Report from Dagstuhl Seminar 25192

AUTOBIZ: Pushing the Boundaries of AI-Driven Process Execution and Adaptation

Giuseppe De Giacomo[*1], **Marlon Dumas**[*2], **Fabiana Fournier**[*3], **Timotheus Kampik**[*4], **and Lior Limonad**[*5]

1 **University of Oxford, GB.** giuseppe.degiacomo@cs.ox.ac.uk
2 **University of Tartu, EE.** marlon.dumas@ut.ee
3 **IBM Research Israel – Haifa, IL.** fabiana@il.ibm.com
4 **SAP Berlin, DE & Umeå University, SE.** timotheus.kampik@sap.com
5 **IBM Research Israel – Haifa, IL.** liorli@il.ibm.com

Abstract

Advances in AI are enabling the shift toward Autonomous Business Processes (ABPs), where systems not only suggest actions but also take proactive steps within defined constraints. This concept was introduced in the AI-Augmented Business Process Management Systems (ABPMSs) manifesto, which outlines their lifecycle, features, and research challenges. The "AutoBiz" 25192 Dagstuhl Seminar brought together experts from AI and BPM to collaborate on advancing this vision. The seminar's main goal was to define a research agenda for the realization of ABP systems.

Seminar May 4–9, 2025 – https://www.dagstuhl.de/25192
2012 ACM Subject Classification Computing methodologies → Artificial intelligence; Information systems → Information systems applications; Applied computing → Business process management; Computer systems organization → Self-organizing autonomic computing; General and reference → Surveys and overviews
Keywords and phrases AutoBiz, Artificial Intelligence, Business Process Management, Autonomous Business Processes, Dagstuhl Seminar
Digital Object Identifier 10.4230/DagRep.15.5.21

1 Executive Summary

Fabiana Fournier (IBM Research Israel – Haifa, IL, fabiana@il.ibm.com)

Advances in AI make it possible to push the boundaries of automation into the realm of **Autonomous Business Processes (ABPs)**. In ABPs, AI-based systems not only recommend predefined interventions, as in prescriptive process execution, but they also proactively trigger interventions to respond to unforeseen changes within user-defined constraints. The initial vision towards ABPs was recently introduced in the "*AI-Augmented Business Process Management Systems: A Research Manifesto*" paper. This manifesto coined the concept of **AI-Augmented Business Process Management Systems (ABPMSs)**, their lifecycle, core characteristics, and the research challenges they present. An ABPMS enhances the execution of business processes with the aim of making these processes more: Adaptable, proactive, explainable, and context-sensitive.

[*] Editor / Organizer

AUTOBIZ: Pushing the Boundaries of AI-Driven Process Execution and Adaptation, *Dagstuhl Reports*, Vol. 15, Issue 5, pp. 21–63
Editors: Giuseppe De Giacomo, Marlon Dumas, Fabiana Fournier, Timotheus Kampik, and Lior Limonad
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The most advanced form of ABPs is manifested by **ABP systems**. This Dagstuhl Seminar brought together leading academic and industrial researchers from the AI and BPM communities to collaboratively advance the vision of ABP systems and address the challenges outlined in the manifesto. These include framed autonomy, situation-aware explainability, automated process adaptation, and actionable conversations. The seminar was structured into corresponding working groups, each focusing on one of these aspects.

The discussions in these groups led to the development of four dedicated publications, presented at the PMAI workshop at ECAI 2025. These publications build upon the original manifesto by elaborating on the concrete challenges and technical foundations required to realize different characteristics of ABPMSs. The seminar thus marks a significant step toward a comprehensive research agenda for Agentic BPM systems—systems that integrate autonomous agents capable of reasoning, learning, and acting within framed constraints.

The results of this seminar, as elaborated in this report, form a bold call to action for realizing the vision of Agentic BPM systems. These systems embody the architectural principles and core ideas initially sketched in the manifesto and now further developed through collaborative research and discussion.

For further details, readers are encouraged to consult the full report.

2 Table of Contents

Executive Summary

Overview of Talks

Working Groups

AUTOBIZ Dissemination

3 Overview of Talks

3.1 AI-Augmented BPM – The manifesto and its key characteristics

Marlon Dumas (University of Tartu, EE, marlon.dumas@ut.ee)

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 talk presents a vision for ABPMSs and discusses research challenges that ought 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.

Details are given in [1] and [2].

References

1 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. Ai-augmented business process management systems: A research manifesto. *ACM Trans. Manag. Inf. Syst.*, 14(1):11:1–11:19, 2023.

2 David Chapela-Campa and Marlon Dumas. From process mining to augmented process execution. *Softw. Syst. Model.*, 22(6):1977–1986, 2023.

3.2 Framed Autonomy in AI-Augmented Business Process Management Systems

Marco Montali (Free University of Bozen-Bolzano, IT, montali@inf.unibz.it)

Process framing is about defining suitable boundaries of execution within which AI agents and humans can operate and collaborate to enact one or more work processes. The talk summarized a long-standing line of research aiming at providing an explicitly formal description of the frames, which can be effectively employed in computational terms, e.g., for reasoning, verification, execution, and analysis. Temporal logics on finite traces, together with their automata-theoretic characterization, are the basis of this approach, which has been widely explored within process science to provide a declarative way for process specification and management (cf., in particular the Declare language). The talk started by showing how this approach can be effectively used for process framing using declarative process mining tools for frame elicitation and deviation analysis. It then innovatively expanded the approach to deal with:

- Frames that can be broken – providing anticipatory monitoring techniques to promptly detect deviations at runtime
- Uncertain frames, where “a portion” of traces may violate some constraints
- Data-aware frames, dealing with data attributes or more complex objects and their mutual relationships

All these settings come with suitable extensions of temporal logics on finite traces, with corresponding foundational and practical results on computation. Finally, how agents operating within the frame can employ the repertoire of techniques to operate and enact processes, discussing several open challenges within and across artificial intelligence and process science.

3.3 Uncertainty Quantification and its role in AI-Augmented BPM

Niek Tax (Meta – London, GB, niek@meta.com)

Uncertainty Quantification (UQ) plays a pivotal role in enhancing the reliability, safety, and adaptability of AI-augmented Business Process Management (BPM) systems. This presentation explores the connection between UQ and several key BPM tasks. Various forms of UQ are discussed, including probability calibration and the modeling of the full posterior predictive distribution (beyond point estimates) and its applications in risk-aware decision making, ensuring fairness, and enabling adaptability through guided exploration (e.g., active learning). The talk also presents recent advances in UQ from the Meta Central Applied Science team, including:

InfoShap [1] A SHAP-based method for explaining uncertainty estimates.
AdaptiveWeightSampling [2] A theoretically grounded active learning method that scale to industrial applications.
TCE [3] A calibration error metric suitable for imbalanced domains.
MCE [4] A metric to quantify the extent to which predictions are simultaneously calibrated across multiple subpopulations (i.e., *multicalibrated* [5]).

References

1 D. Watson, J. O'Hara, N. Tax, R. Mudd, and I. Guy. Explaining predictive uncertainty with information theoretic shapley values. In *Advances in Neural Information Processing Systems*, volume 36, pages 7330–7350, 2023.
2 Daniel Haimovich, Dima Karamshuk, Fridolin Linder, Niek Tax, and Milan Vojnovic. On the convergence of loss and uncertainty-based active learning algorithms. *Advances in Neural Information Processing Systems*, 37:122770–122810, 2024.
3 Takuo Matsubara, Niek Tax, Richard Mudd, and Ido Guy. Tce: A test-based approach to measuring calibration error. In *Uncertainty in Artificial Intelligence*, pages 1390–1400. PMLR, 2023.
4 Ido Guy, Daniel Haimovich, Fridolin Linder, Nastaran Okati, Lorenzo Perini, Niek Tax, and Mark Tygert. Measuring multi-calibration. *arXiv preprint arXiv:2506.11251*, 2025.
5 Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.

3.4 AI-augmented Process Mining

Stefanie Rinderle-Ma (TU Munich, DE, stefanie.rinderle-ma@tum.de)

Process mining and automation have emerged as technological megatrends, largely driven by recent advances in Artificial Intelligence (AI). Among these, generative AI and particularly large Language Models (LLMs) hold significant promise for transforming process discovery from textual data and enabling the creation of process models by domain experts in conversation with a chatbot. Furthermore, process technologies facilitate the collection of contextualized data, especially event log data augmented with IoT data, which can be leveraged to optimize the prediction of process behaviour by, e.g., including sensor data, targeting, for example, enhanced compliance checks.

The presentation introduces the concepts of conversational process mining and redesign, showcases the collection of contextualized data, and outlines key research directions.

3.5 Causal business processes: A new paradigm for agentic observability

Lior Limonad (IBM Research Israel – Haifa, IL, liorli@il.ibm.com)

Joint work of Lior Limonad, Fabiana Fournier

"A rooster's crow does not cause the sun to rise, even though it always precedes it." (The book of Why, Mackenzie & Pearl, 2018). This quote captures a common fallacy – post hoc ergo propter hoc – our tendency to infer causation from mere sequences: after this, therefore because of this.

In the presentation, I argued that time-correlated events do not necessarily imply a causal execution relationship. As an example, we presented a simple mortgage application process in which the acceptance decision is followed by informing the customer and archiving the application. While conventional process mining might depict the latter two activities as a direct sequence, this does not imply that informing the customer causes archiving, or vice versa.

Understanding true causal execution dependencies is essential for meaningful process improvement. It helps avoid costly empirical interventions by identifying which outcomes stem from specific changes. Our presented method introduces a novel technique for analyzing activity execution times in business processes, based on logs. We extend recent causal discovery methods – originally designed for continuous variables – and adopt them to timestamped events, while aligning with realistic assumptions about process execution. This enables a new paradigm: **Causal Business Processes**.

The talk presented our foundational work on discovering causal execution dependencies from process event logs and modeling them in a unified framework. This approach augments traditional process mining by enabling predictive insights about the effect of interventions, critical for informed process optimization.

Further details about this work is available in [1] and [2].

References

1 F. Fournier, L. Limonad, I. Skarbovsky, and Y. David. The why in business processes: Discovery of causal execution dependencies. *Künstliche Intelligenz*, 1 2025.

2 Yuval David, Fabiana Fournier, Lior Limonad, and Inna Skarbovsky. The why in business processes: Unification of causal process models. In *BPM Forum in BPM Conference (to appear)*, 9 2025.

3.6 Generative AI for process explainability

Fabiana Fournier (IBM Research Israel –Haifa, - IL, fabiana@il.ibm.com)

Joint work of Fabiana Fournier, Lior Limonad

This talk tackles the interpretability aspect of SAX explanations and explore the question on how effectively large language models (LLMs) can explain outcomes and decisions in business processes (e.g., loan rejection, parking fines) as perceived by users. In the SAX framework, a set of knowledge extractors analyses an input log (a historical record of process executions within an organization) and generates key perspectives – or “views” – of the process. These three views are then combined into a prompt for an LLM, along with the user’s inquiry (e.g., why my loan application was rejected?). The LLM uses this input to generate an explanation tailored to the user’s specific question. To assess this, we conducted a user study with 50 participants with the goal to assess both perceptions of fidelity and interpretability, while also considering the roles of trust and curiosity as experienced by the users. Our results show that adding different perspectives does, in fact, improve the perceived fidelity of the generated explanations. However, our study concluded that adding knowledge views improves perceived fidelity, but such improvement can actually come at the cost of perceived interpretability. Once we had developed a scale to assess the quality of explanations generated by LLMs, we began exploring if we can leverage the developed scale for LLM selection and refinement. To address this challenge, we launched another study – this time focused on evaluating how our scale could be used to quantify user perceptions and enable a meaningful comparison between different LLMs. The experiment centered on a tax refund process and included 128 participants, who were asked to rate explanations for various tax refund decisions using our evaluation scale.

Further details about this work is available in [1] and [2].

References

1 Lior Limonad, Fabiana Fournier, Hadar Mulian, George Manias, Spiros Borotis, and Danai Kyrkou. Selecting the right llm for egov explanations, 2025.

2 Dirk Fahland, Fabiana Fournier, Lior Limonad, Inna Skarbovsky, and Ava J. E. Swevels. How well can large language models explain business processes as perceived by users? *Data & Knowledge Engineering*, 157:102416, 2 2025.

3.7 AI-Assisted Prescriptive Business Process Monitoring

Andreas Metzger (University of Duisburg-Essen, DE, andreas.metzger@paluno.uni-due.de)

Main reference Andreas Metzger, Tristan Kley, Aristide Rothweiler, Klaus Pohl: "Automatically reconciling the trade-off between prediction accuracy and earliness in prescriptive business process monitoring", Inf. Syst., Vol. 118, p. 102254, 2023.
URL http://dx.doi.org/10.1016/J.IS.2023.102254

Prescriptive business process monitoring aims to guide process managers on adapting processes to avoid negative outcomes.

A key challenge is to balance prediction accuracy with tardiness: earlier predictions provide more time for adaptations, but are often less reliable.

This talk explores existing methods for managing this trade-off and compares their performance by using real-world datasets. By evaluating their cost savings, this talk identifies factors influencing effectiveness and provides practical recommendations for selecting appropriate approaches.

Based on these insights, the talk explores directions & opportunities for future research, involving emerging topics, such as explainable AI and Large Language Models (LLMs).

See: `https://doi.org/10.1016/j.is.2023.102254`

3.8 Foundations of Agentic AI: GenAI meets Strategic Reasoning and Planning

Giuseppe De Giacomo (University of Oxford, GB, giuseppe.degiacomo@cs.ox.ac.uk)

We are entering an era where businesses adopt AI agents to transform operations, drive impact across functions, and accelerate value realization. AI agents are software entities with goal-directed autonomy, capable of choosing and executing actions using AI techniques, see e.g., the 2025 IBM white paper "AI agents: Opportunities, risks, and mitigations". AI agents have long been studied in Reasoning about Actions (KR), Planning (ICAPS), and Autonomous Agents (AAMAS). Recently, LLM-based agents have emerged, combining language models with agency and showing strong potential of bringing about "open-endedness" and "common sense", as suggested by John McCarthy in his 1959 paper "Programs with Common Sense". Agentic AI systems combine AI agents with tools, planners, memory, and data to pursue goals autonomously, often including hardware control. In this talk, we discuss various opportunities and challenges that Agentic AI could deliver.

4 Working Groups

Seminar facilitation methodology adhered to an adapted design-thinking style. Overall, the agenda included four major phases: talks-&-ideation, working groups, presentations, and writ-a-thon (see Figure 1).

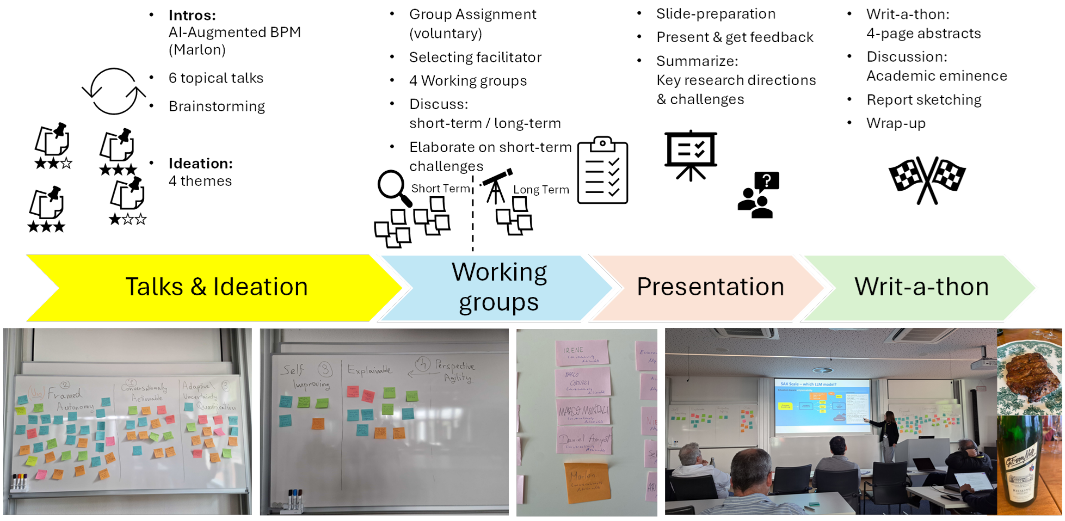


Figure 1 "Design-thinking"-like seminar structure.

4.1 Working Group on Framed Autonomy

Giuseppe De Giacomo (University of Oxford, GB, giuseppe.degiacomo@cs.ox.ac.uk)
Andrea Marrella (Sapienza Università di Roma, IT, marrella@diag.uniroma1.it)
Yves Lesperance (York University – Toronto, CA, lesperan@eecs.yorku.ca)
Andrea Matta (Politecnico di Milano, IT, andrea.matta@polimi.it)
Timotheus Kampik (SAP Berlin, DE & Umeå University, SE, timotheus.kampik@sap.com)
Diego Calvanese (Free University of Bozen-Bolzano, IT, diego.calvanese@unibz.it)

Framed autonomy refers to agents acting and interacting with maximal flexibility within potentially dynamic *frames* consisting of rules, restrictions, and regulations. With the increased deployment of AI-based technologies – recently and most notably large language models – framing the autonomy of agents that enact business processes can be expected to be a key challenge. In this working group report, we sketch problem scenarios and provide a conceptual architecture for framing autonomy in business processes. We highlight a list of practical challenges for the framing of autonomous business process behaviors and conclude with the sketch of a research roadmap.

4.1.1 Introduction

Software systems providing the operational backbone of organizations are becoming increasingly autonomous [1], partially driven by advances in deep learning-based technologies such as Large Language Models (LLMs) [2]. While in this context, *autonomy* may pertain to an overall and potentially tightly coupled system, the distributed and complex nature of large organizations requires intelligence at the level of autonomous submodules, reflecting how intelligent business decisions are made by humans. In order to deploy autonomous software agents safely and effectively, one must ensure that they comply with normative requirements, while still utilizing their substantial degrees of autonomy to accomplish their goals to the best possible extent [3].

As abstractions for managing guardrails, we propose the notion of (normative) *frames* that – in contrast to the more operational notions of declarative or procedural business processes and rules – focus only on deontic requirements of how organizations should run. Frame representation and reasoning can draw from a wealth of research on deontic logic [4, 5], temporal reasoning [6], planning [7, 8], and normative multi-agent systems [9]. We provide informal definitions of frames and position them in the context of related abstractions, and sketch scenario types describing how frames can be applied to agents enacting business processes. These partially subsymbolic AI agents must then be augmented with symbolic capabilities for synthesizing plans that guarantee frame compliance, reasoning about their own, others', and process-level goals in order to maximize objectives *within* the frames. Accordingly, on a fundamental level these agents require capabilities for plan and behavior synthesis [10, 11, 12, 13, 14], as well as for goal reasoning [15]. We also highlight a list of practical challenges that require solving to (better) utilize frames in large organizations. Considering these challenges, we outline a research roadmap for framing autonomous business process execution.

As our motivating example, we introduce a simple excerpt of a (fictional) order-to-cash process (Figure 2). A customer sends a natural language wishlist to a retailer, which then generates a symbolic order proposal from it. If all items in the proposal are available, the proposal is sent to the customer. If some items are not available, a gift is added to the proposal. The retailer chooses between chocolate or wine as the gift and sends the gift-augmented proposal to the customer. Upon receiving the answer, the retailer assesses whether the answer is positive or negative. In the former case, the retailer acknowledges the acceptance of the proposal; in the latter, the retailer generates a new order proposal. However, in total not more than three proposals will be generated for a given incoming request. The process – i.e., the *retailer* pool – is enacted by one or several agents that can autonomously generate proposals (including alternative proposals) and decide on the type of gift to add to a proposal. The objective of the process is to maximize the margins of order proposals that are accepted by customers, also considering the cost of gifts.

Some frame constraints for this example are:

- If the customer is underaged, do not add wine as the gift (i.e., a gift containing alcohol). This constraint can be easily specified using linear temporal logic and verified on finite traces (LTL_f).
- Once an item has been added to an order proposal, the item's price must not increase in a subsequent proposal. This constraint is harder to verify, as it requires reasoning over quantities over time.

Notice that we may have a single agent taking these decisions, or multiple agents, with different agents taking decisions in various moments during the execution of a single process instance. This gives rise to different scenarios that are discussed in the next section.

4.1.2 Framed Autonomy in Business Processes

Framed autonomy requires that the autonomous system operates within its current frame. Intuitively, a frame is a set of rules, restrictions, and regulations, which may evolve over time. The frame establishes the boundaries within which the system may operate with maximal flexibility, making autonomous decisions. In BPM, frames may exist – at least – on *agent type*, *process*, and *organization* level (as well as potentially across organizations).

More analytically, frames are *normative*, i.e., they specify deontic requirements to the process; in contrast, classical process specification languages, such as BPMN and DECLARE are *operational*, i.e., they specify behavior required to accomplish a business goal. In terms

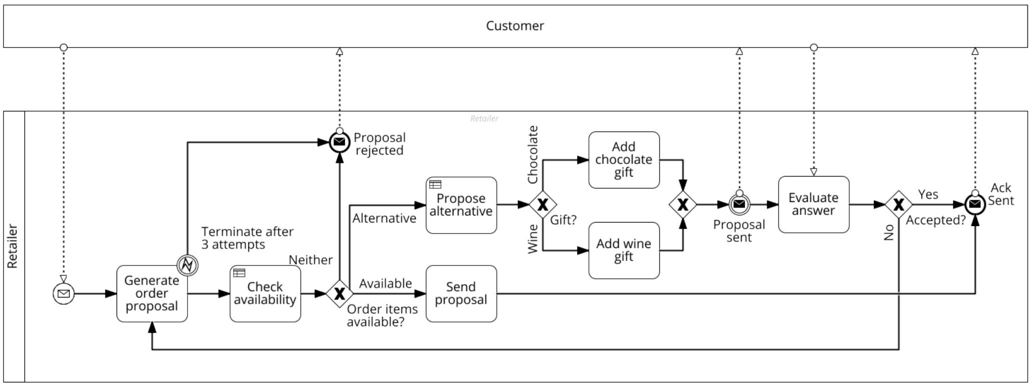


Figure 2 An excerpt of an order-to-cash process, in which a retailer turns a natural language wishlist into an order proposal.

of strategies, operational specifications require choosing a strategy to achieve a goal, while the frame requires, in principle, identifying the entire set of strategies that are compatible with the norm and then ensuring that the strategy chosen for the goal is one of them.

Notice that sometimes the operational specifications have been called *frames* as well [1]. Indeed, they can be considered a sort of operational frame. In this document, however, we our focus of "frames" is on the normative specification. When we need to distinguish, we call the two frames *normative frame* and *operational frame*, respectively.

Observe that if there are no choices to be made (no autonomous decision-makers), then the normative frame is just an additional condition over the operational frame; but if decision-making is possible then the operational frame requires finding a strategy to satisfy the objective, whereas the normative frame requires choosing a strategy that remains within what is allowed (with respect to the frame).

Strategies for achieving goals under framed autonomy are associated with decision-makers, including software agents. This gives rise to several problem setups, reflecting centralized as well as distributed intelligence.

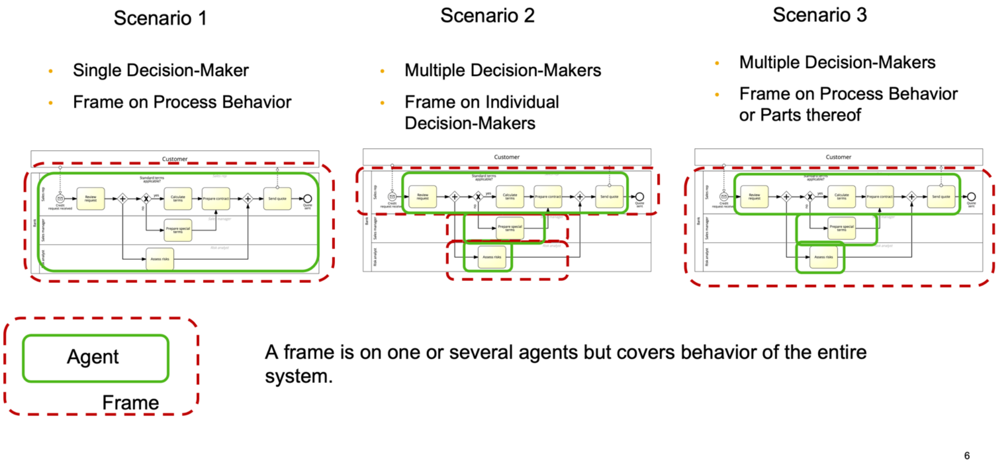
Centralized intelligence. We consider the "AI agents" as a single entity that orchestrates the *process* that is executed in a mutually fully observable and coordinated manner. The *environment* may be stochastic and not fully observable. The frame is over the process. The single entity may have active or passive responsibility for the frame. If we have multiple agents we may break down the problem into several of the above scenarios.

Distributed intelligence. We consider AI agents as distributed entities that enact the process as resources. This has wide-ranging implications:

- A resource has partial observability of what other resources do.
- Coordination may be effortful and not always possible.
- Agents may have individual goals that may not be consistent with process-level goals.
- We can frame resources or the entire process.
- We need to assign responsibility to individual agents or groups thereof, and there may be strategic interactions affecting responsibility.

From these problem setups, we can derive three different blueprint scenarios for framed autonomy in business processes (Figure 3):

1. We have a single decision-maker and place a frame on process behavior.
2. We have multiple decision-makers and place frames on individual decision-makers.
3. We have multiple decision-makers and place frame(s) on process behavior or parts thereof.



Scenario 1

- Single Decision-Maker
- Frame on Process Behavior

Scenario 2

- Multiple Decision-Makers
- Frame on Individual Decision-Makers

Scenario 3

- Multiple Decision-Makers
- Frame on Process Behavior or Parts thereof

A frame is on one or several agents but covers behavior of the entire system.

Figure 3 Different scenarios of framed autonomy in business processes.

In practice, there may be additional variance to the scenarios. For example, normative frames may be partially represented within operational process specifications, restricting overall agent autonomy. An example is a purchasing process where purchase orders can only be created and paid through a central IT system that enforces normative rules, e.g. regarding four-eyes approval policies. Other parts of the global normative frame can potentially be projected to local agent-level norms. For example, overall spending limits may apply on the global level, but could be operationalized locally. Also, from a process perspective, operationalizing some frame constraints (such as the spending limits from above) may require very broad case notions; e.g., a case may be *all purchases executed by a specific business unit within a given month.*

4.1.3 Practical Challenges

Achieving framed autonomy in business processes comes with practical challenges. Below, we list (and briefly discuss) three that we consider of particular importance.

1. *What is a pragmatic notion of an agent in the context of business process execution?* Before the broad adoption of LLMs, the notion of an agent did not play a major role in the engineering of business information systems and the processes that run them. Consequently, practitioners cannot be expected to be familiar with the depth and sophistication of agent-related abstractions. To the contrary, a practitioner may consider as an agent a software tool that makes use of an LLM, without much thought about further properties. Defining a more precise and robust notion of an agent that is still intuitively understandable by business process practitioners can thus be considered a key prerequisite. See also Working Group Report 4.2.
2. *How to elicit and specify frames?* Next, approaches for eliciting and specifying frames need to be devised. This requires a meta-model for frames, and one or several specification languages. To this end, existing specification languages can be reused; potentially several languages and their underlying concepts can be combined. For example, declarative

approaches to process specification – such as DECLARE [16] and in more practical contexts business rule and query languages with temporal reasoning capabilities [17] – can be augmented with deontic notions in order to promote normativity to a first-class abstraction. For elicitation, both symbolic and subsymbolic approaches can be used and fused. LLMs can generate frames or parts thereof from natural language text, whereas rule mining approaches can be applied to infer normative constraints from the traces of well-behaved agents and multi-agent systems. See also Working Group Report 4.3.

3. *How to operationalize frames on real-world symbolic data?* Once specified, frames need to be integrated with business information systems, to ensure systems' frame-compliance during runtime. A short- to mid-term prerequisite is the operationalization of frames using technologies that do in fact run in large organizations. Here, explainability is a necessity, considering the practical intricacy of normative requirements, as well as the scale of real-world symbolic queries and data. See also Working Group Report 4.4.

4.1.4 Research Roadmap

Given our conceptual architecture of frames for autonomous business process execution, as well as the practical challenges outlined above, we define a research roadmap. The roadmap is divided into four broad categories, each of which comes with a set of simply phrased "how to?" research questions.

4.1.4.1 Frame Representation & Reasoning

The key prerequisite for applying frames to autonomous business process execution is devising ways to represent them and reason about them. Accordingly, our questions are the following:

- *How to combine declarative and procedural paradigms when specifying frames?*
- *How to decide what to model in a trace view versus what to model in a transition system (with choice points)?*
- *How to model agents and integrate agent models with process- and organization-level models?*
- *How to operationalize frames in real-world information systems?*
- *How to ensure responsibility and accountability with respect to outcomes as well as to deontic notions such as obligations, permissions, and prohibitions?*

4.1.4.2 LLMs and Framed Autonomy

The emergence of LLMs as widely applicable natural language processing tools has led to the re-emergence of agents as mainstream abstractions in information systems engineering. Accordingly, a key problem is the assurance of normative compliant LLM agent behavior (frames for LLMs), as well the application of LLMs to generate both normative frames and frame-compliant operational behavior specifications (LLMs for frames). We consider the former direction more relevant than the latter, reflecting the overall objective of frames to maximize autonomy while still ensuring compliance. However, both are covered by the questions below:

- *How to implement LLM agents that can comply with frames?*
- *How to symbolically augment LLM agents to ensure compliance with frames?*
- *How to generate frames from sources in several modalities using LLMs, as well as symbolic approaches?*
- *How to elicit goals in business processes with LLMs?*
- *How to leverage LLMs as tools that help us reason about frames?*

4.1.4.3 Goal Reasoning

While goals are central to informal definitions of business process notions, they are typically not treated as first-class abstractions: the goal of a process or instance thereof is assumed to be implicitly defined by the behavioral specification. However, considering that agents enact processes autonomously, behavioral specification alone is insufficient, as these specifications are necessarily at least partially synthesised from agent and process goals. Accordingly, a range of research questions about goal reasoning in business processes emerge, such as:

- *How to elicit goals in business processes?*
- *How to align agent-level and process-level goals?*
- *How to represent goals in business processes?*
- *How to manage goals (e.g., instantiate, drop, revise, prioritize) in business processes?*
- *How to anticipate future goals?*
- *How to maintain most options open in anticipation of future goals when strategizing and acting?*

4.1.4.4 Meta Frames and Reframing

Finally, we are interested in representing and reasoning about multiple frames, as well as as frames that change:

- *How to compose and decompose different process frames?*
- *How to adopt meta-level frames, e.g., as provided by third-party organizations?*
- *How to navigate through different frames?*
- *How to manage conflicts between frames?*
- *How to revise frames over time?*
- *How to verify that the entity revising the frame has the authority to do so?*

4.1.5 Conclusions

When autonomy is included in a business process execution system, the notion of *normative frame* becomes essential to guardrail autonomous decision-making. Normative frames have a deontic nature and are concerned with the sets of strategies that an agent can choose from while satisfying the frame. Accordingly, when goal-oriented agents synthesize their operational strategies, these strategies are implicitly mapped to those at the normative level and checked against the frame. AI agents – whether based on symbolic or subsymbolic methods – that enact business processes must be able to synthesize such strategies so that frame compliance can be guaranteed and exceptional violations can be justified.

References

1 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. Ai-augmented business process management systems: A research manifesto. *ACM Trans. Manag. Inf. Syst.*, 14(1):11:1–11:19, 2023.

2 Timotheus Kampik, Christian Warmuth, Adrian Rebmann, Ron Agam, Lukas Egger, Andreas Gerber, Johannes Hoffart, Jonas Kolk, Philipp Herzig, Gero Decker, Han van der Aa, Artem Polyvyanyy, Stefanie Rinderle-Ma, Ingo Weber, and Matthias Weidlich. Large process models: A vision for business process management in the age of generative ai. *KI - Künstliche Intelligenz*, 2024.

3 Timotheus Kampik, Adnane Mansour, Olivier Boissier, Sabrina Kirrane, Julian Padget, Terry R. Payne, Munindar P. Singh, Valentina Tamma, and Antoine Zimmermann. Governance of autonomous agents on the web: Challenges and opportunities. *ACM Trans. Internet Technol.*, 22(4), 2022.

4 Dov Gabbay, John Horty, Xavier Parent, Ron Van der Meyden, Leendert van der Torre, et al. *Handbook of Deontic Logic and Normative Systems, Volume 1*. College Publications, 2013.

5 Dov Gabbay, John Horty, Xavier Parent, Ron Van der Meyden, Leendert van der Torre, et al. *Handbook of Deontic Logic and Normative Systems, Volume 2*. College Publications, 2021.

6 Giuseppe De Giacomo and Moshe Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *IJCAI*, pages 854–860, 2013.

7 Giuseppe De Giacomo and Sasha Rubin. Automata-theoretic foundations of FOND planning for ltlf and ldlf goals. In *IJCAI*, pages 4729–4735. ijcai.org, 2018.

8 Malik Ghallab, Dana S. Nau, and Paolo Traverso. *Acting, Planning and Learning*. Cambridge University Press, 2025.

9 A. Chopra, L. van der Torre, and H. Verhagen. *Handbook of Normative Multiagent Systems*. College Publications, 2018.

10 Amir Pnueli and Roni Rosner. On the synthesis of a reactive module. In *POPL*, pages 179–190. ACM Press, 1989.

11 Bernd Finkbeiner, Felix Klein, and Niklas Metzger. Live synthesis. *Innov. Syst. Softw. Eng.*, 18(3):443–454, 2022.

12 Giuseppe De Giacomo and Moshe Y. Vardi. Synthesis for LTL and LDL on finite traces. In *IJCAI*, pages 1558–1564, 2015.

13 Shufang Zhu and Giuseppe De Giacomo. Act for your duties but maintain your rights. In Gabriele Kern-Isberner, Gerhard Lakemeyer, and Thomas Meyer, editors, *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022, Haifa, Israel, July 31 - August 5, 2022*, 2022.

14 Shufang Zhu and Giuseppe De Giacomo. Synthesis of maximally permissive strategies for LTL_f specifications. In *IJCAI*, pages 2783–2789. ijcai.org, 2022.

15 David W. Aha. Goal reasoning: Foundations, emerging applications, and prospects. *AI Mag.*, 39(2):3–24, 2018.

16 Claudio Di Ciccio and Marco Montali. Declarative process specifications: Reasoning, discovery, monitoring. In Wil M. P. van der Aalst and Josep Carmona, editors, *Process Mining Handbook*, volume 448 of *Lecture Notes in Business Information Processing*, pages 108–152. Springer, 2022.

17 Timotheus Kampik and Cem Okulmus. Expressive power and complexity results for signal, an industry-scale process query language. In Andrea Marrella, Manuel Resinas, Mieke Jans, and Michael Rosemann, editors, *Business Process Management Forum - BPM 2024 Forum, Krakow, Poland, September 1-6, 2024, Proceedings*, volume 526 of *Lecture Notes in Business Information Processing*, pages 3–19. Springer, 2024.

4.2 Working Group on Adaptive / Uncertainty Quantification

Estefania Serral (KU Leuven, BE, estefania.serralasensio@kuleuven.be)
Achiya Elyasaf (Ben-Gurion University of the Negev – Beer Sheva, IL, achiya@bgu.ac.il)
Andreas Metzger (paluno – The Ruhr Institute for Software Technology, University of Duisburg-Essen, DE, andreas.metzger@paluno.uni-due.de)
Niek Tax (Meta – London, GB, niek@meta.com)
Sebastian Sardina (RMIT University – Melbourne, AU, sebastian.sardina@rmit.edu.au)
Arik Senderovich (York University, Toronto, CA, sariks@yorku.ca)

4.2.1 Introduction

In today's dynamic digital landscape, business processes (BPs) are expected to operate autonomously and adapt their structure and behavior to evolving goals, conditions, or constraints [4]. Such systems are known as *autonomous business process systems* (ABPs), whose key capability is self-modification.

Modifications in ABPs fall into two main categories [21]:

- **Adaptation**: Short-term, instance-specific changes to handle unforeseen conditions without altering the overall process schema. These real-time adjustments ensure continuity and resilience, for example by rerouting workflows, reallocating resources, or integrating new data sources during execution.
- **Evolution**: Long-term changes to process logic or models, affecting all future instances. These deliberate updates are driven by recurring issues or strategic shifts and may involve redesigning decision logic or updating policies.

Without these capabilities, ABPs risk brittleness and reduced responsiveness. Thus, engineering self-modifying capabilities is essential for robust, flexible process management.

The importance of adaptive and evolving BPM systems has long been recognized [3, 21, 11], with prior work exploring both adaptation [16, 20, 10, 6, 23, 26] and evolution [12, 28, 27].

In this report, we provide:

- A definition of self-modification in ABPs, distinguishing adaptation and evolution.
- A structured view of the dimensions, goals, and triggers of modifications in ABPs.
- An outline of key challenges in governance, uncertainty, and continuous learning.

4.2.2 Running Example: Automated Warehouse

To illustrate self-modifying ABPs, consider an automated warehouse where fleets of robots transport shelves to human pickers across a high-throughput environment.

If a robot fails during the busy holiday season, nearby robots reroute around the blockage, and the order is reassigned. Workers receive updated instructions, ensuring minimal disruption. This short-term response (*adaptation*) resolves the immediate issue without changing the overall process model.

The system also logs the failure and, after noticing similar breakdowns after about 1,000 picks, introduces a maintenance rule requiring inspection every 900 operations. This mid-term, model-level change (*evolution*) refines policies to prevent recurrence.

Together, these examples show how ABPs span from localized adaptations to deliberate evolution, implemented with varying degrees of automation.

4.2.3 Types of Modifications in ABPs

Modifications in ABPs can be classified along several dimensions:

D1: Adaptation vs. Evolution

We introduced this central dimension already in Section 1, referring to adaptations as short-term, instance-specific modifications, and evolution as long-term, multi-instance modifications of the process logic and/or model itself [22].

The concept of adaptation in ABPs strongly connects to work in the software engineering community on self-adaptive software systems [32]. Here, the MAPE-K loop has established itself as a widely adopted conceptual model [1, 2]. This model is depicted in Figure 4.

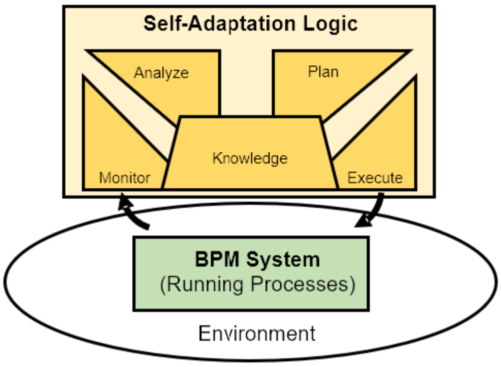


Figure 4 Conceptual model of self-adaptation.

The self-adaptation logic is structured into four main activities (MAPE) that leverage a shared knowledge base (K). The knowledge base includes adaptation goals (requirements), strategies, and rules. The four activities are:

- **Monitoring**: Observing the system and its environment via sensors.
- **Analysis**: Interpreting monitoring data to determine the need for adaptation.
- **Planning**: Deciding on adaptation actions.
- **Execution**: Implementing adaptation actions via actuators.

The concept of evolution in ABPs, in contrast, focuses on systematic, long-term modifications that affect the process model and logic across multiple future instances. Rather than responding to immediate runtime conditions, evolution is driven by aggregated insights from monitoring and analysis over time – such as recurring performance issues, changes in strategic direction, or compliance updates. These insights inform deliberate revisions of process design, typically through:

- Assessment of existing outcomes.
- Planning of structural or behavioral improvements.
- Implementation of changes.
- Validation to ensure alignment with long-term goals.

While adaptation enables resilience and responsiveness, evolution ensures strategic alignment, sustainability, and optimization.

D2: Task vs. Flow vs. Process

Another critical dimension concerns what is modified [22]:

- **Task-level changes**: Modify how individual tasks are performed or configured (e.g., adjusting duration, logic, input/output data, or resource assignments).
- **Control Flow-level changes**: Adjust control flow or routing of tasks (e.g., rerouting orders, skipping steps).
- **Process-level changes**: Alter the structure or central resources of the entire process (e.g., introducing new roles, shifting coordination logic, or replacing subsystems).

These levels often interact. For example, rerouting in the warehouse involves flow-level change, while a new rule for reassigning malfunctioning robots might affect task performance. Revising the maintenance schedule constitutes a process-level change. This dimension also affects implementation complexity: task-level changes can often be handled locally, while process-level changes typically require coordination across components.

D3: Reactive vs. Proactive

Another important distinction concerns the trigger of modifications:

- **Reactive**: Occur in response to specific events or failures, such as a robot malfunction triggering rerouting.
- **Proactive**: Initiated based on forecasts or insights from predictive analytics. Examples include predictive and prescriptive business process monitoring [17], such as scheduling preventive maintenance or restricting robot access during peak loads.

D4: Human-Driven vs. Autonomous

Changes may be:

- **Human-driven**: Users detect, decide, and implement modifications.
- **Autonomous**: The system identifies issues and enacts changes independently.

In the warehouse example, creating a maintenance rule might initially be human-driven, but an advanced ABPS could learn similar patterns and implement them autonomously.

D5: Planned vs. Emergent

- **Planned**: Result from deliberate, top-down redesign (e.g., rolling out a maintenance policy across sites).
- **Emergent**: Arise bottom-up, as patterns learned from execution lead to adaptations or eventual model evolution.

An ABPS that generalizes from local failure logs to propose global improvements exemplifies emergent capability.

Characterizing the Running Example

All key dimensions play out in the warehouse scenario:

- Rerouting around a malfunctioning robot is a short-term, instance-level *adaptation*.
- Introducing a maintenance rule is a deliberate, model-level *evolution*.
- Rerouting is *reactive* when triggered by an event but can be *proactive* if anticipated via predictive monitoring.
- Initially, the maintenance policy is *human-driven* and *planned*.
- Over time, an advanced ABPS could autonomously propose and implement similar rules, making the change *emergent*.

4.2.4 Toward Self-modifying ABPS

Today's business process systems typically operate at an augmented level of autonomy, where intelligent components assist human workers but do not independently drive process execution or change. To move toward truly autonomous business process systems (ABPS), we propose a structured roadmap of autonomy levels, inspired by the SAE J3016 standard for driving automation [24]. While Sheridan's Levels of Automation [29] offer an alternative, they focus on isolated task automation rather than holistic process-level behavior, making them less suitable for ABPS.

We define autonomy levels as follows:

- **Level 0: No Automation** – Execution and orchestration of all tasks are fully manual.
- **Level 1: Process Assistance** – The system provides recommendations or highlights anomalies to human workers (e.g., predictive monitoring [18]).
- **Level 2: Partial Autonomy** – The system independently executes isolated tasks (e.g., call routing, task assignment) within predefined boundaries, but without contextual or adaptive behavior.
- **Level 3: Contextual Autonomy** – The system autonomously performs most tasks and orchestrates flows in a context-aware manner, requiring human intervention only in exceptional cases.

While most current systems lie at Level 1 or 2, we envision Level 3 as the target state for ABPS. Achieving this level necessitates the development of self-modifying capabilities. Specifically, a Level 3 ABPS must:

1. Detect changes in the operating environment (e.g., concept drift detection [8, 14, 25]).
2. Decide whether the detected change requires adaptation or evolution.
3. Select an appropriate modification strategy based on goals, context, and system history.
4. Learn from prior adaptations and generalize successful strategies.
5. Communicate its decisions, rationale, and uncertainty to human stakeholders.

Goals and Capabilities Across Autonomy Levels

The extent and nature of these capabilities depend on the object of modification: task, flow, or process. For instance, at Level 1, task-level modifications might involve recommending better parameter configurations; at Level 3, the system may autonomously reassign or skip tasks. Similarly, flow-level autonomy ranges from highlighting bottlenecks (Level 1) to real-time rerouting (Level 3), while process-level autonomy progresses from alerting on coordination issues to full reconfiguration of goals, roles, or policies.

In the warehouse scenario, task-level changes may involve altering how picking or navigation tasks are performed; flow-level changes may include rerouting due to blocked paths; and process-level autonomy could entail adjusting global maintenance policies. Thus, progressing toward Level 3 autonomy requires integrating sensing, reasoning, learning, and communication across abstraction levels while minimizing reliance on human oversight.

4.2.5 Challenges in Enabling Autonomous Modifications

In this section, we discuss three challenges related to governance and human oversight, continual learning for adaptation management, and uncertainty quantification and communication.

Table 1 Goals and capabilities for autonomy at different levels and modification objects.

Object of Modification	Level 1: Process Assistance	Level 2: Partial Autonomy	Level 3: Contextual Autonomy
Task	Recommending task performers or configurations (e.g., duration, cost).	Automating task assignment or execution in narrow scopes.	ABPS decides and executes full task reconfiguration autonomously, with human fallback only in edge cases.
Flow	Suggesting alternative paths or detecting bottlenecks.	Automating routing based on real-time conditions or rules.	Rerouting and dynamically altering execution paths with learned policies under uncertainty.
Process	Flagging process-wide issues (e.g., coordination delays).	Automating subprocesses, such as resource pooling or exception handling.	Reconfiguring processes, modifying policies or goals, and coordinating across stakeholders with minimal human oversight.

Challenge 1: Governance, Oversight, and Human Interaction

To operate safely, ABPs must embed governance and human oversight. A central question is when to refrain from automation and return control to humans, especially in ambiguous or high-risk situations. AI planning and runtime monitoring can help define thresholds for escalation, while prediction with a reject option [7] offers approaches for "learning to defer."

Determining how users validate proposed changes requires explainable adaptation, where the system justifies its modifications. Methods from explainable AI [5] or simulations of expected outcomes can support transparency.

Table 2 Challenge mapping across modification levels.

Object of Modification	Governance & Oversight	Continuous Learning & Adaptation Mgmt.	Modeling & Uncertainty
Task	When should execution shift from autonomous to manual?	How can the system learn improved task performance?	How confident is the system in altering task strategies?
Flow	Who approves new routing decisions?	Can routing rules generalize without overfitting?	What risks arise when rerouting or skipping tasks?
Process	When must humans approve structural redesign?	How to record and evolve process-level modifications?	How is uncertainty modeled across processes?

Aligning ABP goals with those of human stakeholders remains complex. Multi-objective optimization [15] can help balance performance, cost, compliance, and satisfaction, but resolving conflicting objectives is difficult.

In fully autonomous scenarios, ABPs need internal mechanisms to evaluate whether adaptations succeed. Techniques such as anomaly detection, causal reasoning, and formal verification are promising. LLMs may also support human audits by generating decision justifications [33].

Human input for validation is often costly, motivating efficient solutions like active testing [13]. Lastly, systems must be able to assess whether they have sufficient context to act safely or should defer decisions.

Core Research Questions:

- When should control shift between autonomous processes and humans in self-modifying systems?
- How can user validation be incorporated in real-time adaptation without bottlenecking autonomy?
- How can systems optimize multiple objectives while remaining within formal and ethical constraints?
- How can ABPs evaluate the success or failure of modifications when human validation is unavailable?
- What are the data quality and coverage thresholds for safe, autonomous decision-making?

Challenge 2: Continuous Learning and Adaptation Management

A core capability of self-modifying ABPs is learning from experience to improve behavior over time. This requires continuously recording adaptations – reactive and proactive – and assessing their impact both per instance and across instances. Such meta-knowledge enables feedback loops where effective modifications are reinforced and ineffective ones discarded. AI planning and reinforcement learning [20], especially when combined with causal modeling, are promising for managing these loops.

Evaluating generalization poses another challenge. An adaptation that works once may fail under different conditions. Systems must infer when learned strategies apply and calibrate their confidence accordingly. For example, if a task is often skipped under heavy load, the ABP may learn to skip it when similar patterns arise, but must avoid overgeneralizing. Formal guarantees and validation frameworks are needed to ensure safe transfer of learned policies. This also relates to interpretability, as stakeholders expect explanations for behavior shifts.

Finally, ABPs operating over long periods or at high volume face limits of memory and context. They must build bounded knowledge representations – such as compressed summaries, predictive abstractions, or sliding windows of relevant history. Techniques like stream reasoning, process mining, and transformer-based models can help. Deciding what to remember, forget, or query is essential for scalable, continuous learning.

Core Research Questions:

- How can ABPs continuously record adaptations and assess their effectiveness over time?
- What metrics and techniques enable safe generalization of learned behavior across varying contexts?
- How can bounded knowledge representations be maintained for long-running or high-frequency processes?

Challenge 3: Modeling and Measuring Uncertainty

ABPs must handle both aleatoric (inherent randomness) and epistemic (lack of knowledge) uncertainty [9]. Aleatoric uncertainty can be modeled with probabilistic distributions over durations and outcomes, while epistemic uncertainty often requires exploration or additional sensing to reduce ambiguity. Identifying which type is present guides whether to gather more data or hedge decisions probabilistically. Probabilistic modeling, Bayesian inference, and fuzzy logic are key tools for representing uncertainty.

Quantification combines qualitative thresholds set by experts and quantitative measures like Bayesian models or ensemble predictions [16]. ABPs must integrate both, relying on statistical models when data are sufficient and heuristics when information is limited. Hybrid approaches that combine fuzzy logic and ensemble learning can improve robustness in uncertain domains.

Uncertainty modeling links closely to Challenges 1 and 2. For Challenge 1, quantifying uncertainty informs when to defer to humans [7]. For Challenge 2, accurate epistemic estimates are essential for learning adaptive policies efficiently [19, 30].

Finally, communicating uncertainty effectively is critical. Operators must grasp what the system plans to do and how confident it is. Explainable AI [31], visual confidence intervals, and natural language outputs can help convey this information, along with potential impacts on risks and service quality.

Core Research Questions:

- How can ABPs differentiate between epistemic and aleatoric uncertainty during execution?
- How can qualitative and quantitative uncertainty metrics be combined for robust decision-making?
- What are effective representations for modeling uncertainty using probabilistic or fuzzy paradigms?
- How should ABPs communicate uncertainty and associated risk to users in a transparent and actionable way?

4.2.6 Conclusion and Outlook

In this paper, we have laid the conceptual groundwork for understanding and engineering self-modifying capabilities in autonomous business process systems (ABPS). We defined what it means for an ABPS to self-modify, distinguished between the short-term reactivity of adaptation and the long-term reconfiguration of evolution, and introduced a structured framework for levels of business process autonomy. We further mapped these autonomy levels across different objects of modification – task, flow, and process – and connected them to system goals and capabilities.

As ABPS strive toward higher levels of autonomy, we identified three core research challenges that must be addressed to enable safe, intelligent, and human-aware self-modification:

1. Establishing robust governance mechanisms and human oversight.
2. Managing continuous learning to support sustainable and generalizable adaptations.
3. Modeling and communicating uncertainty in ways that foster trust and informed decision-making.

Together, these challenges highlight a critical insight: autonomy is not simply about replacing humans but about redesigning systems that can responsibly decide when to act, when to adapt, and when to defer. The path forward requires integrating techniques from AI planning, machine learning, explainable AI, causal inference, process mining, and human-computer interaction into coherent architectures that balance autonomy with accountability.

We envision ABPS of the future not as black-box automation engines but as collaborative agents capable of engaging with their environments and human counterparts in a transparent, contextual, and goal-aligned manner. To that end, we encourage the community to build benchmarks, share evaluation frameworks, and develop modular toolkits that bring us closer to truly self-modifying, trustworthy, and adaptive business processes.

References

1 Y. Brun, G. Di Marzo Serugendo, C. Gacek, H. Giese, H. Kienle, M. Litoiu, H. Müller, M. Pezzè, and M. Shaw. Engineering self-adaptive systems through feedback loops. In *Software Engineering for Self-Adaptive Systems*, pages 48–70. Springer, 2009.

2 IBM Computing. An architectural blueprint for autonomic computing. Technical Report 31, IBM White Paper, 2006.

3 P. Dadam and M. Reichert. The adept project: a decade of research and development for robust and flexible process support: challenges and achievements. *Computer Science - Research and Development*, 23:81–97, 2009.

4 M. Dumas, F. Fournier, L. Limonad, A. Marrella, M. Montali, J.-R. Rehse, R. Accorsi, D. Calvanese, G. De Giacomo, and D. Fahland. Ai-augmented business process management systems: a research manifesto. *ACM Transactions on Management Information Systems*, 14:1–19, 2023.

5 R. Dwivedi, D. Dave, H. Naik, S. Singhal, R. Omer, P. Patel, B. Qian, Z. Wen, T. Shah, G. Morgan, et al. Explainable ai (xai): Core ideas, techniques, and solutions. *ACM Computing Surveys*, 55:1–33, 2023.

6 D. Greenwood and R. Ghizzioli. Goal-oriented autonomic business process modelling and execution. In *Multiagent Systems*. IntechOpen, 2009.

7 K. Hendrickx, L. Perini, D. Van der Plas, W. Meert, and J. Davis. Machine learning with a reject option: A survey. *Machine Learning*, 113:3073–3110, 2024.

8 F. Hinder, V. Vaquet, and B. Hammer. One or two things we know about concept drift – a survey on monitoring in evolving environments. part a: detecting concept drift. *Frontiers in Artificial Intelligence*, 7:1330257, 2024.

9 E. Hüllermeier and W. Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021.

10 K. Jander, L. Braubach, A. Pokahr, W. Lamersdorf, and K.-J. Wack. Goal-oriented processes with gpmn. *International Journal on Artificial Intelligence Tools*, 20:1021–1041, 2011.

11 N. R. Jennings, P. Faratin, M. Johnson, T. J. Norman, P. O'Brien, and M. E. Wiegand. Agent-based business process management. *International Journal of Cooperative Information Systems*, 5:105–130, 1996.

12 H. Kir and N. Erdogan. A knowledge-intensive adaptive business process management framework. *Information Systems*, 95:101639, 2021.

13 J. Kossen, S. Farquhar, Y. Gal, and T. Rainforth. Active testing: Sample-efficient model evaluation. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5753–5763, 2021.

14 J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31:2346–2363, 2018.

15 R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26:369–395, 2004.

16 A. Metzger, T. Kley, and A. Palm. Triggering proactive business process adaptations via online reinforcement learning. In *Business Process Management - 18th International Conference, BPM 2020*, volume 12168 of *Lecture Notes in Computer Science*, pages 273–290. Springer, 2020.

17 A. Metzger, T. Kley, A. Rothweiler, and K. Pohl. Automatically reconciling the trade-off between prediction accuracy and earliness in prescriptive business process monitoring. *Information Systems*, 118:102254, 2023.

18 A. E. Márquez-Chamorro, M. Resinas, and A. Ruiz-Cortés. Predictive monitoring of business processes: a survey. *IEEE Transactions on Services Computing*, 11:962–977, 2017.

19 V.-L. Nguyen, M. H. Shaker, and E. Hüllermeier. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111:89–122, 2022.

20 A. Palm, A. Metzger, and K. Pohl. Online reinforcement learning for self-adaptive information systems. In *Advanced Information Systems Engineering - 32nd International Conference, CAiSE 2020*, volume 12127 of *Lecture Notes in Computer Science*, pages 169–184. Springer, 2020.

21 M. Reichert and B. Weber. *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*, volume 54. Springer, 2012.

22 H. A. Reijers. Process design and redesign. In *Process-Aware Information Systems: Bridging People and Software through Process Technology*, pages 205–234. Wiley, 2005.

23 L. Sabatucci, C. Lodato, S. Lopes, and M. Cossentino. Towards self-adaptation and evolution in business process. In *AIBP@AI*IA*, pages 1–10, 2013.

24 SAE International. Taxonomy and definitions for driving automation systems for on-road motor vehicles. Technical report, SAE Recommended Practice J3016, 2021.

25 D. M. V. Sato, S. C. De Freitas, J. P. Barddal, and E. E. Scalabrin. A survey on concept drift in process mining. *ACM Computing Surveys*, 54:1–38, 2021.

26 E. Serral, J. De Smedt, M. Snoeck, and J. Vanthienen. Context-adaptive petri nets: Supporting adaptation for the execution context. *Expert Systems with Applications*, 42:9307–9317, 2015.

27 E. Serral, P. Valderas, and V. Pelechano. Supporting runtime system evolution to adapt to user behaviour. In *Advanced Information Systems Engineering: 22nd International Conference, CAiSE 2010*, volume 6051 of *Lecture Notes in Computer Science*, pages 378–392. Springer, 2010.

28 E. Serral, P. Valderas, and V. Pelechano. Addressing the evolution of automated user behaviour patterns by runtime model interpretation. *Software & Systems Modeling*, 14:1387–1420, 2015.

29 T. B. Sheridan. Adaptive automation, level of automation, allocation authority, supervisory control, and adaptive control: Distinctions and modes of adaptation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 41:662–667, 2011.

30 A. Tharwat and W. Schenck. A survey on active learning: State-of-the-art, practical challenges and research directions. *Mathematics*, 11:820, 2023.

31 D. Watson, J. O'Hara, N. Tax, R. Mudd, and I. Guy. Explaining predictive uncertainty with information theoretic shapley values. In *Advances in Neural Information Processing Systems*, volume 36, pages 7330–7350, 2023.

32 D. Weyns. *An Introduction to Self-Adaptive Systems: A Contemporary Software Engineering Perspective*. John Wiley & Sons, 2020.

33 L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623, 2023.

4.3 Working Group on Conversational Actionability

Daniel Amyot (University of Ottawa, CA, damyot@uottawa.ca)
Marco Comuzzi (Ulsan National Institute of Science and Technology, KR, mcomuzzi@unist.ac.kr)
Marlon Dumas (University of Tartu, EE, marlon.dumas@ut.ee)
Marco Montali (Free University of Bozen-Bolzano, IT, marco.montali@unibz.it)
Irene Teinemaa (Google DeepMind – London, GB, iteinemaa@google.com)

The discussions in this working group focused on the use cases that drive the requirement of *conversational actionability* in AI-Augmented Business Process Management Systems (ABPMS), the key functions an architecture of a conversationally actionable ABPMS, and various concerns and challenges related to the realization of such systems.

4.3.1 Definitions and Use Cases

An ABPMS is *conversationally actionable* if it can interact with users or external agents using a *conversational interface* to support, trigger, or guide actions related to business processes. Depending on the user and use case, this conversational interface can support communications using natural language, textual abstractions of process related artifacts, images and graphical models, or specific agent communication protocols.

We identify four principal use cases in a conversationally actionable ABPMS:

1. *Creation.* This use case is concerned with the elicitation of knowledge leading to the creation of business process related artifacts, such as imperative or declarative business process models, constraints, etc. The ABPMS creates the artifacts via several means, including from conversations with domain experts [1], from manually-created process documentation and maps, through discovery from event logs [2], or a combination thereof.
2. *Query.* This use case is concerned with answering questions about the past and current states of business processes and their executions. Besides process monitoring information, e.g., KPI dashboards, constraint violations, and related explanations, queries may also address design time concerns, e.g. process model updates and explanations of process changes. The queries can be issued by users using natural language and other communication means or by agents using some mutually agreed upon protocol. A query can return answers in natural language, as a graphical output (dashboard, annotated process model, etc.) or using a specific agent communication protocol.
3. *Recommendation.* This use case is concerned with generating recommendations for process instance adaptation and process evolution (future states). The ABPMS can provide such recommendations based on internal parametric knowledge or it can combine its knowledge with output retrieved from external *components*, such as simulators, optimizers, and solvers. Recommendations can be issued proactively by the ABPMS or requested by users or agents through a conversational interface. A recommendation can be presented to users in natural language or sent directly to an automation component to be directly implemented.
4. *Automation.* This use case is concerned with business process execution and evolution, which can be triggered by the user through natural language, automatically by an agent after the creation of a process-related artefact, or as the implementation of a

recommendation. Depending on the type of action to be taken, automation is delegated to a suitable component, e.g., an RPA bot for automating the execution of an activity or an agent updating the business rules of an ERP system.

Figure 5 maps the use cases to the phases of the traditional BPM lifecycle.

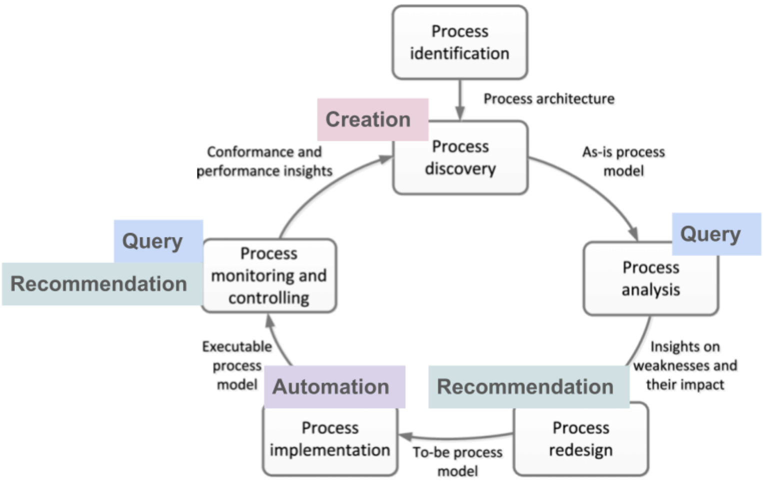


Figure 5 BPM lifecycle annotated with use cases of a conversationally actionable ABPMS.

4.3.2 Architecture

We envision an architecture for a conversationally actionable ABPMS consisting of four subsystems (see Figure 6): a data layer, an intelligence layer, an action layer, and a conversational fabric.

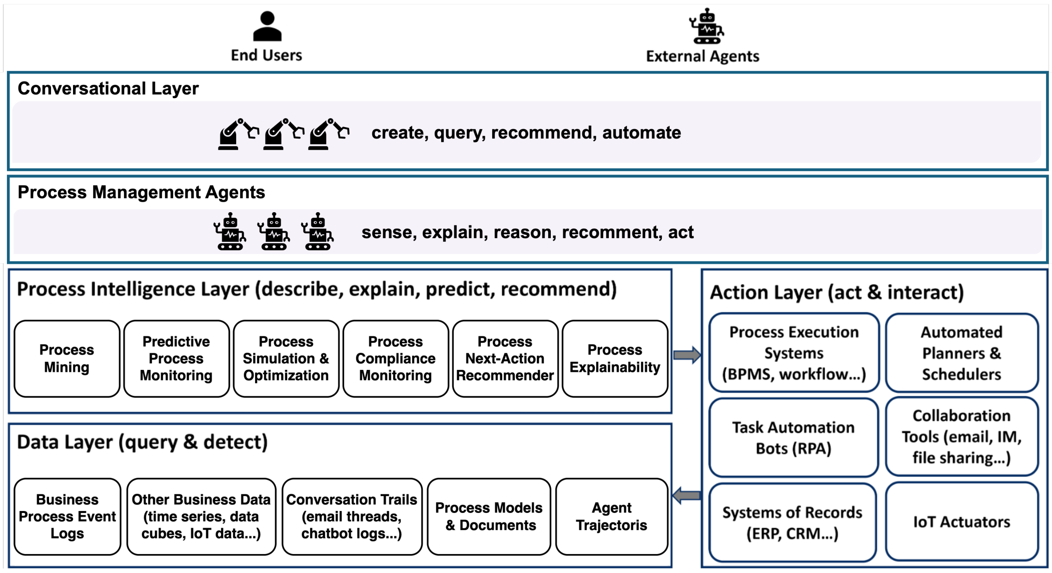

End Users
External Agents
Conversational Layer
create, query, recommend, automate
Process Management Agents
sense, explain, reason, recomment, act
Process Intelligence Layer (describe, explain, predict, recommend)
Process Mining
Predictive Process Monitoring
Process Simulation & Optimization
Process Compliance Monitoring
Process Next-Action Recommender
Process Explainability
Data Layer (query & detect)
Business Process Event Logs
Other Business Data (time series, data cubes, IoT data...)
Conversation Trails (email threads, chatbot logs...)
Process Models & Documents
Agent Trajectoris
Action Layer (act & interact)
Process Execution Systems (BPMS, workflow...)
Automated Planners & Schedulers
Task Automation Bots (RPA)
Collaboration Tools (email, IM, file sharing...)
Systems of Records (ERP, CRM...)
IoT Actuators

Figure 6 Architecture of a conversationally actionable ABPMS.

The data layer brings together structured and unstructured data about a business operation, including event logs of business processes, other business data (financial data, time series), business process documentation (including text documents and process models), as well as data from physical systems relayed via IoT sensors. This layer allows the ABPMS to sense the current state and evolution of the business process and its environment. It provides a querying interface allowing other layers to leverage business process data and metadata from a variety of sources.

On top of the data layer sits a process intelligence layer, which provides capabilities for process discovery and performance analysis (process mining), predictive capabilities (e.g. based on simulation and machine learning), as well as prescriptive capabilities (recommending intervention). In other words, the process intelligence layer provides capability to describe and explain the current state of the process and to predict future states of the process under different scenarios.

Next to the process intelligence layer, the action layer provides capabilities for triggering actions affecting one or more business processes, or interactions with external actors (suppliers, customers, etc.). Example of actions include: (1) creating or altering the state of a case in a business process orchestrated by a Business Process Management System (BPMS); (2) triggering a software bot; (3) sending notifications via a communication platform; (4) updating records in a CRM, ERP or other System of Records.

The fourth component is the *conversational fabric*. This layer makes available the capabilities of the data layer, process intelligence layer, and action layer to different types of agents. The capabilities of the lower layers are exposed via *tools*. These tools are exposed to agents via a *Model Context Protocol* (MCP) [3] layer, which provides semantically rich descriptions of the tools for consumption by agents

An agent may leverage some of these tools to, for example, detect degradations in the performance of a process that may lead to a violation of a Service Level Agreement (SLA). Having detected this risk, the agent may leverage the process intelligence tools to determine interventions that may be triggered to prevent this SLA violation, and it may then leverage the tools coming from the action layer to trigger actions or notifications.

The agents in the conversational fabric receive instructions and goals from end users, directly via for example a chat interface, or indirectly via agents operating in other systems, such as an agent running in a CRM platform.

Some agents rely on general-purpose LLMs, others are based on fine-tuned or domain-specific models, and some possess explicit planning or reasoning capabilities. Each agent operates autonomously but may collaborate with other agents by passing tasks, sharing context, or requesting specific operations.

4.3.3 On Actions and their Composition

Conversations do not only happen when external (human or artificial) agents interact with the ABPMS to obtain information, insights, and explanations about the current state or the history of the ABPM itself. They are also central to enable *actionability*, that is, to support actions in a broad sense. Actions may pertain to:

- The very execution of the process, e.g., approving a purchase order request, or sending a warehouse replenishment request.
- Process (re)framing, e.g., adding or removing constraints, and (re)modelling, e.g., evolving a model due to changing requirements.
- Decisions and interventions for process improvement, e.g., hiring a new resource to improve cycle time, or deciding to cancel an order because of economic considerations)

In this context, conversating is essential to gather information and insights before taking – or deciding whether to take – an action. For example, an agent would need to ponder the impact of hiring a new resource before committing to do so, or may decide to cancel an order depending on the corresponding penalty. This can only be achieved with good quality guarantees if the agent can conversate with other agents, as well as it can invoke the tools needed for the task at hand.

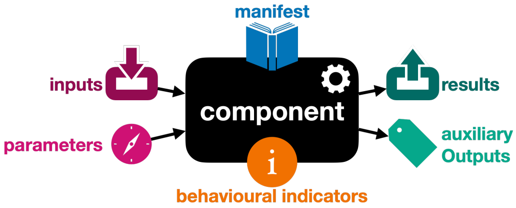


Figure 7 A component with its inputs, outputs, and metadata.

In general terms, a *component* is conceived as a deterministic, verifiable unit of software that realises a certain task and/or produces some result given some inputs and preconditions. A tool may be a manually crafted software or it may rely on generative AI, but in any case, it is deemed to have gone through sufficient quality control to be trusted.

As shown in Figure 7, to enable agents in using components and interpreting the obtained results, some key aspects must be described:

- *Input* – an input object (possibly with some preconditions) necessary to invoke the component; for example, the process model required by a simulator.
- *Parameter* – an auxiliary input used to tune the component; for example, hyperparameters used to configure the simulator.
- *Result* – an output object produced by the component; for example, the process cycle time produced by the simulator.
- *Auxiliary output* – an auxiliary output produced by the component to help interpreting the output; for example, quantified uncertainty associated to a prediction.
- *Manifest* – a description of the component, its behaviour, and its inputs and outputs; the manifest relates to MCP, as its main purpose is to enable agents to discover and properly consume the component.
- *Behavioral indicator* – a quantitative or qualitative indicator characterising the (functioning of the) component along a key functional or non-functional concern (cf. Section 4.3.4), such whether the component produces analytical or approximate results, or whether it guarantees data privacy, whether it requires a rigid description of the input, and the like; relevant aspects that cannot be clearly described as indicators could still be included in the manifest of the component.

An agent may jointly employ multiple tools through different usage patterns, like concatenating two components to obtain a new functionality, or invoking two components in parallel and then aggregating their results. In addition, as pointed out before, it may conversate with other agents. This could lead to an autonomous decision taken by the agent relying on the result of the conversation and of the employed software components, or in an even more complex setting, resulting from collective decision making (akin to multi-agent systems [4]).

A crucial aspect of this architecture is *trust*. Agents are only deemed trustworthy if they can transparently explain their outputs. This is achieved by enabling agents to trace and articulate how they invoked specific components, what input they provided, and how the resulting outputs contributed to their decisions or responses have been used. This traceability, which relates to the more general concept of ABPMS explainability, ensures that agent outputs are not opaque, but grounded in reproducible and verifiable actions.

In other words, trust propagates from tools to agents if the agents are able to trace their own outputs to outputs obtained from the components. Actionability is in the components, while conversationality is provided by the agents. This trust propagation approach from tools to agents is sketched in Figure 8.

In the context of an ABPMS, tools provide access to descriptions about the current state of the process, diagnosis for issues, predictions, explanations, and recommendations, among others. Other components may provide actions, e.g. executed by rule-based bots, or by bots driven by automated planners, which perform actions on top of systems that have an impact on the "real world" such as transactional systems (which maintain ground-truth of reality) or communication systems (e.g. sending an email).

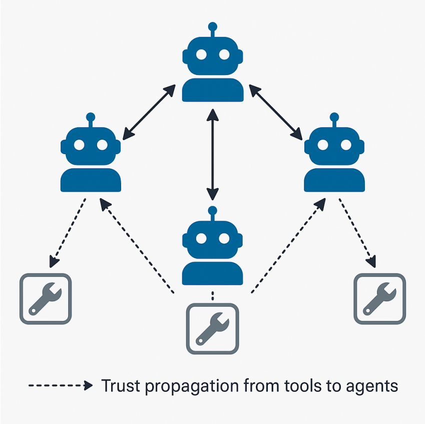


Figure 8 Trust propagates from tools to agents, which are connected in a network.

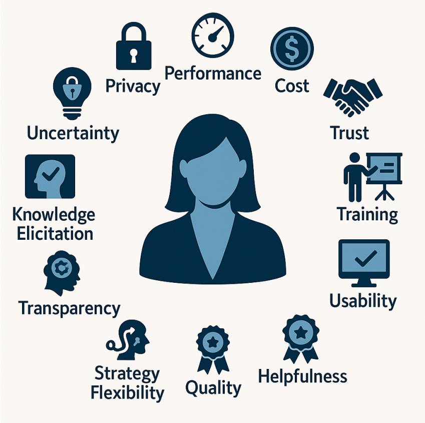


Figure 9 Important conversationally actionable ABPMS concerns.

4.3.4 Key Concerns and Challenges

We have identified a non-exhaustive list of key concerns (mostly non-functional in nature) closely relevant to the conversational actionability of ABPMS (Figure 9 and Table 3). These concerns (except Training and probably Strategy Flexibility and Knowledge Representation) are expected to be specified, tracked, and composed using measurable indicators (e.g., in components), as has been done in other areas such as Service-Oriented Architecture [5].

Table 3 Key concerns for conversationally actionable ABPMS.

Concern	**Description**
Performance	The extent to which the ABPMS meets time, throughput, and capacity requirements.
Cost	The financial impact of invoking external tools and services (including LLMs) on the architecture (e.g., smaller but specialized LLMs) and operation of the ABPMS.
Trust	The degree to which users and agents believe that the ABPMS outputs (e.g., models, recommendations, automations) are true, accurate, and dependable.
Training	The level of effort and education required for users to effectively learn and utilize the ABPMS and its conversational capabilities.
Usability	The degree to which the ABPMS interface and interaction mechanisms enable users to accomplish their goals efficiently and satisfactorily.
Helpfulness	The extent to which the outputs of the ABPMS support users in completing or improving their tasks and decision-making.
Quality	The accuracy, correctness, and completeness of the results provided by the ABPMS in support of user and agent tasks.
Transparency	The extent to which the ABPMS provides understandable and traceable explanations for its outputs, actions, or reasoning.
Strategy Flexibility	The degree to which the ABPMS can support the tailoring of human requests to generic or specific strategies (e.g., by specifying which external tools to use and combine).
Knowledge Elicitation	The ability of the ABPMS to query and integrate external or internal knowledge sources (e.g., web, databases, ontologies) to enhance its reasoning or task performance.
Uncertainty	The extent to which the ABPMS can represent, communicate, and act upon confidence levels of its outputs and outputs.
Privacy	The degree to which the ABPMS respects and enforces privacy constraints arising from its frame (laws, policies, or user preferences) during its operations and model fine-tuning.

Stemming from these concerns and the previous discussion, we identify important challenges for the development of conversationally actionable ABPMS. Most are short-term, but the last two are longer-term.

1. How are actionable conversations defined and evolved?
2. How can conversation needs and mechanisms (e.g., MCP) be elicitated and specified to enable an ABPMS to use discover, understand, and invoke the capabilities of other tools and services (e.g., simulators or AI-based agents)?
3. How can the concerns identified in Table 3, and especially trust, be understood and increased in an ABPMS?

4. How can indicators for these concerns be specified, measured, and aggregated within the ABPMS but also within component contexts (e.g., with MCP)?
5. How can unreliable AI-based components and certified tool components be composed dynamically to contribute towards satisfying the goals of an actionable conversation while addressing the identified concerns?
6. How can an ABPMS itself become a component (MCP-based or other) that can be used by larger ecosystems?
7. (Long-term) How can we support the automated generation of automations (RPA, AI agents, others) from within an APBMS?
8. (Long-term) How can frames related to conversations and to interactions with external services/tools be flexibly managed?

References

1 Nataliia Klievtsova, Timotheus Kampik, Juergen Mangler, and Stefanie Rinderle-Ma. Conversationally actionable process model creation. In Marco Comuzzi, Daniela Grigori, Mohamed Sellami, and Zhangbing Zhou, editors, *Cooperative Information Systems - 30th International Conference, CoopIS 2024, Porto, Portugal, November 19-21, 2024, Proceedings*, volume 15506 of *Lecture Notes in Computer Science*, pages 39–55. Springer, 2024.

2 Wil M. P. van der Aalst and Josep Carmona, editors. *Process Mining Handbook*, volume 448 of *Lecture Notes in Business Information Processing*. Springer, 2022.

3 Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. Model context protocol (MCP): landscape, security threats, and future research directions. *CoRR*, abs/2503.23278, 2025.

4 Ali Dorri, Salil S. Kanhere, and Raja Jurdak. Multi-agent systems: A survey. *IEEE Access*, 6:28573–28593, 2018.

5 Hanane Becha and Daniel Amyot. Consented consumer-centric non functional property description and composition for SOA-based applications. *Int. J. Web Eng. Technol.*, 10(4):355–392, 2015.

4.4 Working Group on Explainability

Peter Fettke (Saarland University – Saarbrücken, DE & German Research Center for Artificial Intelligence (DFKI) – Saarbrücken, DE, peter.fettke@dfki.de)
Fabiana Fournier (IBM Research Israel – Haifa, IL, fabiana@il.ibm.com)
Lior Limonad (IBM Research Israel –Haifa, IL, liorli@il.ibm.com)
Andreas Metzger (University of Duisburg-Essen, DE, andreas.metzger@paluno.uni-due.de)
Stefanie Rinderle-Ma (TU Munich, DE, stefanie.rinderle-ma@tum.de)
Barbara Weber (University of St. Gallen, CH, barbara.weber@unisg.ch)

4.4.1 Introduction

An autonomous business process (ABP) is the next generation of AI-Augmented Business Process Management System (ABPMS) [1], which is a self-executing ABPMS that leverages advanced technologies such as Artificial Intelligence (AI) and Machine Learning (ML) to operate with minimal to no human intervention. ABPs can sense and respond to various inputs, reason, make decisions, and adapt to changing circumstances in real time, all without relying on manual triggers or continuous oversight. Think of it like a self-driving car for your business operations. Instead of a human driver controlling all aspects, the system uses sensors, data analysis, and intelligent algorithms to navigate and achieve its objectives.

The notion of ABPs was recently introduced at the AutoBiz Dagstuhl seminar[1], during which the core material for this paper was jointly developed.

One reason for realizing ABPs is to improve operational efficiency, reduce errors, lower costs, improve response times, and free human workers for more strategic and creative work. However, despite the promises that ABPs offer, their characteristics can lead to particular concerns in the context of BPM:

- ABPs may erode *trust* among stakeholders – including process owners, business analysts, end-users, and customers – who may be hesitant to rely on or adopt AI-based process recommendations or automated decisions if they cannot understand the rationale behind them.
- The opacity of ABPs may make it difficult to *debug* process models, as well as identify potential failures, or understand why a process might be under-performing.
- Using ABPs may hinder *accountability*; if an ABP leads to a failure or an unfair outcome, the inability to explain its underlying decisions makes it challenging to assign responsibility or implement corrective actions.
- ABPs may perpetuate hidden *biases* of their underlying AI and ML components. Such biases may lead to discriminatory or unfair process outcomes, which can be difficult to detect and mitigate.
- Demonstrating the *compliance* of ABPs with regulatory frameworks, such as the EU's GDPR and AI Act, requires an increasing level of transparency, particularly in high-risk domains like finance, healthcare, and human resources, which are common areas for BPM applications.

We argue that *explainability* will be a key characteristic of ABP systems to address the aforementioned concerns [1, 2], leading to the notion of **eXplainable ABPs (XABPs)**.

XABPs are particularly relevant when ABPs are realized in the form of *Agentic BPM* systems. An Agentic BPM system is an advanced approach to managing and automating complex business workflows by integrating autonomous AI agents. Unlike traditional BPM or Robotic Process Automation (RPA) systems that follow rigid, predefined rules and workflows, agentic BPM leverages AI to enable systems to make independent decisions, adapt to changing conditions, and learn from experience with minimal human intervention. Here, explainability offers a central mechanism through which agents can articulate the rationale behind their behavior. As such, explainability becomes a first-class citizen in the realization of Agentic BPM systems, supporting agent autonomy from two perspectives:

- Enabling agents to independently resolve misalignments in other agents' behavior.
- Reducing human intervention by making agent behavior understandable and transparent.

Employing state-of-the-art explainable AI (XAI) techniques [3] for XABPs pose several limitations:

1. Inability to express business process model constraints [4].
2. Failure to capture the richness of contextual situations that affect process outcomes [5].

[1] See `https://www.dagstuhl.de/25192`. We express our gratitude to the Scientific Directorate and staff of Schloss Dagstuhl for their invaluable support. We also thank our fellow participants for their engaging discussions.

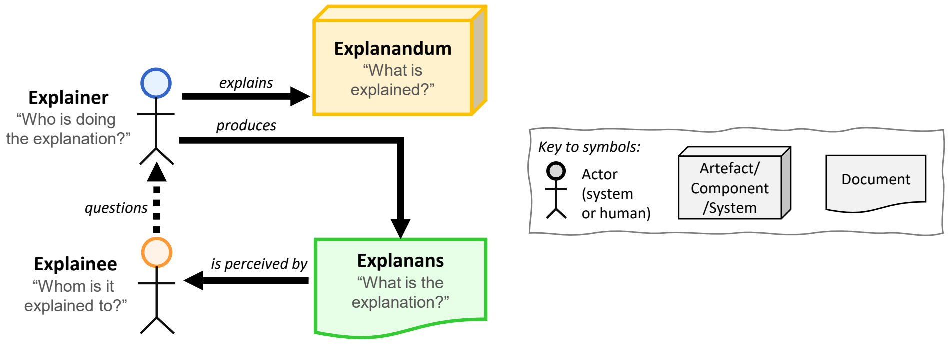


Figure 10 Explainability Concepts.

3. Inability to reflect causal execution dependencies among activities in the business process [6].
4. Explanations are often nonsensical or not interpretable for human users [7].

4.4.2 Characterization and Needs of XABPs

We start with a generic conceptualization of explainability and then refine this to particular concerns in the BPM setting.

4.4.2.1 Fundamental Explainability Concepts

Figure 10 illustrates the key explainability concepts.

The *explainer* provides an explanation of the *explanandum* (explanation subject) by offering one or more *explanans/explanantia* (explanation/explanations). The explanation is generated by the explainer using a specific explanation mechanism at a defined generation time, and is delivered to the *explainee* in a particular presentation format – typically visual or textual. In its simplest form, the explanantia produced by the explainer should provide information about the causes of the explained phenomenon (explanandum) [8]. The content of the explanation must align with both the nature of the explanandum and the needs of the explainee. Furthermore, interaction of the explainee with the explanation can follow different modes, ranging from one-shot explanations to conversational or multi-round interactions.

4.4.2.2 Explanandum: Explanation Subjects *"what is explained?"*

Figure 11 shows the key types of aspects of an explanandum that may be explained, as elaborated below.

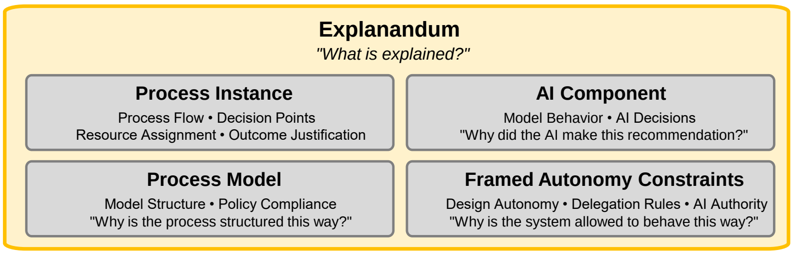

Explanandum
"What is explained?"
Process Instance
Process Flow • Decision Points
Resource Assignment • Outcome Justification
AI Component
Model Behavior • AI Decisions
"Why did the AI make this recommendation?"
Process Model
Model Structure • Policy Compliance
"Why is the process structured this way?"
Framed Autonomy Constraints
Design Autonomy • Delegation Rules • AI Authority
"Why is the system allowed to behave this way?"

Figure 11 Explainability Concepts.

Process Instance Explanation: ***"Why did this specific process execution take the path and produce the result it did?"***

- *Process Flow* - The sequence of activities, decisions, and events in the business process. Example: "Why did the invoice approval take 5 days?"
- *Decision Points* – Why certain paths or outcomes were chosen during process execution. Example: "Why was a customer's request escalated instead of being resolved at Tier 1?"
- *Resource Assignment* – Why specific tasks were assigned to certain roles or individuals. Example: "Why was this case handled by the senior team?"
- *Outcome Justification* Why a specific result occurred. Example: "Why was this loan application rejected by the process?"

Process Model Explanation: ***"Why is the process structured the way it is?"***

- *Model Structure:* Why are certain activities, decisions, or flows included? E.g., "Why do we have a credit history check as a decision point?"
- *Policy Compliance:* Whether and how policies shaped the model or its execution E.g., "Was the data retention policy followed?"

AI Component Explanation: ***"Why did an AI component make this recommendation or decision?"***

Note, that this here very much refers to explainable AI (XAI). In more detail:

- *AI Model Outcome:* "Why did the AI component predict deviations or prescribe proactive adaptations?" [9] E.g., "Why was an alarm raised for process event e_j?"
- *AI Model Behavior:* "Why does the AI model have certain characteristics or properties?" E.g., "Why does the LSTM prediction model have a Mean Absolute Error (MAE) of only .35 for the given process domain?"

Framed Autonomy Explanation: ***"Why is the system or process allowed to behave as it does?"***

- *Design Autonomy:* "Why can the process bypass manual review?"
- *Delegation Rules:* "Why do Tier 1 agents have approval authority?"
- *AI Authority:* "Why can the AI act without a human in the loop?"
- *Escalation Thresholds:* "Why is escalation triggered only after 3 attempts?"
- *Compliance Limits:* "Why is this exception allowed under the GDPR?"

4.4.2.3 Explainer

XABPs involve a range of human and system actors that either generate or consume explanations. Some actors – especially autonomous systems or agents – may fulfill both roles, such as generating explanations for others while also using explanations for self-reflection or system adaptation.

Figure 12a shows the key aspects of the explainer, elaborated below.

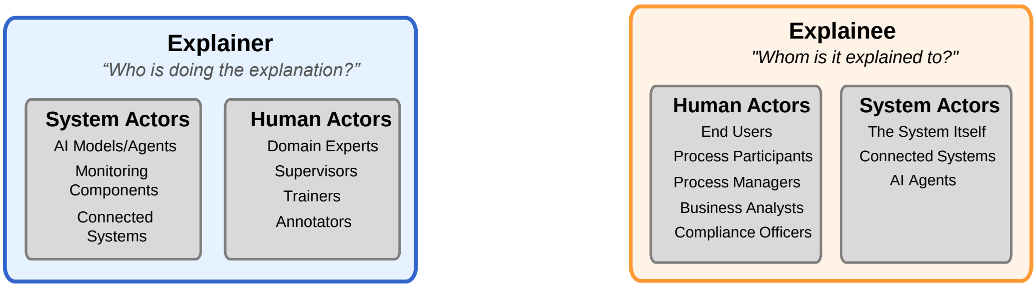
Table 4 Explanation Providers in Business Processes (Explainers).

(a) System Explainers: *systems providing or formalizing explanations*

Actor Type	**Role in Ecosystem**	**Explanation About**	**Mechanism or Method**
AI Agents	Intelligent assistants, bots	Predictions, task outcomes, alerts	SHAP, LIME, rule-based reasoning, counterfactuals
Monitoring Components	Process mining engines, workflow monitors	Process events, performance, exceptions	Temporal rules, KPI tracking, log analysis
Connected Systems	External APIs or services	State changes, synchronization info	Metadata contracts, semantic logging

(b) Human Explainers: *humans providing or formalizing explanations*

Actor Type	**Role in Ecosystem**	**Explanation About**	**Communication Mechanism**
Domain Experts	Analyst specifying business rules	Process logic, decision criteria, exception handling	Process documentation, model annotations, verbal explanation
Supervisors / Managers	Operations or compliance manager	Justification for overrides, escalations, or decisions	Reports, notes, emails, verbal feedback
Trainers / Annotators	Labelers or human-in-the-loop operators	Ground truth or feedback to train explainable systems	Annotation tools, structured forms, chat interfaces



(a) Aspects of Explainer

(b) Aspects of Explainee

Figure 12 Overview of explainer and explainee aspects.

4.4.2.4 Explainee

Figure 12b shows the key aspects of the explainee, elaborated below.

Table 5 Explanation Recipients in Business Processes (Explainees).

(a) Human Explainees: *humans consuming explanations.*

Actor Type	**Example Role**	**Needs from Explanations**	**Preferred Explanation Style**
End Users	Customer applying for a loan	Understand decisions about them (e.g., rejections, delays)	Simple, outcome-focused, natural language
Process Participants	Agent handling loan verification	Know what task to do next and why	Step-by-step task rationale, alerts, real-time updates
Process Managers	Operations lead, shift manager	Monitor KPIs, react to anomalies, adapt resources	Dashboards, alerts, summaries, what-if analysis
Business Analysts / Domain Experts	Person modeling the process	Improve efficiency, detect bottlenecks, validate rule logic	Process mining results, causal analysis, counterfactuals
Compliance Officers / Auditors	Internal or external auditors	Ensure traceability, legality, policy adherence	Audit trails, rule execution logs, exception reports

(b) System Explainees: *self-reflective systems consuming explanations.*

Actor Type	**Role in the Ecosystem**	**Needs from Explanations**	**Communication Mechanism**
The System Itself	Autonomous BPM or AI component	Self-monitoring, internal diagnosis, reconfiguration	Logs, symbolic reasoning, anomaly detection
Connected Systems	CRM, ERP, or DMS components	Data or process synchronization with semantic clarity	API contracts, structured events, semantic metadata

4.4.2.5 Explanans

Figure 13 shows the key aspects of an explanans, elaborated below.

Explanation Mechanism: ***"How is the explanation generated?"***
The explanation mechanism refers to the approach employed by the explainer to generate an explanation – such as attributing feature importance, selecting representative examples, deriving symbolic rules, constructing interpretable approximations, identifying counterfactuals, or visualizing model behavior.

- *Feature Attribution:* Assigns contribution (credit or blame) to input features. *Examples:* SHAP, LIME, Saliency Maps
- *Example-Based:* Uses similar or contrasting examples to justify a decision. *Examples:* k-NN, Prototypes, Counterfactuals

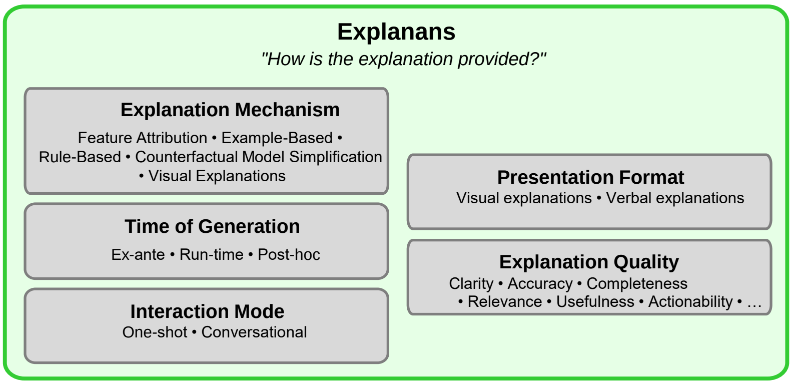

Explanans
"How is the explanation provided?"
Explanation Mechanism
Feature Attribution • Example-Based • Rule-Based • Counterfactual Model Simplification • Visual Explanations
Time of Generation
Ex-ante • Run-time • Post-hoc
Interaction Mode
One-shot • Conversational
Presentation Format
Visual explanations • Verbal explanations
Explanation Quality
Clarity • Accuracy • Completeness • Relevance • Usefulness • Actionability • ...

Figure 13 Aspects of an Explanans.

- *Rule-Based* Derives symbolic or logical rules from data or models. *Examples:* Decision Trees, Rule Lists, Association Rules
- *Model Simplification:* Approximates complex models with interpretable surrogates. *Examples:* Surrogate Decision Trees, Linear Proxies
- *Counterfactual:* Explains what minimal input change would alter the outcome [10]. *Example:* “If income were $5,000 higher, the outcome would have changed.”
- *Visual Explanations:* Uses visual indicators to represent decision logic or model behavior. *Examples:* Heatmaps, Partial Dependence Plots

Time of Explanation Generation: *“When is the explanation produced?”*
The timing of an explanation determines its role in the lifecycle of decision-making systems. Explanations may be generated before, during, or after system execution:

- *Ex-ante Explanations (Before Execution):* Provided before the system executes or makes a decision to validate models or justify decisions before deployment.
- *Run-time Explanations (During Execution):* Delivered while the process is running to support human-in-the-loop oversight or adaptive user feedback.
- *Post-hoc Explanations (After Execution):* Generated after the process completes its actions or decisions in order to audit, debug, or help users understand outcomes.

Presentation Format of Explanation: *“How is the explanation presented to the user?”*
The chosen presentation method has a direct effect on user comprehension and, therefore, on the success of the explanations [11].

- *Visual explanations*: Heatmaps, charts, dashboards, saliency maps
- *Verbal explanations*: Natural-language output, written rules, factual/counterfactual statements

Interaction with Explainee: *“How does the user interact with the explanation?”*
Interaction of the explainee with the explanation refers to the mode and extent of user involvement in the explanation process:

- *One-shot explanations:* Explanation provided once, passively
- *Query-based explanations:* Explanation provided on-demand, actively
- *Multi-round / Conversational:* Interactive, iterative, potentially adaptive dialogue [12]

Explanation Quality: ***"How to assess the quality of explanations?"***

Explanation quality may be assessed along two complementary dimensions:

- *Technical quality:* This relates to the technical properties of the explanation method itself. Examples include fidelity aka. faithfulness aka. soundness (which measures how accurately the explanation reflects the reasoning or behavior of the explanans), and stability (an explainer should provide similar explanations for similar input or minor perturbations of the input).
- *User-centric quality:* This relates to how the explanation is perceived by humans (in the role of explainees). Examples include usefulness (which quantifies how well it helps the explainee to solve a problem, understand a concept better, or apply the knowledge in a new situation) and meaningfulness (explanation is relevant to the specific explainee and the question or topic at hand and avoids unnecessary tangents or irrelevant information that could confuse the explainee).

4.4.3 Challenges for Explainable ABPs

We structure the challenges along the four main explainability concepts as well as along overarching concerns.

4.4.3.1 Challenges Related to Explainee

Challenge 1: How to specify preferences regarding explanations? Specifying preferences for explanations presents multifaceted challenges. First, input mechanisms must effectively capture preferences through various channels, whether explicitly declared upfront, interactively elicited through dialogue, or implicitly inferred from user behavior. Systems must accommodate both static preferences that remain consistent and those that dynamically adapt to changing contexts, while supporting the natural evolution of preferences as the explainee's understanding develops. Second, inevitable preference conflicts need to be navigated. This involves carefully balancing competing dimensions, such as detail versus conciseness and speed versus accuracy. This requires finding trade-offs without sacrificing critical explanatory qualities.

4.4.3.2 Challenges Related to Explanandum

Challenge 2: What explanation subjects are needed for ABPMs? Explainability related to AI components are rather well understood - not so for ABPM. From our understanding, explanation subjects such as process instance, process models, and framed autonomy constraints are interesting and relevant. However, a more mature taxonomy of explanation types might evolve.

4.4.3.3 Challenges Related to Explainer

Challenge 3: Which techniques are needed for the explainer to generate explanations? In the broader field of XAI, several techniques for XAI are known. In addition, in the field of BPM, several techniques are emerging. On this foundation, new techniques should be developed, e.g., what-if analysis, process outcome analysis. All these techniques should take causality into account. In the future, it should be clarified how existing BPM techniques can be integrated, e.g., visualization, and be exploited for creating explanations. Explainability of frames is a nearly unexplored field.

4.4.3.4 Challenges Related to Explanans / Explanantia

Challenge 4: How may one articulate actionable explanations (e.g., to other agents) to preserve autonomy? While the explanans is constructed to make an explanation informative about the circumstances that may led to the situation being inquired (i.e., the explanandum), in the context of XABPs, the explanans may also adopt an actionable style – indicating to the explainee which corrective or mitigating actions could be taken to alter the state of the explanandum, particularly without escalating the situation to any external agent. In this way, the explainee may be able to autonomously act upon the condition at hand. However, further work is needed to devise a systematic approach that enables the explainer to determine the most effective content for the explainee, to elicit such corrective action – taking into account both the explanandum and the behavioral intentions of the explainee.

Challenge 5: When to generate explanations (generation time) and how long to preserve them [13]? The question here is whether explanations should be generated upfront, whenever possible, or whether we can or should be more conscious about the generation time of the explanation. Another question is when to discard outdated explanations.

Challenge 6: How can explanations automatically adapt their form to suit the identity of the explainee? The question is how the explanation can be presented in a way that is easily understandable for the explainee, e.g., leads to low cognitive load for human explainees, and answers the explanation needs of the explainee. This could also be motivated by organizational motivation and goals.

Challenge 7: How can we accommodate explanations that consider (why) certain behaviors did not occur? Explaining non-occurring behavior is more challenging than explaining occurring behavior and requires capturing or acquiring knowledge about non-occurring behavior. Causal analysis might be helpful here.

Challenge 8: How may we synthesize a variety of perspectives (e.g., data, contextual, exogenous) into the explanation? The first challenge here is to collect and create data sets that cover different perspectives and are of sufficient quality. It is essential to be able to link the synthesized data to process instances. Moreover, providing explanations on synthesized data might also require selecting and filtering the data adequately again to provide adequate explanations.

Challenge 9: How to identify causal explanations? Causality vs. correlation: Not every correlation between two variables has a causal explanation. It is therefore important to distinguish between spurious correlations and causality. This classical distinction is well known, but must also be observed in the context of explainable ABPMN. The explainer can provide the explainee with information about the degree of certainty of the explanation offered.

Challenge 10: How do explanations evolve over time based on feedback or changing context? Explanations might have to be updated based on changing context and feedback, e.g., if sensor data starts to deviate. The first question is how to detect that an explanation that (partly) takes into account the sensor data has to be updated? Another question is when to present the updated explanation to the user, i.e., directly after a changing context was detected or at another, possibly better fitting moment? This question is related to the question of explanation update frequency. Here, the challenge is to find the sweet spot between keeping explanations up to date and not confusing the explainee. Finally, we have to think about when and how to provide full versus incremental explanation updates.

Challenge 11: How does the realization of the "frame" in ABPMSs affect the one of explainability? i.e., with autonomous agents, it may be the means for the agents to share with other agents the rationale for their own behavior.

4.4.3.5 Overarching Challenges

Challenge 12: How may one assess the quality of the explanations? Evaluating explanation quality presents a fundamental challenge requiring both empirical and theoretical approaches. From an empirical perspective the question needs to be answered how can we effectively measure explanation quality when objective and subjective dimensions must both be considered? Objective measures include, for example, factual accuracy and completeness, while user-centered aspects cover, for example, comprehensibility, usefulness, effectiveness and efficiency (e.g., see [14]), as well as being actionable.

Challenge 13: Which kind of datasets are needed to serve as explainability benchmarks? Benchmarking is a typical approach to evaluating system performance. We expect that such an idea can also promote the development of the field of accountability. However, benchmarking typically relies on adequate benchmark data. In principle, such data can be generated in a laboratory setting. But adequate data are also needed for benchmarking explainability systems in the field.

Challenge 14: How to ensure that explanations do not reveal information that may be privacy-sensitive, reveal business-critical IPR, or make it easier to undermine the security of the system? In essence, the challenge lies in providing enough information to satisfy the need for explainability without compromising other crucial aspects of the business, such as data privacy, competitive advantage, and system security.

4.4.4 Conclusion

An autonomous business process (ABP) represents a paradigm shift towards self-executing workflows driven by AI and ML Yet ABPs introduce challenges related to trust, transparency, accountability, bias, and regulatory compliance within BPM. To address these issues, this paper introduced the notion of explainable ABPs (XABPs), which can articulate the rationale behind their actions and underlying models. Current explainable AI (XAI) techniques fall short in capturing the complexities of the BPM setting. We therefore introduced a set of challenges to stimulate further research on XABPs.

References

1 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. AI-augmented business process management systems: A research manifesto. *ACM Trans. Manag. Inf. Syst.*, 14(1):11:1–11:19, 2023.

2 Nijat Mehdiyev, Maxim Majlatow, and Peter Fettke. Interpretable and explainable machine learning methods for predictive process monitoring: A systematic literature review, 2023.

3 Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6, 2018.

4 Guy Amit, Fabiana Fournier, Shlomit Gur, and Lior Limonad. Model-informed lime extension for business process explainability. In *PMAI@IJCAI'22*. CEUR, 2022.

5 Guy Amit, Fabiana Fournier, Lior Limonad, and Inna Skarbovsky. Situation-aware explainability for business processes enabled by complex events. In *BPM-W 2022*, volume 460 of *LNBIP*, pages 45–57. Springer, 2022.

6 F. Fournier, L. Limonad, I. Skarbovsky, and Y. David. The why in business processes: Discovery of causal execution dependencies. *Künstliche Intelligenz*, 1 2025.

7 Dirk Fahland, Fabiana Fournier, Lior Limonad, Inna Skarbovsky, and Ava J. E. Swevels. How well can large language models explain business processes as perceived by users? *Data & Knowledge Engineering*, 157:102416, 2 2025.

8 Peter Lipton. *Causation and Explanation*, pages 619–631. Oxford University Press, 1 2010.

9 Andreas Metzger, Tristan Kley, Aristide Rothweiler, and Klaus Pohl. Automatically reconciling the trade-off between prediction accuracy and earliness in prescriptive business process monitoring. *Inf. Syst.*, 118:102254, 2023.

10 Tsung-Hao Huang, Andreas Metzger, and Klaus Pohl. Counterfactual explanations for predictive business process monitoring. In Marinos Themistocleous and Maria Papadaki, editors, *EMCIS 2021*, volume 437 of *LNBIP*, pages 399–413. Springer, 2021.

11 Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, and Navid Nobani. Convxai: a system for multimodal interaction with any black-box explainer. *Cogn. Comput.*, 15(2):613–644, 2023.

12 Andreas Metzger, Jone Bartel, and Jan Laufer. An AI chatbot for explaining deep reinforcement learning decisions of service-oriented systems. In *ICSOC 2023*, volume 14419 of *LNCS*, pages 323–338. Springer, 2023.

13 Nijat Mehdiyev and Peter Fettke. Explainable artificial intelligence for process mining: A general overview and application of a novel local explanation approach for predictive process monitoring. In *Interpretable Artificial Intelligence: A Perspective of Granular Computing*, pages 1–18. Springer, 2021.

14 Andreas Metzger, Jan Laufer, Felix Feit, and Klaus Pohl. A user study on explainable online reinforcement learning for adaptive systems. *ACM Trans. Auton. Adapt. Syst.*, 19(3):15:1–15:44, 2024.

5 AUTOBIZ Dissemination

This section gives a brief overview of how existing and new knowledge of AUTOBIZ is and will be disseminated. In the former case, the focus is on education (covered by Subsection 5.1, whereas in the latter case, the focus is on venues for the dissemination of novel results, i.e., workshops, journals, and conferences (Subsection 5.2).

5.1 Embedding of ABPMS-related Training into University Curricula

With the rise of "agentic AI" in enterprise software, autonomous business process execution has reached the mainstream. However, there is substantial uncertainty on how to achieve autonomy in a reliable and sustainable manner, with analysts such as Gartner using the term "agent washing" to describe pseudo-autonomous enterprise subsystems and predicting that 40% of agent deployment projects will fail by 2027[2]. In order to provide an education foundation for effective, efficient, and sustainable AUTOBIZ, it is crucial to adjust both business process

[2] Cf. `http://s.cs.umu.se/ni2riq` (Gartner press release, URL shortened), *accessed at 2025-07-18.*

management and applied AI curricula as soon as possible. This should happen on both undergraduate and graduate levels; just as important are continuous education offerings for the professionals who are exposed to autonomous process execution systems already now. Many of the seminar participants are prolific educators, having, for example, co-authored widely used textbooks on business process management and related topics. Accordingly, the expectation is that the seminar content will be integrated not only into local education offerings at participants' universities but also into materials that are more widely distributed. Materials are expected to be tailored to different educational needs, based on the educational level as well as technical and practical skills and experience of several target audiences (e.g., business students vs. engineering students vs. seasoned practitioners). Specific first steps for developing course materials have been taken. For example, at Umeå University (Sweden), funding has been secured to develop a course on "Sustainable AI Transformation".

5.2 Related Publications and Future Venues

The discussions of the working groups, as presented in this report, have matured into four accepted papers to appear in the CEUR proceedings of the PMAI workshop, held at the ECAI 2025 conference. In addition, a tutorial at BPM 2025 is planned for presentation by Prof. Andreas Metzger on the topic of "AI-Assisted Business Process Monitoring", including insights drawn directly from the discussions in this seminar. For novel and strong technical results that participants and other researchers in the BPM and AI communities produce on the topic, a special issue in the *Information Systems* journal – the prime journal venue for technical BPM research results – has been announced, guest-edited by the seminar organizers.

Given the prevalence of AI in BPM research, further steps may be taken to facilitate the continued dissemination of AUTOBIZ-related research results. For example, discussions have started regarding the establishment of a new *AIXBPM forum* at the Business Process Management conference, the prime conference venue of BPM research.

Future research on AUTOBIZ is expected to enjoy substantial funding success. Notably, seminar co-organizer Marlon Dumas has secured an ERC grant for the *AI-Assisted Optimization of Business Processes*[3]. Also, the organizations of many of the industry participants are expected to continue to fund and drive AUTOBIZ research and its transfer into practice at scale.

[3] https://cs.ut.ee/en/news/university-tartu-researcher-was-awarded-erc-grant-turn-frontier-science-practical-solutions, *accessed at 2025-07-18.*

Participants

- Daniel Amyot
University of Ottawa, CA
- Diego Calvanese
Free University of Bozen-Bolzano, IT
- Marco Comuzzi
Ulsan National Institute of Science and Technology, KR
- Giuseppe De Giacomo
University of Oxford, GB
- Marlon Dumas
University of Tartu, EE
- Achiya Elyasaf
Ben Gurion University – Beer Sheva, IL
- Peter Fettke
DFKI – Saarbrücken, DE
- Fabiana Fournier
IBM Israel – Haifa, IL
- Timotheus Kampik
SAP Berlin, DE & Umeå University, SE
- Yves Lesperance
York University – Toronto, CA
- Lior Limonad
IBM Israel – Haifa, IL
- Andrea Marrella
Sapienza University of Rome, IT
- Andrea Matta
Polytechnic University of Milan, IT
- Andreas Metzger
Universität Duisburg – Essen, DE
- Marco Montali
Free University of Bozen-Bolzano, IT
- Stefanie Rinderle-Ma
TU München, DE
- Sebastian Sardiña
RMIT University – Melbourne, AU
- Arik Senderovich
York University – Toronto, CA
- Estefania Serral Asensio
KU Leuven, BE
- Niek Tax
Meta – London, GB
- Irene Teinemaa
Google DeepMind – London, GB
- Barbara Weber
Universität St. Gallen, CH