418.

[6] Antonio Bosco, Adriano Augusto, Marlon Dumas, Marcello La Rosa, and Giancarlo Fortino. 2019. Discovering automatable routines from user interaction logs. In *International Conference on Business Process Management*. Springer, Cham, 144–162.

[7] Antonio Bucchiarone, Marco Pistore, Heorhi Raik, and Raman Kazhamiakin. 2011. Adaptation of service-based business processes by context-aware replanning. In *Proceedings of the 4th International Conference on Service-Oriented Computing and Applications (SOCA'11)*. IEEE, 1–8.

[8] Diego Calvanese, Giuseppe De Giacomo, and Marco Montali. 2013. Foundations of data-aware process analysis: A database theory perspective. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'13)*, Richard Hull and Wenfei Fan (Eds.). ACM, New York, NY, 1–12.

[9] Diego Calvanese, Tahir Emre Kalayci, Marco Montali, and Ario Santoso. 2017. OBDA for log extraction in process mining. In *Proceedings of the 13th International Summer School on Semantic Interoperability on the Web (RW'17) (LNCS)*, Giovambattista Ianni, Domenico Lembo, Leopoldo E. Bertossi, Wolfgang Faber, Birte Glimm, Georg Gottlob, and Steffen Staab (Eds.), Vol. 10370. Springer, Cham, 292–345.

[10] Fabio Casati, Stefano Ceri, Stefano Paraboschi, and Guiseppe Pozzi. 1999. Specification and implementation of exceptions in workflow management systems. *ACM Transactions on Database Systems (TODS)* 24, 3 (1999), 405–451.

[11] Fabio Casati and Gianpaolo Cugola. 2001. Error handling in process support systems. In *Advances in Exception Handling Techniques*. Springer-Verlag, Berlin, 251–270.

[12] Christos G. Cassandras and Stephane Lafortune. 2008. *Introduction to Discrete Event Systems* (2nd ed.). Springer, Boston, MA.

[13] Peter Dadam and Manfred Reichert. 2009. The ADEPT project: A decade of research and development for robust and flexible process support. *Computer Science-Research and Development* 23, 2 (2009), 81–97.

[14] Giuseppe De Giacomo, Riccardo De Masellis, and Marco Montali. 2014. Reasoning on LTL on finite traces: Insensitivity to infiniteness. *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)* 28, 1 (2014), 1027–1033.

[15] Massimiliano de Leoni, Paolo Felli, and Marco Montali. 2021. Integrating BPMN and DMN: Modeling and analysis. *J. Data Semant.* 10, 1–2 (2021), 165–188.

[16] Johannes De Smedt, Jochen De Weerdt, Jan Vanthienen, and Geert Poels. 2016. Mixed-paradigm process modeling with intertwined state spaces. *Bus. Inf. Syst. Eng.* 58, 1 (2016), 19–29.

[17] Vadim Denisov, Dirk Fahland, and Wil M. P. van der Aalst. 2018. Unbiased, fine-grained description of processes performance from event data. In *Business Process Management - 16th International Conference (BPM'18), Proceedings (Lecture Notes in Computer Science)*, Mathias Weske, Marco Montali, Ingo Weber, and Jan vom Brocke (Eds.), Vol. 11080. Springer, 139–157. https://doi.org/10.1007/978-3-319-98648-7_9

[18] Alin Deutsch, Richard Hull, Yuliang Li, and Victor Vianu. 2018. Automatic verification of database-centric systems. *ACM SIGLOG News* 5, 2 (2018), 37–56.

[19] Jon Doyle. 1979. A truth maintenance system. *Artificial Intelligence* 12, 3 (1979), 231–272.

[20] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A. Reijers. 2018. *Fundamentals of Business Process Management, Second Edition*. Springer, Berlin.

[21] Johann Eder and Walter Liebhart. 1995. The Workflow Activity Model WAMO. In *Proceedings of the 3rd International Conference on Cooperative Information Systems (CoopIS'95)*, Steve Laufmann, Stefano Spaccapietra, and Toshio Yokoi (Eds.). dblp, 87–98.

[22] J. Eder and W. Liebhart. 1996. Workflow recovery. In *Proceedings of the 1st IFCIS International Conference on Cooperative Information Systems*. IEEE Computer Society, Los Alamitos, CA, 124–134. https://doi.org/10.1109/COOPIS.1996.555004

[23] Rüdiger Ehlers, Stéphane Lafortune, Stavros Tripakis, and Moshe Vardi. 2017. Supervisory control and reactive synthesis: A comparative introduction. *Discrete Event Dynamic Systems* 27, 2 (2017), 209–260.

[24] Stefan Esser and Dirk Fahland. 2021. Multi-dimensional event data in graph databases. *J. Data Semant.* 10, 1–2 (2021), 109–141. https://doi.org/10.1007/s13740-021-00122-1

[25] European Commission. 2020. *White Paper on Artificial Intelligence: A European Approach to Excellence and Trust*. White Paper COM(2020) 65 final. European Commission, Brussels. https://ec.europa.eu/info/files/white-paper-artificial-intelligence-european-approach-excellence-and-trust_en.

[26] Dirk Fahland. 2019. Describing behavior of processes with many-to-many interactions. In *Application and Theory of Petri Nets and Concurrency - 40th International Conference (PETRI NETS'19), Proceedings (Lecture Notes in Computer Science)*, Susanna Donatelli and Stefan Haar (Eds.), Vol. 11522. Springer, 3–24. https://doi.org/10.1007/978-3-030-21571-2_1

[27] Dirk Fahland, Vadim Denisov, and Wil. M. P. van der Aalst. 2022. Inferring unobserved events in systems with shared resources and queues. *Fundamenta Informaticae* 183, 3–4 (Jan. 2022), 203–242. https://doi.org/10.3233/FI-2021-2087

[28] Dirk Fahland and Robert Prüfer. 2012. Data and abstraction for scenario-based modeling with petri nets. In *Application and Theory of Petri Nets - 33rd International Conference (PETRI NETS'12). Proceedings (Lecture Notes in Computer Science)*, Serge Haddad and Lucia Pomello (Eds.), Vol. 7347. Springer, 168–187. https://doi.org/10.1007/978-3-642-31131-4_10

[29] Dirk Fahland and Matthias Weidlich. 2010. Scenario-based process modeling with Greta. In *Proceedings of the Business Process Management 2010 Demonstration Track (CEUR Workshop Proceedings)*, Marcello La Rosa (Ed.), Vol. 615. CEUR-WS.org, 1–6. http://ceur-ws.org/Vol-615/paper16.pdf.

[30] Paolo Felli, Massimiliano de Leoni, and Marco Montali. 2021. Soundness verification of data-aware process models with variable-to-variable conditions. *Fundam. Informaticae* 182, 1 (2021), 1–29.

[31] Hugo M. Ferreira and Diogo R. Ferreira. 2006. An integrated life cycle for workflow management based on learning and planning. *International Journal on Cooperative Information Systems* 15, 4 (2006), 485–505.

[32] M. Gajewski, M. Momotko, H. Meyer, H. Schuschel, and M. Weske. 2005. Dynamic failure recovery of generated workflows. In *16th International Workshop on Database and Expert Systems Applications (DEXA'05)*. IEEE Computer Society Press, 982–986. https://doi.org/10.1109/DEXA.2005.78

[33] Boris Galitsky. 2019. *Developing Enterprise Chatbots: Learning Linguistic Structures*. Springer International Publishing, Cham.

[34] Michael Georgeff, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. 1999. The belief-desire-intention model of agency. In *Intelligent Agents V: Agents Theories, Architectures, and Languages*, Jörg P. Müller, Anand S. Rao, and Munindar P. Singh (Eds.). Springer, Berlin, 1–10.

[35] Silvio Ghilardi, Alessandro Gianola, Marco Montali, and Andrey Rivkin. 2020. Petri Nets with parameterised data. In *Business Process Management*, Dirk Fahland, Chiara Ghidini, Jörg Becker, and Marlon Dumas (Eds.). Springer International Publishing, Cham, 55–74.

[36] Antonella Guzzo, Mikel Joaristi, Antonino Rullo, and Edoardo Serra. 2021. A multi-perspective approach for the analysis of complex business processes behavior. *Expert Systems with Applications* 177 (2021), 114934.

[37] Claus Hagen and Gustavo Alonso. 2000. Exception handling in workflow management systems. *IEEE Transactions on Software Engineering* 26, 10 (2000), 943–958.

[38] Marcin Hewelt and Mathias Weske. 2016. A hybrid approach for flexible case modeling and execution. In *Business Process Management Forum (BPM'16), Proceedings (Lecture Notes in Business Information Processing)*, Marcello La Rosa, Peter Loos, and Oscar Pastor (Eds.), Vol. 260. Springer, 38–54. https://doi.org/10.1007/978-3-319-45468-9_3

[39] Andreas Holzinger. 2018. From machine learning to explainable AI. In *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA'18)*. IEEE, 55–66. https://doi.org/10.1109/DISA.2018.8490530

[40] Andreas Holzinger, Georg Langs, Helmut Denk, Kurt Zatloukal, and Heimo Müller. 2019. Causability and explainability of artificial intelligence in medicine. *WIREs Data Mining and Knowledge Discovery* 9, 4 (2019), e1312. https://doi.org/10.1002/widm.1312 arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1312.

[41] Amin Jalali and Paul Johannesson. 2013. Multi-perspective business process monitoring. In *Enterprise, Business-Process and Information Systems Modeling*, Selmin Nurcan, Henderik A. Proper, Pnina Soffer, John Krogstie, Rainer Schmidt, Terry Halpin, and Ilia Bider (Eds.). Springer, Berlin, 199–213.

[42] Steve T. K. Jan, Vatche Ishakian, and Vinod Muthusamy. 2020. AI trust in business processes: The need for process-aware explanations. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 8 (Apr. 2020), 13403–13404.

[43] Mark Klein and Chrysanthos Dellarocas. 2000. A knowledge-based approach to handling exceptions in workflow systems. *Computer Supported Cooperative Work (CSCW)* 9, 3–4 (2000), 399–412.

[44] Eva L. Klijn and Dirk Fahland. 2019. Performance mining for batch processing using the performance spectrum. In *Business Process Management Workshops (BPM'19) International Workshops, Revised Selected Papers (Lecture Notes in Business Information Processing)*, Chiara Di Francescomarino, Remco M. Dijkman, and Uwe Zdun (Eds.), Vol. 362. Springer, 172–185. https://doi.org/10.1007/978-3-030-37453-2_15

[45] Eva L. Klijn and Dirk Fahland. 2020. Identifying and reducing errors in remaining time prediction due to inter-case dynamics. In *2nd International Conference on Process Mining (ICPM'20)*, Boudewijn F. van Dongen, Marco Montali, and Moe Thandar Wynn (Eds.). IEEE, 25–32. https://doi.org/10.1109/ICPM49681.2020.00015

[46] Eva L. Klijn, Felix Mannhardt, and Dirk Fahland. 2021. Classifying and detecting task executions and routines in processes using event graphs. In *Business Process Management Forum (BPM'21), Proceedings (Lecture Notes in Business Information Processing)*, Artem Polyvyanyy, Moe Thandar Wynn, Amy Van Looy, and Manfred Reichert (Eds.), Vol. 427. Springer, 212–229. https://doi.org/10.1007/978-3-030-85440-9_13

[47] Sönke Knoch, Shreeraman Ponpathirkoottam, and Tim Schwartz. 2020. Video-to-model: Unsupervised trace extraction from videos for process discovery and conformance checking in manual assembly. In *Proceedings of the 18th*

International Conference on Business Process Management (BPM'20) (Lecture Notes in Computer Science), Dirk Fahland, Chiara Ghidini, Jörg Becker, and Marlon Dumas (Eds.), Vol. 12168. Springer, 291–308.

[48] Marcello La Rosa, Marlon Dumas, Arthur H. M. Ter Hofstede, and Jan Mendling. 2011. Configurable multi-perspective business process models. *Information Systems* 36, 2 (2011), 313–340.

[49] Andreas Lanz, Manfred Reichert, and Peter Dadam. 2011. Robust and flexible error handling in the aristaflow BPM suite. In *Information Systems Evolution*, Pnina Soffer and Erik Proper (Eds.). Springer, Berlin, 174–189.

[50] Volodymyr Leno, Marlon Dumas, Marcello La Rosa, Fabrizio Maria Maggi, and Artem Polyvyanyy. 2022. Discovering data transfer routines from user interaction logs. *Information Systems* 107 (2022), 101916.

[51] Volodymyr Leno, Artem Polyvyanyy, Marlon Dumas, Marcello La Rosa, and Fabrizio Maggi. 2021. Robotic process mining: Vision and challenges. *Business & Information Systems Engineering* 63, 3 (2021), 301–314.

[52] Anselmo López, Josep Sànchez-Ferreres, Josep Carmona, and Lluís Padró. 2019. From process models to chatbots. In *Advanced Information Systems Engineering*, Paolo Giorgini and Barbara Weber (Eds.). Springer International Publishing, Cham, 383–398.

[53] Orlenys López-Pintado, Marlon Dumas, Maksym Yerokhin, and Fabrizio Maria Maggi. 2021. Silhouetting the cost-time front: Multi-objective resource optimization in business processes. In *Business Process Management Forum*, Artem Polyvyanyy, Moe Thandar Wynn, Amy Van Looy, and Manfred Reichert (Eds.). Springer International Publishing, Cham, 92–108.

[54] Xixi Lu, Marijn Nagelkerke, Dennis van de Wiel, and Dirk Fahland. 2015. Discovering interacting artifacts from ERP systems. *IEEE Trans. Serv. Comput.* 8, 6 (2015), 861–873.

[55] Zongwei Luo, Amit Sheth, Krys Kochut, and John Miller. 2000. Exception handling in workflow systems. *Applied Intelligence* 13, 2 (2000), 125–147.

[56] Andrea Marrella. 2018. What automated planning can do for business process management. In *Business Process Management Workshops*, Ernest Teniente and Matthias Weidlich (Eds.). Springer International Publishing, Cham, 7–19.

[57] Andrea Marrella. 2019. Automated planning for business process management. *Journal on Data Semantics* 8, 2 (2019), 79–98.

[58] Andrea Marrella and Yves Lespérance. 2017. A planning approach to the automated synthesis of template-based process models. *Serv. Oriented Comput. Appl.* 11, 4 (2017), 367–392.

[59] A. Marrella and Y. Lespérance. 2013. Synthesizing a library of process templates through partial-order planning algorithms. *Lecture Notes in Business Information Processing* 147 (2013), 277–291.

[60] Andrea Marrella, Massimo Mecella, and Sebastian Sardina. 2016. Intelligent process adaptation in the smartpm system. *ACM Trans. Intell. Syst. Technol.* 8, 2, Article 25 (nov 2016), 43 pages.

[61] Andrea Marrella, Massimo Mecella, and Sebastian Sardiña. 2018. Supporting adaptiveness of cyber-physical processes through action-based formalisms. *AI Commun.* 31, 1 (2018), 47–74.

[62] Assaf Marron, Lior Limonad, Sarah Pollack, and David Harel. 2020. *Expecting the Unexpected: Developing Autonomous-System Design Principles for Reacting to Unpredicted Events and Conditions.* ACM, 167–173.

[63] Roger C. Mayer, James H. Davis, and F. David Schoorman. 1995. An integrative model of organizational trust. *Academy of Management Review* 20, 3 (1995), 709–734.

[64] Andreas Meyer, Luise Pufahl, Kimon Batoulis, Dirk Fahland, and Mathias Weske. 2015. Automating data exchange in process choreographies. *Inf. Syst.* 53 (2015), 296–329. https://doi.org/10.1016/j.is.2015.03.008

[65] M. L. Minsky. 1975. A framework for representing knowledge. In *The Psychology of Computer Vision*, P. Winston (Ed.). McGraw-Hill, New York, NY, 211–277.

[66] Alena Moiseeva, Dietrich Trautmann, Michael Heimann, and Hinrich Schütze. 2020. Multipurpose intelligent process automation via conversational assistant. In *Intelligent Process Automation Workshop at AAAI* (2020).

[67] Marco Montali, Maja Pesic, Wil M. P. van der Aalst, Federico Chesani, Paola Mello, and Sergio Storari. 2010. Declarative specification and verification of service choreographiess. *ACM Trans. Web* 4, 1 (2010), 3:1–3:62.

[68] Raghava Rao Mukkamala and Thomas T. Hildebrandt. 2010. From dynamic condition response structures to Büchi automata. In *Proceedings of the 4th IEEE International Symposium on Theoretical Aspects of Software Engineering (TASE'10)*, Jing Liu, Doron A. Peled, Bow-Yaw Wang, and Farn Wang (Eds.). IEEE Computer Society, 187–190.

[69] Raghava Rao Mukkamala, Thomas T. Hildebrandt, and Tijs Slaats. 2013. Towards trustworthy adaptive case management with dynamic condition response graphs. In *Proceedings of the 17th IEEE International Enterprise Distributed Object Computing Conference (EDOC'13)*, Dragan Gasevic, Marek Hatala, Hamid R. Motahari Nezhad, and Manfred Reichert (Eds.). IEEE Computer Society, 127–136. https://doi.org/10.1109/EDOC.2013.22

[70] Vinod Muthusamy, Merve Unuvar, Hagen Völzer, and Justin D. Weisz. 2020. Do's and Don'ts for Human and Digital Worker Integration. (2020). arXiv:cs.AI/2010.07738

[71] Judea Pearl. 2016 - 2016. *Causal Inference in Statistics : A Primer.* Wiley, Chichester, West Sussex.

[72] Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect* (1st ed.). Basic Books, Inc.

[73] Maja Pesic, Helen Schonenberg, and Wil M. P. van der Aalst. 2007. DECLARE: Full support for loosely-structured processes. In *Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC'07).* IEEE Computer Society, 287–300. https://doi.org/10.1109/EDOC.2007.14

[74] Artem Polyvyanyy, Jan Martijn E. M. van der Werf, Sietse Overbeek, and Rick Brouwers. 2019. Information systems modeling: Language, verification, and tool support. In *Proceedings of the 31st International Conference on Advanced Information Systems Engineering (CAiSE'19) (Lecture Notes in Computer Science)*, Paolo Giorgini and Barbara Weber (Eds.), Vol. 11483. Springer, 194–212.

[75] Shaya Pourmirza, Sander Peters, Remco M. Dijkman, and Paul Grefen. 2017. A systematic literature review on the architecture of business process management systems. *Inf. Syst.* 66 (2017), 43–58.

[76] Manfred Reichert and Peter Dadam. 1998. ADEPTflex – Supporting dynamic changes of workflows without losing control. *Journal of Intelligent Information Systems* 10, 2 (1998), 93–129. https://doi.org/10.1023/A:1008604709862

[77] Manfred Reichert, Stefanie Rinderle, and Peter Dadam. 2003. ADEPT workflow management system. In *Business Process Management*, Wil M. P. van der Aalst and Mathias Weske (Eds.). Springer, Berlin, 370–379.

[78] Manfred Reichert and Barbara Weber. 2012. *Enabling Flexibility in Process-aware Information Systems - Challenges, Methods, Technologies.* Springer, Berlin. https://doi.org/10.1007/978-3-642-30409-5

[79] Yara Rizk, Abhishek Bhandwalder, Scott Boag, Tathagata Chakraborti, Vatche Isahagian, Yasaman Khazaeni, Falk Pollock, and Merve Unuvar. 2020. A unified conversational assistant framework for business process automation. In *Intelligent Process Automation Workshop at AAAI* (2020).

[80] Stuart Russell and Peter Norvig. 2016. *Artificial Intelligence: A Modern Approach.* Pearson Education Limited.

[81] Shazia W. Sadiq, Wasim Sadiq, and Maria E. Orlowska. 2001. Pockets of flexibility in workflow specification. In *Proceedings of the 20th International Conference on Conceptual Modeling (ER'01) (Lecture Notes in Computer Science)*, Hideko S. Kunii, Sushil Jajodia, and Arne Sølvberg (Eds.), Vol. 2224. Springer, 513–526.

[82] Concetta Semeraro, Mario Lezoche, Hervé Panetto, and Michele Dassisti. 2021. Digital twin paradigm: A systematic literature review. *Computers in Industry* 130 (2021), 103469.

[83] Arik Senderovich, J. Christopher Beck, Avigdor Gal, and Matthias Weidlich. 2019. Congestion graphs for automated time predictions. In *The 33rd AAAI Conference on Artificial Intelligence (AAAI'19), The 31st Innovative Applications of Artificial Intelligence Conference (IAAI'19), The 9th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI'19).* AAAI Press, 4854–4861. https://doi.org/10.1609/aaai.v33i01.33014854

[84] Arik Senderovich, Matthias Weidlich, Liron Yedidsion, Avigdor Gal, Avishai Mandelbaum, Sarah Kadish, and Craig A. Bunnell. 2016. Conformance checking and performance improvement in scheduled processes: A queueing-network perspective. *Inf. Syst.* 62 (2016), 185–206. https://doi.org/10.1016/j.is.2016.01.002

[85] Keng Siau and Weiyu Wang. 2018. Building trust in artificial intelligence, machine learning, and robotics. *Cutter Business Technology Journal* 31, 2 (2018), 47–53.

[86] Tijs Slaats, Dennis M. M. Schunselaar, Fabrizio Maria Maggi, and Hajo A. Reijers. 2016. The semantics of hybrid process models. In *Proceedings of On the Move to Meaningful Internet Systems (OTM'16) (Lecture Notes in Computer Science)*, Christophe Debruyne, Hervé Panetto, Robert Meersman, Tharam S. Dillon, Eva Kühn, Declan O'Sullivan, and Claudio Agostino Ardagna (Eds.), Vol. 10033. Springer - Verlag, 531–551.

[87] Arthur H. M. ter Hofstede, Wil M. P. van der Aalst, Michael Adams, and Nick Russell. 2009. *Modern Business Process Automation: YAWL and Its Support Environment.* Springer Science & Business Media, Berlin.

[88] Zahra Toosinezhad, Dirk Fahland, Özge Köroglu, and Wil M. P. van der Aalst. 2020. Detecting system-level behavior leading to dynamic bottlenecks. In *2nd International Conference on Process Mining (ICPM'20)*, Boudewijn F. van Dongen, Marco Montali, and Moe Thandar Wynn (Eds.). IEEE, 17–24. https://doi.org/10.1109/ICPM49681.2020.00014

[89] Ehsan Toreini, Mhairi Aitken, Kovila Coopamootoo, Karen Elliott, Carlos Gonzalez Zelaya, and Aad van Moorsel. 2020. The relationship between trust in AI and trustworthy machine learning technologies. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT*'20).* Association for Computing Machinery, New York, NY, 272–283.

[90] Wil M. P. van der Aalst. 2021. Hybrid intelligence: To automate or not to automate, that is the question. *International Journal of Information Systems and Project Management* 9, 2 (2021), 5–20.

[91] Wil M. P. van der Aalst. 2019. Object-centric process mining: Dealing with divergence and convergence in event data. In *Proceedings of the 17th International Conference on Software Engineering and Formal Methods (SEFM'19) (Lecture Notes in Computer Science)*, Peter Csaba Ölveczky and Gwen Salaün (Eds.), Vol. 11724. Springer, 3–25.

[92] Wil M. P. van der Aalst, Martin Bichler, and Armin Heinzl. 2018. Robotic process automation. *Business & Information Systems Engineering* 60, 4 (2018), 269–272.

[93] Yair Wand and Ron Weber. 1995. On the deep structure of information systems. *Information Systems Journal* 5, 3 (1995), 203–223.

[94] Barbara Weber, Manfred Reichert, and Stefanie Rinderle. 2008. Change patterns and change support features - Enhancing flexibility in process-aware information systems. *Data & Knowledge Engineering* 66, 3 (2008), 438–466.

[95] Barbara Weber, Werner Wild, and Ruth Breu. 2004. CBRFlow: Enabling adaptive workflow management through conversational case-based reasoning. In *Advances in Case-Based Reasoning*, Peter Funk and Pedro A. González Calero (Eds.). Springer, Berlin, 434–448.

[96] Mathias Weske. 2012. *Business Process Management: Concepts, Languages, Architectures* (2nd ed.). Springer, Berlin.

[97] Markus Wiemuth, Denise Junger, M. A. Leitritz, Juliane Neumann, Thomas Neumuth, and Oliver Burgert. 2017. Application fields for the new object management group (OMG) standards case management model and notation (CMMN) and decision management notation (DMN) in the perioperative field. *International Journal of Computer Assisted Radiology and Surgery* 12, 8 (2017), 1439–1449.

[98] W. Murray Wonham and Kai Cai. 2019. *Supervisory Control of Discrete-Event Systems*. Springer, Cham.

Received 3 February 2022; revised 4 November 2022; accepted 16 November 2022