Emergent Behavior and Uncertainty in IoT-Enhanced Business Processes: Challenges and Future Directions

Marco Pegoraro[1], Sara Pettinari[2], Ivan Compagnucci[2], Marco Franceschetti[3] and Ronny Seiger[1]

[1]RWTH Aachen University, Faculty of Computer Science, Aachen, Germany
[2]Gran Sasso Science Institute, L'Aquila, Italy
[3]University of St.Gallen, Institute of Computer Science, St.Gallen, Switzerland

Abstract

IoT-enhanced business processes are characterized by high complexity due to heterogeneous actors, varying levels of autonomy among participating systems, continuously evolving execution contexts spanning the digital and physical worlds, and continuous event streams. In such settings, process behavior partially emerges only at runtime through complex interactions involving humans, IoT devices, physical objects, software systems, agents, and services. This complexity introduces partial observability, uncertainty, and runtime dynamics that are difficult to anticipate and that challenge traditional business process management (BPM) assumptions and systems. We discuss these challenges from three perspectives, addressing 1) uncertainty representation, 2) operationalization of IoT-enhanced processes, and 3) runtime management of emergent behavior. Based on a motivating scenario and an analysis of the state of the art, we identify open research gaps and outline short-, medium-, and long-term recommendations to shape a research agenda on emergent behavior in IoT-enhanced business processes.

Keywords

Business Process Management, Internet of Things, IoT-Enhanced Business Processes, Emergent Behavior, Uncertainty, Dynamics, Systems-of-Systems, Software Architecture

1. Introduction

IoT-enhanced business processes involve continuous interaction among process instances, physical devices (e.g., sensors, actuators), software services, and intelligent agents–creating an interplay between the physical and digital worlds [1]. This interaction makes process behavior more dynamic and less predictable than in traditional BPM settings because execution may depend on uncertain data, changing environmental conditions, real-time events, and autonomous decisions made by IoT components. Nowadays, more and more IoT systems can operate autonomously within their environments to provide useful services. When coming together in unanticipated ways, these self-contained *constituent* systems form *Systems-of-Systems* (SoS) exhibiting *emergent behavior* that is more than the sum of the individual parts [2]. This behavior can be classified as *expected* or *unexpected*, and, more importantly, as having *positive* or *negative* effects, where the latter should be avoided [3]. The main challenge is therefore to move beyond static, centrally controlled process models and to define approaches to modeling, execution, and runtime management that can represent uncertainty, support dynamic behavior, accommodate different interactions among systems, processes, and IoT devices, and respond to runtime changes [1].

The relevance of this problem is evident across domains in which business execution is closely tied to the physical environment through IoT/cyber-physical systems (e.g., smart manufacturing, logistics, healthcare, and smart cities). In these settings, process performance and correctness depend on the ability to monitor real-world events, adapt execution decisions, and dynamically coordinate heterogeneous IoT components in case of unanticipated situations and uncertainties. Addressing this problem requires

BP-Meet-IoT 2026: 10th International Workshop on Business Processes Meet the Internet of Things, co-located with EDOC 2026, September 15, 2026, Enschede, The Netherlands

pegoraro@pads.rwth-aachen.de (M. Pegoraro); sara.pettinari@gssi.it (S. Pettinari); ivan.compagnucci@gssi.it (I. Compagnucci); marco.franceschetti@unisg.ch (M. Franceschetti); ronny.seiger@rwth-aachen.de (R. Seiger)
0000-0002-8997-7517 (M. Pegoraro); 0000-0002-5548-9806 (S. Pettinari); 0000-0002-1991-0579 (I. Compagnucci); 0000-0001-7030-282X (M. Franceschetti); 0000-0003-1675-2592 (R. Seiger)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



published 2026-11-02

CEUR-WS.org/Vol-4310/paper6.pdf



Figure 1: Motivating scenario: emergent smart warehouse operations from the intertwined execution of multiple IoT-enhanced processes by heterogeneous actors. Figure generated using DALL-E by OpenAI.

mechanisms for representing uncertainties, monitoring and controlling process execution via self-adaptive models and digital twins at runtime, as well as reusable, open-source extensions that make BPM systems more flexible, context-aware, and suitable for IoT-integrated scenarios [4].

In this work, we elaborate on the emerging challenges in the development and implementation of evolving IoT-enhanced processes that must become increasingly dynamic and capable of coping with uncertainties in the context of SoS. Our discussion of managing emergent behavior and uncertainties thereby addresses the modeling, operationalization, and runtime phases of IoT-enhanced business processes. With these discussions, we pave the way for future research on IoT-enhanced business processes in real-world domains (e.g., distributed smart manufacturing, smart logistics, smart cities) that require, on the one hand, structured process-based control and analysis of complex systems, and, on the other hand, increasing autonomy of constituent systems to cope with environmental uncertainties during process executions.

The paper is structured as follows: Section 2 presents a motivating scenario. Section 3 decomposes the main challenge of managing emergence in IoT-enhanced business processes into three sub-challenges. Section 4 discusses the state of the art. Section 5 elaborates on identified gaps and opportunities. Section 6 presents our recommendations for future research activities. Section 7 concludes the paper.

2. Motivating Scenario: Sources of Uncertainty

To ground the discussion, we consider a representative IoT-enhanced business process scenario in a *smart warehouse* that involves the collaboration of heterogeneous actors, including human workers, robotic systems, digital information systems, software services, agents, and interconnected IoT devices. Fig. 1 illustrates this scenario. Rather than being orchestrated by a single, well-defined process and process management system, warehouse operations emerge from several intertwined IoT-enhanced processes and ad-hoc interactions that unfold at runtime. Several BPM systems and other information systems that manage only their local execution contexts might be involved. The processes may leverage different IoT interaction modes–active, hybrid, and passive–and must independently adapt to changes in the process context, which may be only partially known to all process participants. Moreover, process actors continuously influence physical and environmental conditions through their close interaction with the real world.

In addition to the internal operations, the scenario extends beyond the warehouse organizational

boundaries. For example, delivery-related activities may require interacting with external actors. In such cases, we must account for inherent uncertainty and only partial a priori knowledge about external actors' processes. These external actors (e.g., delivery robots) might be orchestrated by their own processes and operated by different companies or organizations. Thus, they might be constrained by security and privacy restrictions on the process knowledge and on the context they share. Moreover, external entities (e.g., humans on the warehouse floor) that are not explicitly modeled within individual IoT-enhanced processes can still influence the emergent process's execution, thereby introducing additional complexity. This can happen because these entities modify the context (e.g., the physical environment) in which the process unfolds. For example, an external delivery robot may position itself within the operating area of a robot loading items onto a conveyor belt, thereby obstructing its movement and disrupting its normal operation.

Here, uncertainty emerges from several characteristics of IoT-enhanced business processes:

- Many participating systems are closed and self-managed, exposing only limited interfaces and information while pursuing their own internal goals and missions with their own strategies in a local context. Therefore, their behavior cannot be (fully) anticipated, observed, or controlled by a single process owner. A central entity possessing a complete global view and knowledge of all processes and context is not available.
- The participating (constituent) systems are frequently highly heterogeneous and involve devices from different vendors with proprietary logic and decision mechanisms that cannot be exposed, e.g., for privacy reasons. This results in only partial views of the implementation of the participating processes and executing entities.
- The presence of actors such as humans that act outside of process control introduces additional uncertainty and variability, e.g., via spontaneous interventions, unforeseen interactions, and unpredictable decisions. These characteristics lead to emergent behavior that cannot be fully modeled a priori, nor can it be addressed through post-mortem analyses (e.g., process mining). Instead, it must be handled dynamically at process runtime – which is especially important in safety-critical settings where harm or damage must be avoided.

The combination of distributed control, heterogeneous autonomous systems, external interactions, and emergent behavior makes the smart warehouse a representative scenario for several contemporary challenges at the BPM-IoT intersection. In such dynamic settings, a complete and consistent view of the emergent process state and its context is neither available nor reconstructible. Therefore, interactions occurring under partial observability must be explicitly considered when modeling, operationalizing, and managing emergent IoT-enhanced processes at runtime. In the following, we elaborate on a subset of challenges, focusing on these modeling, operationalization, and runtime management aspects of emergence in IoT-enhanced business processes.

3. Sub-Challenges and Problem Decomposition

We divide the problem into three sub-challenges, SC1–SC3, aligned with the BPM lifecycle phases of process modeling, operationalization, and execution, since these phases are critical for managing uncertainty and emergent behavior *at runtime*.

3.1. SC1 – Representing Uncertainty and Dynamics in IoT-Enhanced Processes

In the *modeling* phase of the BPM lifecycle, the emergent behavior of IoT-enhanced processes poses central challenges that are not fully addressed by current modeling approaches. We discuss this aspect by examining specific threats to effective process modeling induced by emergent behavior.

- *Uncertain event data*: Historical process data may be affected by uncertainties caused by several factors, both inherent and caused by emergence [5]. Sensor readings may be noisy, delayed, missing, duplicated, incorrectly attributed to cases, or ambiguous with respect to the activity

they represent [6]. In this context, emergent behavior challenges the capabilities of established data-recording practices. As a result, process models need to represent not only events, but also confidence, provenance, ambiguity, and alternative interpretations; this is particularly crucial when analysts adopt data-driven modeling principles, such as process discovery.

- *Unknown unknowns*: IoT-enhanced processes are shaped by "unknown unknowns", the parts of a process that are unobserved (and, at times, unobservable). Missing sensors, unobserved human workarounds, environmental conditions outside the measurement scope, and interactions between devices may influence the process without leaving a clean trace. This means that process intelligence cannot rely only on what is recorded, but must also reason about blind spots, incompleteness, and the limits of observability.
- *Lack of semantics*: Devices, platforms, organizations, and physical contexts may each produce data according to their own local meanings, identifiers, time scales, and assumptions. When these environments interact, the same signal may have different implications, while similar events may not be semantically equivalent. IoT-enhanced business processes, therefore, need richer semantic mediation between local observations and global process understanding [7, 8].
- *Emergent drift:* IoT-enhanced processes may exhibit the gradual or sudden divergence between the intended process and the behavior produced by autonomous agents, automated devices, optimization routines, and human adaptation. Agent automation may improve local efficiency while unintentionally reshaping the overall process, creating new dependencies, shortcuts, risks, or coordination patterns.

3.2. SC2 – Operationalizing IoT-Enhanced Business Processes

IoT-enhanced business processes operate in environments characterized by continuous data streams, heterogeneous actors, real-time interactions, and inherent unpredictability [9, 10]. Under these conditions, processes can no longer be treated as static and predefined control-flow structures. Instead, their operationalization is challenged by the need to account for unexpected runtime events, changing contextual information, and continuous interactions among process engines, IoT devices, software services, and external systems [11, 9]. We discuss the operationalization of IoT-enhanced business processes, focusing on the main concerns that arise when BPM systems are brought into direct interaction with IoT environments to make these processes executable.

- *Dynamic, event-driven, and context-aware execution*: Traditional process execution assumes that process models can be enacted according to predefined control-flow structures. In IoT-enhanced settings, however, execution is affected by runtime events, such as device unavailability, sensor observations, and physical-world interactions, which may trigger relevant context changes [12]. IoT-enhanced business processes may be driven by continuous streams of events produced by sensors, devices, platforms, and external services, while execution decisions may depend on environmental conditions, device states, user behavior, location, timing, or other situational factors [10, 13, 14]. The challenge is that such events and contextual information may be frequent, heterogeneous, uncertain, delayed, missing, conflicting, distributed across different systems, or only available during execution. Process engines, therefore, have to cope with executions that may deviate from predefined assumptions and require runtime adjustment, while preserving process consistency and understandability.
- *Interaction and architectural integration*: IoT-enhanced business processes often involve multiple IoT devices, actuators, platforms, services, and organizational systems [15]. A key challenge concerns how process engines interact with these heterogeneous components during execution. Such interaction may be *passive*, when the process observes data streams and reacts to events produced by sensors or external systems; *active*, when the process invokes devices, actuators, or IoT services during execution; or *hybrid*, when both forms are combined within the same process. These interaction modes expose the limitations of isolated, monolithic, and ad hoc BPM solutions and architectures, raising challenges related to platform heterogeneity, communication protocols,

reliability, latency, and coordination between the digital and physical worlds. This also requires managing dependencies among components that may evolve and fail independently, or interact in unforeseen ways during execution.

- *Modeling, execution alignment, and reuse*: BPMN provides a rich and widely adopted notation for representing different types of process models, but its large set of notation elements and extensions also increases modeling complexity and requires clear usage practices [16]. In IoT-enhanced settings, a key challenge is aligning what is modeled at design time with what can be executed at runtime, especially when processes depend on dynamic events, external devices, contextual information, and platform-specific capabilities [1]. This challenge is intensified by the proliferation of solutions that are tied to specific domains or platforms, making it difficult to compare approaches, define reusable components (e.g., notation extensions), and build shared knowledge across systems. Open-source BPM systems are particularly relevant in this respect, but they also pose the challenge of defining standard, reusable extensions and shared practices to support IoT-driven process scenarios without becoming tied to specific use cases or platforms.
- *Dynamic and mobile execution systems along the Cloud-Edge continuum*: In traditional process execution settings, the BPM systems are assumed to be centralized and statically placed on local or Cloud-based servers to facilitate client-server interactions and process executions involving multiple participants. For IoT-enhanced business processes, we see the need for more pervasive and mobile, personalized–*ubiquitous*–BPM systems that are able to move along with the users–seamlessly transitioning between Cloud servers for resource-intensive and shared process executions and computations, to deployments on personal devices (e.g., smart phones) and closer to the edge for local sensor processing and actuations as part of the process executions [17]. Thereby, edge computing promises better support for real-time, safety, and privacy constraints through decentralized, local data processing and autonomous process execution, thereby reducing dependencies on constant, high-bandwidth Cloud connectivity [18].

3.3. SC3 – Managing Emergent Behavior of Dynamic IoT-Enhanced Business Processes at Runtime

We organize discussions on managing the emergent behavior of IoT-enhanced business processes during execution based on the *MAPE-K* feedback loop proposed for autonomous computing [19]. This architecture has proven applicable for implementing autonomous behavior needed to deal with unanticipated situations, uncertainties, and emergence through self-adaptation across different system contexts [20].

- *Monitor*: Process and system monitoring has to be implemented in a traditional way within individual BPMS and IoT systems. Thereby, sensors can provide rich context information to contextualize process executions [7]. However, when multiple BPM and IoT systems interact in unforeseen ways, monitoring needs to dynamically extend and adapt to multiple systems at runtime. This adaptation of monitoring is challenged by 1) limited a priori knowledge regarding what specifically to monitor, 2) changes in what to monitor at runtime, and 3) constrained data collection capabilities (e.g., for privacy and security reasons).
- *Analyze*: During analysis, it is determined if system and process operations are within expected, normal parameters. This analysis usually assumes predetermined rules that define *normal/desired* conditions and distinguish them from *critical* conditions that require intervention. When dealing with emergence during process execution, we face uncertainty about how these analyses should be performed, since situations only emerge at runtime, when interventions might be needed (e.g., to avoid *detrimental* emergent behavior [2]). Data analysis at runtime might require the dynamic synthesis and adaptation of analysis rules, as well as accepting incomplete knowledge as given to cope with changing situations and uncertainties.
- *Plan*: Planning is responsible for deriving an adaptation plan to change the current systems and process executions based on change requests from the analysis, bringing the system (of systems)

operations back into a desired state. Thereby, planning is challenged by uncertainties about what constitutes a desired state — knowledge that may be only partially available at design time. At runtime, novel, unanticipated situations might emerge under an open-world assumption, which can push state-of-the-art planning approaches to their limits.

- *Execute*: Execution carries out the action plan derived during planning to move operations into a desired state again. In highly complex SoS scenarios where constituent systems influence each other in unanticipated ways, the execution of each corrective action might lead to unforeseen effects on other systems and their environment. The execution is challenged by continuously changing local and global system states and contexts, as well as newly emerging ones that require constant, high-frequency situation analysis, evaluation, and re-planning. Still, execution has to maintain real-time and safety guarantees in these highly dynamic settings as we assume that the execution of a corrective action might also involve manipulating the physical world via actuators.
- *Knowledge*: The knowledge base contains all relevant information and data to enable the autonomous behavior as each of the four phases constantly consults and updates this knowledge during operations. Creating and maintaining this knowledge base is challenging, as we constantly face uncertainties and incomplete knowledge when novel situations arise. During process execution, knowledge might be available only locally to the responsible BPM system and the involved IoT systems, and be constrained to their execution contexts. Other systems might operate in parallel and in their vicinity, but knowledge is incomplete and not shared for different reasons (e.g., privacy or unanticipated situations). Larger, new execution systems and contexts might emerge, but it is unclear how the knowledge should be managed, shared, and distributed.

4. State of the Art

Representing Uncertainty and Dynamics in IoT-Enhanced Processes. Several approaches incorporate contextual information into IoT-enhanced execution by using ontologies, high-level events, or context models derived from IoT data to influence process behavior at runtime [7, 13, 10, 21]. Nevertheless, context is often treated as external information consumed by the process rather than as a first-class construct explicitly represented and managed within the execution model.

In some lines of work, context carries meta-information about data uncertainty itself. When available, information related to uncertainty can be leveraged to obtain more trustworthy process analysis [22]. Various post-mortem analysis methods for metadata-augmented event logs are available, including conformance checking and process discovery [23]. While this paradigm can address data uncertainty, the available techniques are foundational, and their application to IoT is in its infancy. Early work includes process querying in settings with anomalous timestamps [24]. Other approaches focus on *repairing* the data, rather than analyzing it natively: several error-correction techniques are available, generally model- or rule-based, for both data correction and imputation [25, 26].

Ambiguity stemming from IoT data processing, a specific source of uncertainty in BPM, is discussed by the authors in [27]. The authors argue that there is often a gap between the low-level data streams emitted by IoT devices and the associated business process data. Performing event abstraction of IoT data to close this gap might lead to ambiguities for several reasons [6] and requires an explicit representation of identified uncertainties for subsequent process analytics [28].

In summary, recent survey work identifies preprocessing, activity recognition/discovery, event abstraction, case identification, event-activity correlation, and event ordering as recurring steps in analysis [29, 30], which help address anomalous IoT data recordings. However, there is still limited work on a unified framework that represents IoT data uncertainty explicitly and carries it through repair, event-log construction, and downstream process-mining analysis.

Operationalizing IoT-Enhanced Business Processes. Research on the operationalization of IoT-enhanced business processes has initially addressed the problem from a modeling perspective, focusing on how IoT entities, data, events, and physical-world interactions can be represented within process

models. Early contributions mainly focused on extending process modeling languages to represent IoT-related elements and physical-world interactions. For instance, [31] extends BPMN with sensing and actuation tasks, physical entities, and real-world data objects, while [32] introduces dedicated tasks for ubiquitous process interactions, including sensors, actuators, readers, and collectors. Other approaches focus on more specific IoT execution settings: [33] focuses on operations executed by wireless sensor network devices, whereas [34] introduces constructs for specifying and processing event streams produced by smart sensors. These works make IoT interactions explicit at design time and help reduce the gap between process models and their runtime interpretation.

However, modeling IoT elements is only one part of operationalization. IoT-enhanced processes must also be enacted in environments characterized by continuous data streams, heterogeneous devices, and changing contextual conditions. This shift toward execution is reflected in recent evidence: among 84 analyzed approaches, 43 focus only on design, whereas 41 also address enactment or execution [9]. This indicates that operationalization is now recognized as a central concern, although the relation between design-time process models and executable IoT-aware systems remains only partially consolidated.

Given the event-driven nature of IoT-enhanced execution, several approaches have been proposed to support runtime event processing and event integration [35]. In these approaches, BPM engines are integrated with Complex Event Processing (CEP), stream processing, or event processing components to enable process instances to respond to real-time IoT data streams [36, 10]. Other contributions consider how contextual information, such as location, device states, or environmental data, can influence process execution and monitoring [13]. While these works support more responsive and data-driven execution, event and context handling is typically delegated to external components, highlighting the limited native support for IoT-aware execution in BPM engines.

Further work focuses on the architectural integration between BPM systems and IoT environments [17]. Existing solutions include service- and microservice-based approaches that abstract IoT devices as services [10], end-to-end frameworks that extend BPMN and integrate IoT-aware execution components [37, 13], and BPMN-driven frameworks for cyber-physical or robotic systems [38]. While these works all assume centralized BPM systems, the authors in [39] also discuss distributed process executions in IoT scenarios (e.g., involving robots) that require additional self-adaptation capabilities. These approaches demonstrate the feasibility of operationalizing IoT-enhanced processes, but they usually rely on framework-specific assumptions, execution architectures, and integration mechanisms.

As a consequence, the state of the art still lacks shared practices and reusable extension mechanisms to support the systematic operationalization of IoT-enhanced business processes across heterogeneous IoT settings. In particular, existing approaches offer limited support for migrating from IoT-aware process models to reusable, portable, and runtime-aware execution infrastructures capable of handling heterogeneous devices, dynamic events, contextual changes, and execution-time uncertainty.

Managing emergent behavior of dynamic IoT-enhanced business processes at runtime. Several existing contributions focus on adaptive business processes and BPM systems in traditional (non-IoT related) settings, e.g., with a focus on context adaptation [40] and self-adaptation mechanisms [30]. IoT-enhanced business processes do not play a significant role in these works. More IoT/CPS-related approaches investigate how running process instances can react to contextual changes, exceptional situations, or sensor-detected events, for instance, through feedback-control mechanisms such as MAPE-K loops [41, 42] or through broader BPM-IoT adaptation perspectives in centralized process executions [43, 44, 45], but also in decentralized, pervasive execution scenarios [39, 46]. However, adaptation is often addressed at the architectural level or through external mechanisms, rather than being explicitly governed within the BPM execution layer. Furthermore, these approaches focus on individual, self-contained IoT system contexts and local feedback loops. More complex SoS configurations and emergent behavior that require decentralized, dynamic–and potentially adaptive–self-adaptation mechanisms are not discussed.

Emergent behavior in connection with business processes, e.g., using process mining for emergent behavior analysis, is discussed in [47]. In a follow-up work, the authors propose an agent-based

simulator for the detection of emergent behavior in CPS [48]. While this research is highly relevant for observing and detecting emergent behavior in IoT-enhanced business processes within the monitor and analysis phases of the feedback loops, they do not extend to the reactive part, and they do not discuss runtime capabilities. On the other hand, approaches to address and enact emergent behavior in IoT systems (e.g., via *emergent configurations* [49], domain objects [50] or *Fog Computing* [51]) exist, but they do not link to (business) processes. A comprehensive investigation of emergent behavior in cyber-physical systems of systems can be found in [2]. This work constitutes a fundamental pillar for research on emergence in SoS and IoT. However, the link to IoT-enhanced *business processes* still needs to be created.

5. Gaps and Opportunities

Building on insights from state-of-the-art investigations, we identify current research gaps and propose novel research directions.

From predefined process models to runtime emergence: Traditional BPM approaches rely on predefined process models, assuming that relevant execution paths can be anticipated at design time. In IoT-enhanced environments, this assumption no longer holds. In such contexts, multiple sources of uncertainty arise: devices may fail, human workers may deviate from expected behavior, environmental conditions may change, and unforeseen events may occur. These *unknown unknowns* generate emergent behaviors that can not be fully anticipated during process modeling [3, 52]. Despite prior work on adaptive and flexible processes, existing BPM engines still typically require fully specified models and offer limited support for partial process definitions, dynamic composition, or runtime emergence. Accordingly, a key opportunity is to define modeling approaches that make uncertainty explicit, together with execution mechanisms that can support emergent behavior while preserving process consistency and understandability.

Uncertainty as a first-class citizen in self-adaptive systems: Uncertainties are ubiquitous in SoS contexts and we assume that it is rarely possible to resolve all uncertainties. Thus, uncertainty needs to be made explicit throughout all considered phases of the BPM lifecycle and, specifically, in the MAPE-K architecture. Accordingly, a key opportunity is to make uncertainty explicit not only within the BPM lifecycle but also across the MAPE-K phases, so that monitoring, analysis, planning, execution, and knowledge management can reason under incomplete or missing information.

Context as a first-class citizen: In IoT-enhanced processes, context not only describes the execution environment. It can also trigger, constrain, and redirect process behavior. Location, device state, environmental conditions, resource availability, and human-related factors may determine which activities are enabled, how decisions are made, and when adaptations are required. Yet, context is often treated as external data consumed by the process, rather than as a first-class element that shapes execution semantics and runtime behavior. Accordingly, a key opportunity is to define context models that explicitly capture the relation between contextual conditions and process behavior, including context validity, uncertainty, and impact on adaptation.

Adoption of unified ontologies and taxonomies: Shared ontologies and taxonomies provide a common vocabulary for linking low-level sensor data, contextual information, process concepts, and behavioral models, and thus are essential for analyzing IoT processes affected by uncertainty and emergent behavior [8]. Data-oriented taxonomies help characterize quality issues such as missing, ambiguous, noisy, or imprecise events, making assumptions explicit and comparable across studies. Model-oriented ontologies support the representation of activities, resources, objects, constraints, and expected interactions, enabling uncertain observations to be interpreted using domain knowledge rather than discarded. Together, they improve interoperability, support reproducible event-log construction,

guide repair and abstraction decisions, and enable distinguishing true process deviations from artifacts of sensing, correlation, or modeling limitations.

From discrete events to continuous event streams: BPM engines traditionally react to discrete events such as messages, timers, or service responses. In IoT-enhanced environments, execution may depend on continuous, high-frequency, heterogeneous, and uncertain event streams produced by sensors, devices, platforms, and external actors. Although CEP and stream-processing technologies are available, their integration with BPM engines is often externalized and architecture-specific [15]. A key opportunity is to clarify how event processing and process execution should interact, including event-process correlation, stream-aware execution, and scalability under continuous IoT data.

Distributed BPM systems for ubiquitous process executions: Moving away from centralized client-server architectures, BPM systems have to become more modular, distributed, and mobile to support decentralized and personalized executions of IoT-enhanced business processes. These process executions should facilitate migration between low-footprint, pervasive, and portable process engines on embedded edge devices (e.g., in containerized environments or WebAssembly modules), personal devices (e.g., smartphones and laptop computers), and highly distributed and scalable Cloud resources. This requires the development of 1) workflow engines that are able to operate in resource-constrained environments (e.g, on robots [53]) and 2) interoperable solutions that allow moving process instances between different types of BPMS provided and operated by and on different entities.

Continuous, emerging, high-frequency adaptation loops with safety constraints: In IoT settings and SoS, operations and context are constantly changing, creating unanticipated situations that need to be reacted and adapted to via feedback loops (e.g., implementing MAPE-K) at runtime. These feedback loops have to be used to manage all involved components and processes and they might have to be dynamically synthesized and adapted at runtime. Despite the flexibility and potential overhead introduced by these additional control loops, safety and real-time constraints imposed by IoT systems must be met. The implementation of AI-based agents appears to be a promising direction to realize the self-adaptation mechanisms, as they are capable of advanced reasoning for runtime decision making using large knowledge bases and world models. However, they cannot meet the current safety requirements of CPS/IoT systems, which require real-time decision-making and action.

From local to ad-hoc, global context models and knowledge: The knowledge base is an important pillar for implementing feedback loops to manage the emergent behavior of IoT-enhanced business processes at runtime. So far, knowledge is available only locally within the individual constituent systems. In emerging interactions in SoS scenarios, the knowledge bases have to be created and extended at runtime to allow for knowledge sharing among systems in order to agree upon a common *world model*, which could, for example, be based on a multi-faceted *digital twin* of the relevant systems and their business processes [4, 54]. However, privacy and security constraints of data imposed upon by individual systems have to be respected.

From ad-hoc adaptation to governed runtime adaptation: IoT-enhanced business processes must adapt to continuous context changes and event streams. This can be framed as a form of *self-modification*, where the presented gap concerns runtime *adaptation* rather than longer-term process *evolution* [11]. Runtime adaptation introduces risks related to process consistency, constraint violations, and unintended behavior. Although some engines and solutions support adaptability, adaptation is often ad hoc and context-dependent. Accordingly, a key opportunity is to define governed adaptation mechanisms that support controlled runtime changes, including adaptation policies, verification, rollback, recovery, and principles for deciding when human oversight is needed.

Meta-adaptations in highly dynamic System-of-Systems contexts: Currently, MAPE-K-based mechanisms for self-adaptation rely on static approaches to monitoring, analysis, planning, and execution, which work well in closed systems. When aiming to manage emergent behavior in SoS, these approaches may also have to become more decentralized, dynamic, and self-adaptive [55], i.e., each of the MAPE phases managing a system or a process might itself be managed by a feedback loop, leading to *meta-adaptations*. By treating the MAPE-based feedback loops as partial business processes themselves, these meta-adaptations can be realized in a unified, process-driven way, using the same process syntax and execution systems as the business processes or fragments managed by these control loops.

Shared architectures and practices: Existing BPM-IoT solutions often rely on modeling- and engine-specific extensions, platform-specific integrations, or domain-specific architectures. While these solutions demonstrate feasibility, they remain difficult to compare, reuse, and transfer across settings. Accordingly, a key opportunity is to define shared extension mechanisms, reusable integration patterns, and open reference architectures that clarify how BPM engines, IoT middleware, event processing, context management, and adaptation logic should interact.

6. Recommendations

Based on the identified gaps and opportunities, we recommend the following action items for future, short-term, medium-term, and long-term research and development activities. With every recommendation, we associate the specific sub-challenges (SC) introduced in Section 3 that it addresses.

6.1. Short-Term Recommendations

In the short term, research should focus on clarifying the assumptions and technical foundations needed to study emergent IoT-enhanced business processes. First, existing techniques for uncertainty-aware event data recording and publishing should be implemented and assessed, while uncertainty should also be investigated across the phases of MAPE-K feedback loops. This supports the representation and propagation of uncertainty across modeling and runtime management **[SC1, SC3]**.

Second, existing BPM engines should be analyzed and benchmarked for IoT-driven and event-based execution. This includes clarifying how BPM engines can interact natively with event-processing technologies, such as CEP and stream-processing platforms, and assessing the suitability and resource impact of current BPM systems for edge deployments **[SC2]**.

Third, future work should identify key sources of unpredictability and define initial first-class context models. At the same time, the implications of incomplete, local, and evolving knowledge for runtime management should be analyzed, including the role of digital twins in maintaining up-to-date knowledge bases that can be integrated while preserving privacy **[SC1, SC2, SC3]**.

Finally, requirements for runtime monitoring and adaptation to emergent behavior in IoT-enhanced business processes operating in continuously evolving execution contexts should be identified **[SC3]**.

6.2. Medium-Term Recommendations

In the medium term, research should move from assessment toward the development of concrete mechanisms. For uncertainty-aware modeling and analysis, process analysis techniques should be adopted that account for non-deterministic and probabilistic information, while also relaxing assumptions such as case-based and total-order event structures **[SC1]**.

At the execution level, new mechanisms are needed to handle uncertainty and emergent behavior at runtime. This includes defining explicit runtime adaptation policies, constraint-preservation mechanisms, and rollback strategies, as well as establishing structured approaches for integrating context models into execution logic **[SC1, SC2, SC3]**.

To improve operationalization, standardized extension points should be introduced at both modeling and execution levels. In parallel, low-footprint and standard-compatible BPM systems should be developed to support process execution on resource-constrained devices **[SC2]**.

For runtime management, MAPE-K feedback loops should be integrated as process-driven (*process-native*) concepts into IoT-enhanced business processes. This requires decentralized runtime monitoring mechanisms and privacy- and goal-oriented approaches for knowledge disclosure and distributed world-model construction across interacting systems **[SC3]**.

6.3. Long-Term Recommendations

In the long term, the community should work toward shared practices for emergent IoT-enhanced BPM. This includes rigorously defined semantics, taxonomies, and ontologies for process components, actors, and variables, as well as techniques for estimating emergent drift caused by automation, limited environmental control, and external agents **[SC1]**.

At the architectural level, scalable BPM architectures for IoT environments should be designed and validated. These architectures should integrate BPM engines, IoT middleware, event processing, adaptation mechanisms, and reusable integration patterns. They should also support the seamless migration of BPM systems and process executions along the Cloud-Edge continuum **[SC2]**.

The community should also promote reusable components and shared practices across BPM and IoT ecosystems, enabling adaptive, context-aware processes that can evolve continuously **[SC2, SC3]**.

Finally, runtime management should evolve toward distributed and meta-adaptive infrastructures. This includes distributed knowledge for large-scale emergent IoT-enhanced SoS, (AI) agent-based feedback-loop mechanisms with real-time decision-making and actuation guarantees, and meta-adaptive MAPE-K architectures capable of modifying their own feedback-loop mechanisms at runtime **[SC3]**.

7. Conclusions

IoT-enhanced business processes increasingly operate in dynamic environments characterized by heterogeneous actors, autonomous systems, and evolving execution contexts. In such settings, process behavior cannot be fully anticipated as it emerges through complex interactions among humans, IoT devices, and (possibly external) software systems. Emergent behavior in such system-of-systems contexts is characterized by partial observability and uncertainty, which challenge traditional BPM assumptions. In this paper, we discussed these challenges from three complementary perspectives grounded in the BPM lifecycle, addressing uncertainty representation, the operationalization of IoT-enhanced process execution, and the runtime management of emergent behavior during execution. Our analysis of the state of the art identified several research gaps. Based on these gaps, we proposed short-, medium-, and long-term recommendations to stimulate further research and development toward more adaptive, context-aware, interoperable approaches to IoT-enhanced business processes. Overall, we argue that effective management of emergent behavior requires shifting from static, centrally controlled processes toward self-adaptive, decentralized, and uncertainty-aware approaches that can cope with continuously evolving execution contexts, supported by shared practices. Future research should, therefore, focus on uncertainty-aware approaches to modeling, execution, and runtime management; distributed knowledge and world-model construction; decentralized, self-adaptive mechanisms and executions; autonomous and interoperable architectures and shared practices. The presented challenges, gaps, and recommendations contribute toward establishing a future research agenda at the intersection of BPM, IoT, and emergent SoS.

Acknowledgments

This work has been partially funded by the MUR (Italy) Department of Excellence 2023 - 2027 for GSSI. This work has received funding from the Swiss National Science Foundation under Grant No. 10002384

(*TaSSAreCt* project). This work has been supported by the Internet of Processes and Things (IoPT) community through discussions, datasets, and feedback.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-5.5 and Grammarly for grammar and spell checks. Figure 1 was also AI-generated, as stated in its caption. After using these tools/services, the authors reviewed and edited the content as needed and took full responsibility for it.

References

[1] C. Janiesch, A. Koschmider, M. Mecella, B. Weber, A. Burattin, C. Di Ciccio, G. Fortino, A. Gal, U. Kannengiesser, F. Leotta, et al., The internet of things meets business process management: a manifesto, IEEE Systems, Man, and Cybernetics Magazine 6 (2020) 34–44.

[2] H. Kopetz, A. Bondavalli, F. Brancati, B. Frömel, O. Höftberger, S. Iacob, Emergence in cyber-physical systems-of-systems (cpsoss), in: Cyber-Physical Systems of Systems: Foundations–A Conceptual Model and Some Derivations: The AMADEOS Legacy, Springer, 2016, pp. 73–96.

[3] D. Perez-Palacin, R. Mirandola, Uncertainties in the modeling of self-adaptive systems: A taxonomy and an example of availability evaluation, in: ACM/SPEC international conference on Performance engineering, 2014, pp. 3–14.

[4] F. Fornari, I. Compagnucci, M. C. De Donato, Y. Bertrand, H. H. Beyel, E. Carrión, M. Franceschetti, W. Groher, J. Grüger, E. Kilic, et al., Digital twins of business processes: A research manifesto, Internet of Things 30 (2025) 101477.

[5] M. Pegoraro, W. M. P. van der Aalst, Mining uncertain event data in process mining, in: International Conference on Process Mining (ICPM), IEEE, 2019, pp. 89–96.

[6] M. Franceschetti, R. Seiger, H. A. López, A. Burattin, L. García-Bañuelos, B. Weber, A characterisation of ambiguity in bpm, in: International Conference on Conceptual Modeling, Springer, 2023, pp. 277–295.

[7] Y. Bertrand, S. Veneruso, F. Leotta, M. Mecella, E. Serral, NICE: The native IoT-centric event log format for process mining, in: Internet of Things Meets Business Process Management: A Synergistic Integration, Springer, 2026, pp. 219–249.

[8] Y. Bertrand, C. Imenkamp, L. Malburg, M. Ehrendorfer, M. Franceschetti, J. Grüger, F. Leotta, J. Mangler, R. Seiger, A. Koschmider, et al., An object-centric core metamodel for iot-enhanced event logs, arXiv preprint arXiv:2506.21300 (2025).

[9] I. Compagnucci, F. Corradini, F. Fornari, A. Polini, B. Re, F. Tiezzi, A systematic literature review on IoT-aware business process modeling views, requirements and notations, Softw. Syst. Model. 22 (2023) 969–1004.

[10] P. Valderas, V. Torres, E. Serral, Modelling and executing IoT-enhanced business processes through BPMN and microservices, Journal of Systems and Software 184 (2022) 111139.

[11] A. Elyasaf, A. Metzger, S. Sardina, A. Senderovich, E. S. Asensio, N. Tax, Toward Self-Modifying Autonomous Business Process Systems, in: International Workshop on Process Management in Artificial Intelligence, PMAI, 2025.

[12] S. Schönig, L. Ackermann, S. Jablonski, A. Ermer, An integrated architecture for IoT-aware business process execution, in: International Conference on Business Process Modeling, Development and Support, Springer, 2018, pp. 19–34.

[13] L. Poss, S. Schönig, Location-aware business process modeling and execution, Software and Systems Modeling 24 (2025) 37–67.

[14] J. vom Brocke, S. Zelt, T. Schmiedel, On the role of context in business process management, International journal of information management 36 (2016) 486–495.

[15] S. Schönig, L. Ackermann, S. Jablonski, A. Ermer, IoT meets BPM: a bidirectional communication architecture for iot-aware process execution, Softw. Syst. Model. 19 (2020) 1443–1459.

[16] I. Compagnucci, F. Corradini, F. Fornari, B. Re, A Study on the Usage of the BPMN Notation for Designing Process Collaboration, Choreography, and Conversation Models, Bus. Inf. Syst. Eng. 66 (2024) 43–66.
[17] C. Chang, S. N. Srirama, R. Buyya, Mobile cloud business process management system for the internet of things: a survey, ACM Computing Surveys (CSUR) 49 (2016) 1–42.
[18] R. Seiger, F. Niebling, T. Schlegel, A distributed execution environment enabling resilient processes for ubiquitous systems, in: 2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS), IEEE, 2014, pp. 220–223.
[19] S. R. White, J. E. Hanson, I. Whalley, D. M. Chess, J. O. Kephart, An architectural blueprint for autonomic computing, in: International Conference on Autonomic Computing (ICAC), IEEE Computer Society, 2004, pp. 2–9.
[20] H. Muccini, M. Sharaf, D. Weyns, Self-adaptation for cyber-physical systems: a systematic literature review, in: Proceedings of the 11th international symposium on software engineering for adaptive and self-managing systems, 2016, pp. 75–81.
[21] W. Ochoa, J. Legaristi, F. Larrinaga, A. Pérez, Dynamic context-aware workflow management architecture for efficient manufacturing: A ROS-based case study, Future Generation Computer Systems 153 (2024) 505–520.
[22] M. Pegoraro, Probabilistic and non-deterministic event data in process mining: Embedding uncertainty in process analysis techniques, in: CAiSE 2022 Doctoral Consortium, volume 3139, 2022, pp. 37–46.
[23] M. Pegoraro, M. S. Uysal, W. M. P. van der Aalst, PROVED: A tool for graph representation and analysis of uncertain event data, in: International Conference on Applications and Theory of Petri Nets and Concurrency, Springer, 2021, pp. 476–486.
[24] N. Busany, H. V. D. Aa, A. Senderovich, A. Gal, M. Weidlich, Interval-based queries over lossy IoT event streams, ACM Transactions on Data Science 1 (2020) 1–27.
[25] M. Shirali, Z. Ahmadi, C. Fernández-Llatas, J. L. Bayo Montón, G. Di Federico, An interactive error-correcting approach for IoT-sourced event logs, ACM Transactions on Internet of Things 5 (2024) 1–30.
[26] A. Seeliger, M. Schreiber, F. Giger, J. Metternich, M. Mühlhäuser, Inferring missing event log data from IoT sensor data-a case study in manufacturing, in: International Conference on Business Process Management, Springer, 2024, pp. 232–248.
[27] M. Franceschetti, R. Seiger, L. García-Bañuelos, B. Weber, ProAmbitIon, reloaded: A two-year retrospection, Research Projects Exhibition@ CAiSE 2025 (2025).
[28] M. Franceschetti, D. M. Buchegger, R. Seiger, B. Weber, Toward IoT-based process analytics: Extending event knowledge graphs with ambiguity, in: International Conference on Business Process Modeling, Development and Support, Springer, 2025, pp. 176–192.
[29] E. Brzychczy, M. Aleknonytė-Resch, D. Janssen, A. Koschmider, Process mining on sensor data: a review of related works, Knowledge and Information Systems 67 (2025) 4915–4948.
[30] J. Oukharijane, I. Ben Said, M. A. Chaâbane, R. Bouaziz, E. Andonoff, A survey of self-adaptive business processes, in: Int. Business Information Management Association Conference, Seville, Spain, 2018, pp. 1388–1403.
[31] S. Meyer, A. Ruppen, C. Magerkurth, Internet of things-aware process modeling: Integrating IoT devices as business process resources, in: Advanced Information Systems Engineering, volume 7908 of *Lecture Notes in Computer Science*, 2013, pp. 84–98.
[32] A. Yousfi, C. Bauer, R. Saidi, A. K. Dey, uBPMN: A BPMN extension for modeling ubiquitous business processes, Information and Software Technology 74 (2016) 55–68.
[33] C. T. Sungur, P. Spiess, N. Oertel, O. Kopp, Extending BPMN for wireless sensor networks, in: Conference on Business Informatics, IEEE, 2013, pp. 109–116.
[34] S. Appel, P. Kleber, S. Frischbier, T. Freudenreich, A. P. Buchmann, Modeling and execution of event stream processing in business processes, Information Systems 46 (2014) 140–156.
[35] Y. Kirikkayis, F. Gallik, R. Seiger, M. Reichert, Integrating IoT-driven events into business processes, in: International Conference on Advanced Information Systems Engineering, 2023, pp. 86–94.

[36] R. Seiger, L. Malburg, B. Weber, R. Bergmann, Integrating process management and event processing in smart factories: A systems architecture and use cases, Journal of Manufacturing Systems 63 (2022) 575–592.
[37] Y. Kirikkayis, F. Gallik, M. Winter, M. Reichert, BPMNE4IoT: a framework for modeling, executing and monitoring IoT-driven processes, Future Internet 15 (2023) 90.
[38] F. Corradini, S. Pettinari, B. Re, L. Rossi, F. Tiezzi, A BPMN-driven framework for multi-robot system development, Robotics and Autonomous Systems 160 (2023) 104322.
[39] R. Seiger, S. Herrmann, U. Aßmann, Self-healing for distributed workflows in the internet of things, in: 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), IEEE, 2017, pp. 72–79.
[40] G. Hermosillo, L. Seinturier, L. Duchien, Creating context-adaptive business processes, in: International Conference on Service-Oriented Computing, Springer, 2010, pp. 228–242.
[41] R. Seiger, S. Huber, P. Heisig, U. Aßmann, Toward a framework for self-adaptive workflows in cyber-physical systems, Software & Systems Modeling 18 (2019) 1117–1134.
[42] L. Malburg, M. Hoffmann, R. Bergmann, Applying MAPE-K control loops for adaptive workflow management in smart factories, Journal of Intelligent Information Systems 61 (2023) 83–111.
[43] D. Domingos, F. Martins, R. Martinho, M. Silva, Ad-hoc changes in IoT-aware business processes, in: 2010 Internet of Things (IOT), IEEE, 2010, pp. 1–7.
[44] F. Leotta, A. Marrella, M. Mecella, IoT for BPMers. challenges, case studies and successful applications, in: International Conference on Business Process Management, Springer, 2019, pp. 16–22.
[45] A. Marrella, M. Mecella, S. Sardina, Intelligent process adaptation in the SmartPM system, ACM Transactions on Intelligent Systems and Technology 8 (2016) 1–43.
[46] C.-Y. Li, A. Antonov, T.-H. Huang, M. Pegoraro, S. S. F. Lau, M. X. B. Hiew, I. Y. N. Law, B. Drescher, W. M. P. van der Aalst, Cargo distribution analysis: a process mining approach, Process Science 3 (2026) 4.
[47] R. H. Bemthuis, M. Koot, M. R. Mes, F. A. Bukhsh, M.-E. Iacob, N. Meratnia, An agent-based process mining architecture for emergent behavior analysis, in: 2019 IEEE 23rd International Enterprise Distributed Object Computing Workshop (EDOCW), IEEE, 2019, pp. 54–64.
[48] R. Bemthuis, M. Mes, M.-E. Iacob, P. Havinga, Using agent-based simulation for emergent behavior detection in cyber-physical systems, in: 2020 Winter Simulation Conference (WSC), IEEE, 2020, pp. 230–241.
[49] F. Alkhabbas, R. Spalazzese, P. Davidsson, Architecting emergent configurations in the internet of things, in: 2017 IEEE International Conference on Software Architecture (ICSA), IEEE, 2017, pp. 221–224.
[50] F. Alkhabbas, M. De Sanctis, R. Spalazzese, A. Bucchiarone, P. Davidsson, A. Marconi, Enacting emergent configurations in the iot through domain objects, in: International Conference on Service-Oriented Computing, Springer, 2018, pp. 279–294.
[51] D. Roca, R. Milito, M. Nemirovsky, M. Valero, Tackling IoT ultra large scale systems: Fog computing in support of hierarchical emergent behaviors, in: Fog Computing in the Internet of Things: Intelligence at the Edge, Springer, 2017, pp. 33–48.
[52] G. Filippone, J. A. Piñera García, M. Autili, P. Pelliccione, Handling uncertainty in the specification of autonomous multi-robot systems through mission adaptation, in: International Symposium on Software Engineering for Adaptive and Self-Managing Systems, 2024, pp. 25–36.
[53] R. Seiger, S. Pettinari, L. Malburg, Autonomous mobile robots with business process management systems at the edge, in: European Conference on Software Architecture, Springer, 2026.
[54] I. Compagnucci, B. Re, E. S. Asensio, M. Snoeck, A Digital Process Twin Conceptual Architecture for What-If Process Analysis, in: Enterprise Design, Operations, and Computing. EDOC 2024 Workshops, Lecture Notes in Business Information Processing, 2024, pp. 373–388.
[55] R. De Lemos, H. Giese, H. A. Müller, M. Shaw, J. Andersson, M. Litoiu, B. Schmerl, G. Tamura, N. M. Villegas, T. Vogel, et al., Software engineering for self-adaptive systems: A second research roadmap, in: Software Engineering for Self-Adaptive Systems II, 2013, pp. 1–32.