Software and Systems Modeling
https://doi.org/10.1007/s10270-024-01224-7

SPECIAL SECTION PAPER

Location-aware business process modeling and execution

Leo Poss[1] · **Stefan Schönig**[1]

Received: 16 October 2023 / Revised: 28 May 2024 / Accepted: 7 October 2024

Abstract

Locally distributed processes include several process participants working on tasks at different locations, e.g., craftspeople working on construction sites. Compared to classical IT environments, new challenges emerge due to the spatial context of a process. Real-time location data from Internet of Things (IoT) devices can help businesses implement more efficient and effective processes through business process management (BPM). However, only small parts of existing research have touched on those advantages, while the architecture and implementation of actual executable location-aware processes area has only been vaguely considered. Therefore, we introduce and present a non-exhaustive list of patterns for using location data in BPM while also including an actual implementation of a location-aware approach using a multilayer system architecture based on standard BPM technology. These can be used to leverage the location perspective of process entities as contextual data in BPM.

Keywords Process execution · Location-awareness · Distributed processes

1 Introduction

Business process management (BPM) is a management approach to goal-driven design and the optimization of diverse and cross-organizational business processes, providing methods and tools to support the design, execution, management, and analysis of processes [1].

Different new areas, such as digital marketing, where the location of the customer can be used (e.g., location-dependent pricing or digital nudging—e.g., ads based on the current location or other data based on this), or the use of locations of customers and employees by large e-commerce companies and package deliverers (utilizing location for routing optimization as well as for customer benefits, such as knowing when a package will arrive), or their use in crisis response/paramedics (where the current location of the rescue team in comparison with the victim's location is crucial) could benefit from an integrated approach providing support and easing the use of location data. However, on the other hand, established areas not directly tied to the classical use of business processes in production sites or factories but in service-provisioning could benefit from using location data within business processes and value creation (e.g., craftspeople relying on mobile and locally distributed processes, where several process participants work on subsequent tasks at different locations).

All those processes come with different challenges compared to other business processes, where the spatial context is often disregarded due to negligible distances between fulfillment locations [2]. Looking back at already existing approaches, location information can be used to adapt the price or change the content of an ad space, where the location of a user is part of a more extensive set of decision variables and can either be used to improve the user experience or improve the value generation on the companies side. On the other hand, with crisis response and craftspeople at a construction site, the location of the target entity is the core of the process.

Historically, BPM focuses on classical IT processes with little to no spatial context, mainly on the automation and support of tasks through machines and their standardization. The Internet of Things (IoT) describes a world where every object can connect to a data network with its own identity [3], enabling electronic hardware to communicate remotely with

Communicated by Han van der Aa and Rainer Schmidt.

✉ Leo Poss
leo.poss@ur.de

Stefan Schönig
stefan.schoenig@ur.de

[1] University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Bavaria, Germany

physical objects. The data obtained from the IoT devices representing the locations of different entities involved describe their conditions and circumstances [4, 5].

As previously outlined, the prevailing solutions are predominantly rigidly programmed, provided by specific companies, and customized for particular use cases. BPM and its toolchain offer a solution to conceptualize such location-aware functionality by providing a generic toolkit for implementing and leveraging location data in any mobile and distributed business process open for any process-aware system and organization. Despite those practical applications, existing research in BPM has mainly focused on different facets of location awareness, such as process adaptation, modeling, and mining [6, 7], or managerial considerations in the realm of context-aware processes [8, 9].

The description and implementation of an executable system to capture and process location data for business processes are either neglected, only vaguely contained, or described for a closed environment scenario, such as a smart factory [10]. In particular, to the best of the authors' knowledge, the allocation of concrete actors to tasks based on location data within BPM[1] and reordering the task lists of multiple process instances based on distance is still an open research issue. The described research gap poses the following research questions:

RQ1. What potential patterns exist for utilizing the location context of actors and entities in BPM?
RQ2. How can these identified patterns be effectively employed during the modeling and execution phases of the business process?

We fill this gap in the following sections by introducing different patterns for the use of IoT-based location information in BPM (and the needed operators) as well as an implementation of an executable and well-defined location-aware approach for process modeling and execution (see Fig. 1). Specifically, we outline a multilayer system architecture grounded in standard BPM technology. We implement a comprehensive suite of location-aware patterns applicable to different real-world application areas that enhance mobile distributed processes. These patterns include: *(i)* location-based task allocation, *(ii)* location-based automated start and termination of tasks and process instances as well as location-event-based dispatching of tasks and restricting the execution based on location, and *(iii)* location-based reordering of tasks.[2] The different patterns have been implemented and evaluated using common real-life distributed processes from crafts businesses, including a repair process at a customer location, the rental of devices like heating units, or the delivery of monitored goods between different branches, each combining the potential of the use of location data with an executable implementation to adapt to other real-life processes.

This article is an extension of the conference publication [13]. It extends the original article in various ways:

(i) The presented patterns have been extended in different directions.
(ii) The background on location has been extended, and a formal description of used operators has been added.
(iii) The prototypical implementation changed fundamentally and is presented in more detail.
(iv) The related work section has been extended and discussed more thoroughly.
(v) The evaluation section has been extended and includes presenting and discussing three implemented real-life use cases from our research project.

The remainder of this paper is organized as follows. In Sect. 2, we present the background and related work w.r.t. location-aware BPM, as well as our research methodology after that Sect. 3 is dedicated to establishing the foundations for information generation and processing from location data. After that, we present the different patterns for the use of location in BPM in Sect. 4.

To augment the understanding of our contributions, Sects. 5 and 6 give a brief overview of the BPMN extension LABPMN from [12] that enables the use of the presented patterns for BPMN as the de facto standard for process modeling and provides one possible implementation, and classify the patterns within the well-established data patterns of [14]. The prototypical implementation details, supporting the integration of location data in process modeling and execution, are explicated in Sect. 7.

Furthermore, Sect. 8 provides insights into real-world applications, illustrating the application of these patterns in three distinct processes from partners of a research project within crafts businesses—namely repair processes at customer locations, device rentals, and monitored goods delivery between different branches and one process from emergency services also inherently dependent on location information adapted from [15] showcasing the use of the patterns in another application area. Finally, the paper concludes with a summary and outlook in Sect. 9.

[1] The algorithmic optimization of an allocation problem and other optimization problems, e.g., for process planning, are existent and can be used for the actual implementation. [11], for example, describes the classic house allocation problem with an example of allocating three agents to three different tasks, which could be used for the allocation pattern presented below. Here, the focus is on the different patterns and how they can be used, independently of actual algorithms and implementation choices

[2] For LABPMN, the matching BPMN extension extending the metamodel for the inclusion of location, see [12]

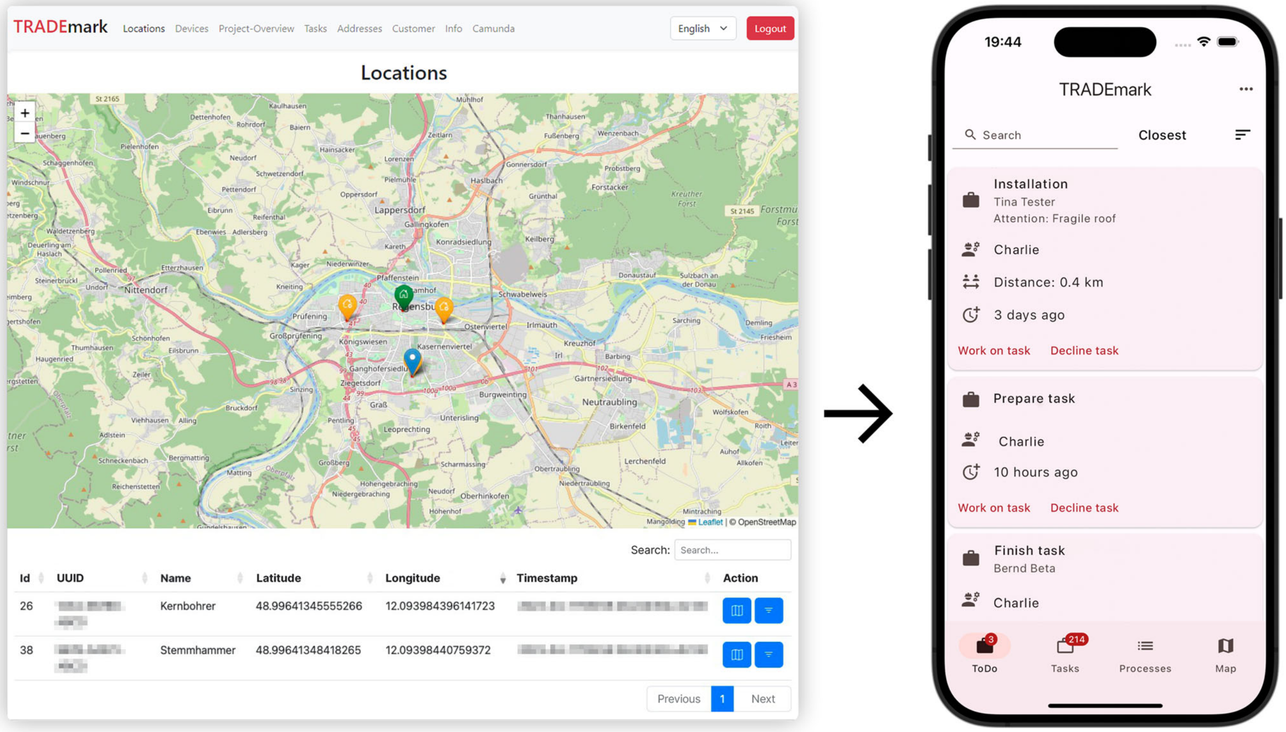


Fig. 1 Using location information to dynamically adapt mobile applications in business process execution (prototypical implementation presented in Sect. 7)

2 Theoretical background and related work

We commence with foundational knowledge regarding business processes, process context, and location awareness and proceed to present related work. This review aims to highlight the existing research gap concerning the integration of location data from IoT devices and BPM.

2.1 Background on context- and location-aware business processes

Based on the research field of business process management (BPM), the following sections will explain the connection to the Internet of Things (IoT), the use of contextual information during the execution of business processes, and location data as a considerable part of context information and its connection to BPM.

Business Process Management A *business process* is a series of activities to achieve a specific business outcome [1]. *Business Process Management (BPM)*, on the other hand, is a management approach aimed at designing, optimizing, and overseeing diverse cross-organizational business processes. This is achieved by using methods and tools that support the design, execution, management, and analysis of these processes. By leveraging BPM, organizations can identify opportunities for improvement and align their processes with the needs of both clients and their objectives. While standardization may seem advantageous initially, excessive standardization can impede the management of external influences. While this approach may work well for a production line, it hinders the optimization of mobile work processes. As stated in [16], "the management of processes should fit the process nature." In the case of a production line, external data may be less relevant from a process perspective and, therefore, disregarded, while mobile processes are heavily dependent on various locations.

Business processes can be represented by one or more *business process models*, consisting of activity models and instructions detailing how the processes should be executed. These models serve as a blueprint for real-world scenarios and cases, providing a structured representation [17].

Several modeling languages exist for this purpose, with *business process model and notation (BPMN)* [18] being the widely accepted current standard. BPMN incorporates various standardized elements that enhance both human and machine readability, enabling machine-supported execution of business processes. Furthermore, BPMN allows for the extension of the meta-model to introduce new functionality that is not part of the standard's core features. This extension capability can be employed to directly reintroduce logic necessary for the following patterns into the models, visually representing the actions during execution.

Over the years, the extension mechanism introduced in BPMN in 2011 [19, 20] has facilitated the development of numerous specialized extensions tailored to various application domains. These extensions enhance BPMN's functionality and adaptability to meet the specific requirements of diverse real-world scenarios (as discussed in [2, 19]). The reuse and enhancement of standard BPMN elements offer advantages such as standardization and widespread tool support. The BPMN extension mechanism is closely tied to the Meta Object Facility (MOF) meta-model [21], which separates the meta-model, process model, and process instances. This separation provides enhanced flexibility through interchangeable model properties and parts. In other words, extending individual elements of the BPMN meta-model does not strictly necessitate the modification of standard BPMN elements.

Finally, it is essential to emphasize that business process management (BPM) is not a one-time activity; rather, it constitutes an ongoing cycle with various versions documented in the literature, as elucidated in [22]. This work adopts the *BPM life cycle* framework delineated in [1], which is derived from Deming's PDCA cycle (Plan-Do-Check-Act). The patterns presented in this context emphasize the use of location information during process execution, as opposed to focusing on the incorporation of contextual information such as sociocultural surroundings and organizational resources during process modeling (cf. [8, 16]).

IoT-aware BPM

The *Internet of Things (IoT)* concept envisions a world where every object can connect to a data network and possesses its unique digital identity [3]. This digital infrastructure allows electronic hardware to communicate remotely with physical objects. Bridging the gap between human actors and electronic devices is made possible by tightly integrated systems that leverage data from IoT devices to enhance efficiency and effectiveness and enable data-driven work. IoT data are a foundational technology that provides context information and data that can be seamlessly integrated into business process management systems (BPMS). Using IoT data for process execution, monitoring, and analysis can offer a more comprehensive perspective on business processes. Incorporating real-time data from devices and sensors into BPM technology provides businesses with added value by reducing costs and improving efficiency.

The Internet of Things encompasses the use of sensors, devices, and networks to collect, transmit, and process real-time data from the physical world. This capability facilitates the creation of intelligent environments capable of adapting to the needs of users and organizations. The data collected through IoT can be leveraged to establish intelligent environments that respond to both user and organizational requirements, thereby enhancing process efficiency and effectiveness [23, 24]. In essence, the Internet of Things fundamentally enables the development of new functionalities that would otherwise be unattainable.

Context-aware BPM

Data acquired from IoT devices, representing various locations of the entities involved, provide information on the conditions and circumstances of these entities, making it an integral component of the *process context* [4]. In the field of *Human-Computer Interaction (HCI)*, the concept of context has been a focal point of research for numerous years, resulting in various definitions, the distinguishing factor being the relationship involved. In HCI, the relationship between an actor and a system becomes contextualized, whereas, in the context of a *business process management system (BPMS)*, we contextualize the entity incorporated within a process instance's execution. Schilit et al. (1994) define a *context-aware system* as a system capable of detecting and responding to environmental changes, while [5] define *context-awareness* as an application's ability to adapt to user context, with the *user's context* denoting the circumstances or situations in which a computing task occurs.

[25] describes *location-based services*, predating their connection to business processes and the practical use of the IoT. These services are a subset of context-aware services based on two context levels. The *primary context* encompasses factors such as time, location, identity, and activity, which are derived from sensors. In contrast, the *secondary context* level includes personal, technical, social, physical, and spatial contexts, which can be deduced or filtered from the primary context data [25].

Location-aware BPM

[25] define *spatial locations* as based on well-defined reference systems that subdivide a geographical area into units of a common shape and size, most commonly specified as a numerical explicit representation in a coordinate system (e.g., latitude and longitude), while in the real world, we commonly use street names and numbers (addresses) for defining a location of an entity. With a possible loss in precision, both formats can be converted into each other using geocoding. While addresses are more human-readable, they often need to be more accurate, considering both timeliness and accuracy and preventing numerical calculations on location information (cf. Section 3). Following the earlier differentiation of primary and secondary context, in this work *IoT-data* will refer to all data obtained from IoT devices, which can be *location data* (data to represent a location, primary context), while *spatial data* include the location of an entity, as well as other information based on the location of an entity (secondary context, e.g., the distance between two entities).

In the context of *complex event processing* (CEP), [26] distinguishes between different types of context: temporal, state-oriented, segmentation-oriented, and spatial. For this research, only the spatial classification is essential, which gets further divided into three different parts, where spatial information can be used as part of the generation of events [26]: (i) *Fixed location*: Describes a fixed location, such as the position of a customer or the company office, (ii) *Entity distance location*: Represents locations within a certain radius of an entity, for example, geofencing, (iii) *Event distance location*: Similar to entity distance location, but describes the radius drawn around an event happening (e.g., an accident) as the basis to include subsequent events. Another differentiation that must be made for the location information of the entities is the primary difference between *dynamic* and *static locations*. At the same time, the latter remains the same, and the first can change over time.

Although the concepts and goals of CEP and BPM have been focused on different application areas, within the context of IoT, they share various similarities, and both approaches can be combined [27].

Finally, we define the term *mobile distributed process* as a process involving multiple actors operating on sequential tasks at various locations, *implicitly constraining* the task's execution to a specific location, e.g., a craft business employing multiple workers working on different construction sites. This leads to a distinct set of problems compared to classical (IT) environments of business processes, where the spatial context of a process is less critical, or the distances between places of fulfillment are pretty small and might be negligible.

2.2 Related work

The following section describes currently existing approaches for combining BPM and IoT and using location data or parts of a possible future approach.

For the literature review, we formulate and apply the following search string, which encompasses the primary areas of interest, business processes, and location. The search string had to be present in at least one of the three components: title, abstract, or keywords:

(BPM OR "business process" OR "workflow*") AND (location* OR spatial OR geographic* OR context*)*[3]

Deliberately absent from the search string were terms directly associated with the Internet of Things (IoT) and related terms, such as sensor networks or cyber-physical systems. This omission accounts for discrepancies in terminology usage over time, which could result in older publications using the terms differently or including publications primarily focusing on hardware-related research. In a previous attempt to retrieve related works, we experimented with incorporating all combinations of context with IoT, but this approach failed to include several relevant publications.

The selected digital libraries for our search comprised widely used repositories in the fields of Computer Science and Information Systems, including the ACM Digital Library, IEEE Xplore, EBSCO, Science Direct, AIS eLibrary, as well as other publications identified through forward and backward searches to lower the possible miss rate on relevant articles. Different existing SLRs and meta-publications on the topic, such as [19, 29, 30] were used to add possibly missing publications. Additionally, we limited our review to articles published in English and German and, following the methodology of [29], we filtered the data to only include publications after 1999. This timeframe was chosen to encompass possible approaches predating the standardization of BPM in 2007 but postdating the initial use of the term Internet of Things around that year.

The PRISMA Statement with the different steps and selections, criteria, and results can be found in Fig. 2. The contents of the identified relevant literature are listed in the concept matrix in Table 1, giving an overview of the definition of the corresponding problem, as well as the goal, the chosen approach, and the result—independent of their original application domain [31]. Individual approaches can be classified into four non-exclusive categories, namely: *graphical modeling extensions* where the spatial perspective aims to explicitly model locations in business processes, *conceptual modeling extensions* where new use cases are presented (mainly the location-based restriction of the execution of tasks, e.g., in [32, 33] as well as research considering the *architecture of BPMSs* and the *use of spatial information to derive new functionality*, e.g., the use of location data from event logs to get an overview of task locations [7] or the visualization of work items and resources [34]. Note that even if the approach of [32] is conceptually defined and executable, the author and others like [33] and [35] only consider location constraints as use for spatial information during process modeling or execution. [2], on the other hand, gives an overview of related work for combining IoT and BPM, but not specifically for location data or the use of context information. They determine that most extensions avoid handling data, thus disregarding the core of the connection between location and their use for business processes.

Graphical Modeling for Location-Aware BPM

In reality, many business processes are location-dependent and should be executed differently based on the specific locations of the entities involved. However, current business process modeling languages do not inherently support incorporating location information into business processes. As a

[3] The asterisk notation, utilized with supporting libraries, ensured the inclusion of all results beginning with the respective search term

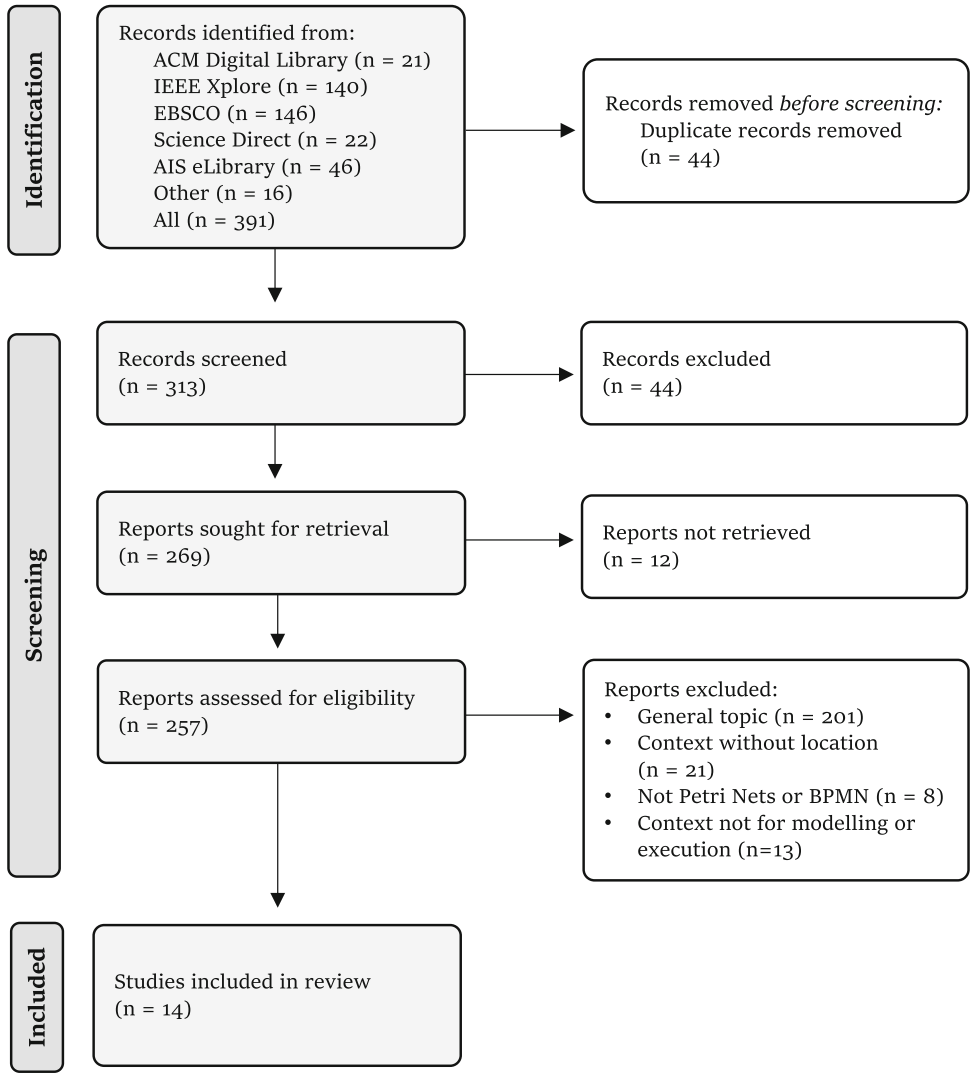


Fig. 2 PRISMA statement [28] of our literature review

result, the processes modeled using these languages often do not accurately reflect real-world business processes [36].

[37] focus on extending BPMN events to better capture events in IoT scenarios. Their extension is not limited solely to location events but more generally reuses `data objects` mapped explicitly to IoT resources. They maintain the basic structure of events, including the three different types of events (start, intermediate, and end) and the two flavors (catching and throwing). The given event definition in BPMN determines the nature of the event, whether it is a message event, a timer event, or a different type. Next to reusing existing elements, they also introduce a new type of event, the location event, marked with a pin icon that can describe either the current position or the movement of devices or physical entities. Based on IoT-A [38], they extend the event definition to include sections related to fault tolerance, specifying the maximum number of retries allowed to receive requested data and information quality, which restricts the data to be used. However, no information was provided on the extension of the BPMN meta-model (which is one of the requirements of the standard [18]), as well as an overview of the actual concepts for the actual use of location.

Conceptual Modeling Extension for Location-Aware BPM

Extending existing business process modeling notations with graphical elements alone is insufficient to make business processes truly location-aware. Business processes need both visual representation and machine-readability for their execution. Additionally, new concepts must be developed and integrated into these notations to respond to location information effectively.

One such concept is *Location Constraints* (LCs), utilized by [33] among others. An LC can be associated with an activity and defines where the activity must be performed or where it is prohibited. The former is termed a positive constraint, while the latter is a negative constraint. Within the publication, a two-dimensional location model is introduced that describes the spatial attributes required for defining loca-

Table 1 Related work concept matrix following [31]

	Goal	Approach/procedure	Outcome	Drawback	Notation	Executable?	Graphical	Conceptual	Architectural	New functionality
[39]	Develop a system to effectively handle constraints of network connectivity and mobility of workflow resources	Two-level approach: Global level: Dynamic assignment of activities based on, e.g., resource location. Local level: Additional resource attributes to prioritize activities for participants	Resource management by custom work list handler, supporting late and dynamic binding to location-dependent activities. Integration of map view	Dated approach focusing on constraints in mobile connectivity, only worklist reordering	–	●	○	○	●	●
[40]	Modeling location-aware business processes based on architectural primitives	Business-rule driven architecture with semantic coordination and location primitives. Process flow: Business rules restricting the execution of an activity	Influence of location laws on activities based on predefined location	Not executable, handles only location constraints	–	○	○	●	○	○
[33]	Include location constraints in BPM	Location constraints on activities at design time and dynamic constraints during run time. Graphical symbols and extension of the BPMN meta-model	Extension of the BPMN meta-model	Not executable, handles only location constraints	B	○	●	●	○	○
[41]	System architecture for location information in process optimization within a factory of things	Focus on the provision of location information to location-aware client applications. Externalize location rules from the source code into a knowledge base. System architecture handling aggregation, mapping, and allocation of location information	Conceptual idea to location-aware application architecture	Not executable, processes event logs to gain insights of execution locations	–	○	○	○	●	○
[42]	Comparison of business process notations regarding location aspects	Comparison of adaptions of notations, attempt to create notation for mobile information systems from scratch	Suggestion on the visual appearance (color-fill, icon, etc.) of location aspects of existing notations	Not executable, graphical representations only	B/P	○	●	○	○	○

Table 1 continued

	Goal	Approach/procedure	Outcome	Drawback	Notation	Executable?	Graphical	Conceptual	Architectural	New functionality
[34]	Decision support for choosing next work task	Map metaphor and distance notions to visualize work items and resources. Distance notions for priority of work items based on different factors, e.g., location. Visualization is split into work items and resource layers	Work list visualization framework based on maps and distance notions implemented as YAWL component	Decision support system for worklist reordering, location only	–	●	○	○	○	●
[35]	Shorten wait times in the cardiac care process	Combination of location system and BPM by complex event processing (CEP)	Abstract location-aware BPM system architecture. Implementation prototype of architecture	Different application area, CEP for process improvement, fixed process	B	●	○	○	●	●
[36]	Modeling location-aware business processes. Location-aware extension of process execution system	Declarative language for modeling business processes. Provision of additional modeling elements for location dependencies in DPIL. Description of how elements could be executed by business process execution engine	Concept and prototype of declarative location-aware process execution engine. Location-dependent task assignment and process progression	Only declarative modeling, no implementation, worklist reordering	B	⊙	○	●	○	○
[43]	Identify and conceptualize location-dependencies in process modeling	Pattern-based approach to identify location-dependency in process models	Location dependent task symbol, location dependent condition	Only graphical, influence control flow with location data	B	○	○	●	○	○
[37]	BPMN event extension for IoT	Create new BPMN events with an extended event definition (includes location event)	Start, intermediate, and end event and extended event definition for location, fault tolerance, and quality of information	Retrospective event processing based on location (and other IoT) data	B	○	●	○	○	○
[44]	Constrain control flow and process behavior based on location	Petri net modeling extension, mapping to colored Petri nets. Usage of location constraints using geospatial relationships	Modeling technique can specify how location impacts the basic control flow	Only Location constraints	P	●	○	●	○	○

Table 1 continued

	Goal	Approach/procedure	Outcome	Drawback	Notation	Executable?	Graphical	Conceptual	Architectural	New functionality
[6]	Extend BPMN to enable the modeling of mobile context-aware business processes	Sensor modeling for the dependencies of context information. Context marker, context event, and context expression language by extending the BPMN meta-model	Context4BPMN extension with context-aware BPMN elements and a modeling guideline	Focus on general modeling general context events, no implementation	B	○	●	○	○	○
[32]	Integration of an executable location constraint annotation into business process modeling languages	Integration of an executable location annotation in Petri nets and BPMN. Petri nets: Introduction of new constructs. BPMN: Usage of artifacts and associations	Extension of graphical and formal representation of Petri nets and BPMN by location constraints	Only location constraints during modeling	B/P	⊙	●	●	○	○
[7]	Include location in process modeling and process mining	Geospatial information in event logs. Geo-enabled process modeling: Nodes represent geographic locations, and transitions represent the activities. Process model is visualized on a map	Location aspects of business processes are treated as "first-class" modeling entities. Activity flow is secondary	Location data for process mining, process model from enhanced event logs	–	⊙	○	●	○	●

Notation: **B**: BPMN, **P**: (colored) Petri nets, **–**: other/not specified; Categories: ○ : only conceptual, ⊙ : description of implementation, ● : actual implementation

tion constraints, including two types of location that can be referenced: cycle locations and polygon locations. Additionally, there is a reference area within which all locations must be included. The authors differentiate between location classes and location instances, where location classes are used to conceptually group related location instances. The authors introduce several non-BPMN-conform visual elements, such as double-edged parallelograms for location classes or hexagons with equal signs for positive location constraints. Although the concepts were new in this regard, there is no need for specific non-BPMN elements, as all concepts can be expressed by combining existing and more basic patterns.

On the other hand, [33] also differentiates between direct constraints (defined at design time) and indirect constraints, which are derived during runtime and further divided into internal and external LCs based on the source of the location data (location of another activity vs. external system or manually entered). The BPMN meta-model must also be extended to fully incorporate the logical structure of the defined LCs into BPMN. To achieve this, [33] integrated the classes of the corresponding graphical symbols into the BPMN meta-model by inheriting from existing classes.

Despite the location constraints (LCs) described in [33], the authors do not address the issue of ensuring that activities are performed in the intended locations. Moreover, the focus on only LCs leaves behind a significant part of the possible uses of location data in BPM. [32] also deals with LCs in BPMN. BPMN artifacts are used to model the LCs, allowing the modelers to provide additional information about the process that is not directly related to the sequence flows or the message flows of the process [18]. As a result, the artifacts do not alter the behavior of the process. [32] extends the class diagram of the BPMN meta-model and the corresponding XML schema to include LCs in BPMN. LCs have the same constructs and functionality as introduced in [33], with the representation in the model being the main difference.

In contrast to [32, 33] also extends Petri nets with LCs.[4] For this purpose, [32] introduces new constructs to Petri nets: the constructs needed to represent LCs and a new connection type to link the new constructs with each other and with the existing transitions. The functionality and graphical representation of the LCs in Petri nets are the same as in BPMN. The authors also provide new sets and formulas for the formal representation of the extension of Petri nets.

[44] take a step further by not only restricting where an activity can be performed, as in [33] and [32] but also influencing the control flow of the process-based on location information. To achieve this, they introduced location-dependent transitions and geospatial relationships in colored Petri nets in addition to LCs. Location-dependent transitions are assigned a location after execution. Location constraints are expressions used to limit the locations that can be bound to a location-dependent transition. The expression evaluates to a Boolean result and involves exactly one output transition—the constrained transition. Note that a transition is only enabled if a token exists in all input places and all LCs are satisfied. The LCs can be viewed as location-dependent guard functions within Petri nets.

To make location-aware workflow nets (LAWF-nets) executable, [44] propose a mapping to Colored Petri Nets (CPNs), which can be executed using CPN tools. The location-based expressions used in location constraints are implemented by an extension integrating a geographic information system (GIS). Location constraints imposed by the GIS constrain the control flow. The execution of location-dependent transitions, in turn, triggers changes in the GIS.

So far, all the concepts presented are based on extending imperative process modeling languages such as BPMN and Petri nets. [36] take a different approach and focus on extending a declarative process modeling language. Declarative process modeling is particularly well-suited for scenarios with many unforeseen paths because it describes the desired result rather than the exact path to achieve it. The specific path to the result is derived from constraints that limit the activation of the following possible steps. This paper uses the declarative process intermediate language (DPIL) to model the process. In order to make DPIL location-aware, its meta-model must be extended to include location entities, such as resources and locational constraints. Two distance patterns are introduced to formulate locational constraints: *(i)* The extreme distance pattern, that determines the nearest or furthest entity from a given location, and *(ii)* the exact distance pattern, that determines the distance between two locations of entities, enabling the control of activity execution based on location information.

[40] also describes a declarative approach, where the constraints that connect and restrict activities are called business rules. The paper focuses explicitly on coordination and location laws, which are specific business rules related to coordination and location aspects. Business rules are specified in the event-condition-action form, whereas a coordination law is a business rule that regulates and combines services or activities. A location law models location-dependent aspects of business rules. Therefore, the positions of all location-dependent entities are recorded, meaning that location aspects are externalized from the source code into location laws, simplifying code adjustments and maintenance while also adding complexity by adding another part to the system.

[4] Even though BPMN is the de facto standard for modeling business processes, we include different approaches, as the underlying concepts might also apply

Architecture of Location-Aware BPMS

To effectively and efficiently use location data during the execution of business processes, an intelligent combination of different components in the form of a system architecture is required. The following section presents an exemplary system architecture, as well as an abstract system architecture for the integration of location in BPMS. [41] presents an architecture for location-aware process optimization within a factory of things. The authors focus on providing location information to the location-aware client application, which is not further detailed.

The location preparation is divided into a provision layer, which contains the various sources and technologies for location information; a mapping layer that is responsible for converting the location data into a consistent data format; and a Distribution Layer with a context broker to allocate the location data to its consumers. The interpretation layer consists of a server with a knowledge base, which can be queried via an Application Programming Interface (API) to separate the interpretation of location raw data from the location-aware client application. The client application is located in the application layer and could be, for example, a location-aware BPMS. This architecture provides a centralized location provision and interpretation infrastructure, from which location-aware client applications can benefit from reusability and separation of concerns.

In [39], the Workflow enHAncements for Mobility (WHAM) system is introduced. The system focuses on dynamic activity assignments in mobile environments. For this purpose, a two-level resource management approach is used. At the global level, dynamic assignment of activities is based on resource location and types, activity priority, and load balancing. Additional resource attributes filter and prioritize the assigned location-dependent activities at the local level. According to the understanding of [39], these activities are tied to a specific location. To ensure dynamic and location-dependent assignment of activities, a location status for mobile resources has to be maintained. Furthermore, the late assignment of mobile resources to activities is crucial, as they frequently undergo spatial relocation (i.e., they move) over time.

To implement dynamic and late assignment of mobile resources to activities within workflows, [39] employs a workflow server in conjunction with a mobile resource management component on the server side. The resource's location is determined on the client side and stored within their respective resource profiles. When the client establishes a connection with the server, the worklist handler conveys the current location to the server, generating an appropriate worklist tailored to the client's location. The workflow proxy and the work item data cache are vital in ensuring that activities can be executed, even when mobile participants are temporarily disconnected from the workflow server's network. Furthermore, a map view, integrated into the worklist handler, displays the locations of activities within a participant's worklist, assisting mobile participants in planning their work.

Use of Location for New Functionality

Finally, using location in BPM also fosters new applications, such as including geospatial information in event logs for process mining [7]. Consequently, location aspects become the primary modeling entities, while activity flow becomes secondary. Moreover, while the presented approach differs from earlier ones, at least from a lifecycle standpoint (modeling and execution vs. process monitoring), representing geographic locations and using event logs of transitions of entities between them also incorporates location information into the process analysis.

Detached from any specific modeling language, [45] proposes that location enhances human task assignment in business processes. They suggest using location and user availability status to efficiently assign tasks and prevent tasks from becoming idle.

Finally, another use of location in *augmented process modeling*, similar to the approaches presented above, is visualizing processes and tasks in actual spaces. [34], for example, focuses on visual support for work assignments in mobile environments. They achieve this by visualizing the location of tasks on a map while also employing distance metrics to suggest the next task based on various factors. This approach lifts the use of location away from the process model itself and implies implementing a complementary task list.

On the other hand, [46] also uses the location as part of contextual process information to use augmented reality to display process specifics at the actual location of the task (e.g., live sensor data at the specific machine or a task list next to a workspace with tasks specific to the workspace instead of the actor). These approaches differ fundamentally from the patterns presented in the following sections. While they use the location of a process entity, they are not directly tied to the process model as a way of visualizing and providing the needed logic for process execution.

2.3 Research methodology

The following sections follow the introductory course of the well-established *design science research methodology* (DSRM), which has the fundamental goal of creating valuable artifacts within the information systems discipline [47]. Methodologically, the current procedure model of [48] and [49] divides the research process into six iterative phases: (i) *explicate problem*, (ii) *define requirements*, (iii) *design and develop artifact*, (iv) *demonstrate artifact*, and finally, (vi) *evaluate artifact*. However, the problem and the requirements for a solution have already been outlined in Sects. 1 and 2, Sects. 3 to 5 explain the basics of generation and processing of location information and different patterns of the use of spatial information with BPM and an attached BPMN extension, and Sects. 6 to 8 form the last step by providing a prototypical implementation for the previously presented patterns, combining the demonstration and evaluation in terms of currently running real-world application examples for the patterns.

We have also met the requirements of DSR as an iterative process [49]. In addition to conducting rapid system prototyping, we actively engaged with our industry partners by presenting intermediate results. Their valuable input and expertise were fundamental in refining our objectives and enhancing the system architecture's design and functionality.

3 Location information generation and processing

The context of a process describes the conditions and circumstances involving the entities [4]. This information can be used during the execution of a process to enhance and support its realization. One common aspect of context frequently used is the location of entities, which includes both spatial and temporal information to be used during process execution.

To employ an entity's location as contextual information during the execution of a business process, we first need to acquire current location data. Fortunately, the Internet of Things (IoT) and other advancements in networking and technology have led to various techniques for object localization.

These physical locations can be categorized into three distinct types [25]. For real-world BPM applications focusing on distributed mobile processes, we can primarily focus on the first two[5]:

- *Descriptive Location*: These locations are characterized by descriptive names or numbers and are typically associated with well-defined natural or human-made geographical objects.
- *Spatial Location*: These locations are explicitly represented within a coordinate system, commonly either two-dimensional (latitude and longitude) or three-dimensional (latitude, longitude, and altitude).

Upon acquiring location data, processing it to extract meaningful insights from the various data points becomes necessary. Depending on the type of location data, an initial transformation is required to enable mathematical calculations. Geocoding serves as a tool to facilitate the conversion between spatial and descriptive representations. This process relies on lookup tables that contain mappings between addresses and numerical representations. It is important to note that the transformation from spatial to descriptive representation inherently involves a loss of precision, especially in large company premises with multiple buildings sharing the same official address but having significantly different spatial coordinates. Depending on the specific needs, there are also descriptive location descriptions available that do not rely on regular addresses and can overcome shortcomings for large premises or unserviced land or forests, such as *What3Words*[6] or *FixPhrase*.[7] Both of these services rely on the same fundamental concept: the division of the surface of the earth into fixed small areas that are uniquely identifiable (3×3 meters for What3Words or approximately 10×20 meters for FixPhrase) and the subsequent provision of a service to geocode the addresses of these alternative address systems back to latitude and longitude. This transformation enables comparability between values, using mathematical formulas and calculating distances, making it more suitable for computational purposes. On the contrary, descriptive locations are human-readable, straightforward to remember, and comprehensible.

This duality of representation should be considered when utilizing location data with human actors. Depending on the specific requirements, any application must be capable of supporting both descriptive and spatial representations to accommodate the data consumer's needs.

Even after transformation, a single numerical data point alone is insufficient for direct utilization within a business process. The meaningful analysis typically requires at least two data points containing a combination of two distinct entities, locations, or timestamps. Combining these data points allows us to derive meaningful insights and understand the context of the entity's actions. The subsequent section will formally describe various location operators, assuming the availability of two data points for analysis.

3.1 Location data collection

There exists a multitude of various methods for technologically locating objects, extensively described in [25, 50–52]. Three distinct methods for localization can be distinguished:

1. *Presence*: This is the simplest and least accurate method to determine whether an object is present.

[5] The third category pertains to a location within a communication network (*network location*), which is less relevant for real-world appliances

[6] https://what3words.com/

[7] https://fixphrase.com/

2. *Positioning*: This method aims to identify and locate objects accurately.
3. *Proximity*: This method falls between the other two, aiming to determine presence and location, which depend on each other.

In Table 2, a non-exhaustive compilation of existing location technologies is provided, along with associated techniques and parameters, including range, accuracy, cost, and energy consumption. Notably, many currently available technologies and techniques exhibit drawbacks, encompassing challenges related to cost and energy consumption, as well as complexities and scalability issues.

These techniques are accompanied by several additional developments and specificities, meaning this list should be regarded as an overview of possibilities. Further comprehensive descriptions and explanations can be found in the above sources and other sources for those seeking more detailed insights into the inner workings of different technological approaches and techniques.

Depending on the technology chosen, subsequent steps must be taken to access the collected location data in a machine-readable format, which can then be processed in the next step.

3.2 Location data processing

After collecting and storing the locations of relevant entities, our focus now shifts to determining how to utilize these different locations effectively. The first step in CEP would be to process the incoming data and generate events from them, which can be used in a different system. However, because location information is needed, other challenges of IoT data, like a sub-second frequency or overwhelming data, are less relevant when dealing with location data. Depending on the used source for the data, we can already filter, thereby only updating the location if it changes. For our application and the use of location, we can directly use currently available data without preprocessing it.

Drawing from the patterns presented in [13] and related literature, as exemplified by [53], which will be elaborated upon in Sect. 4, we have identified various operators essential for incorporating location in BPM. We introduce spatial and temporal operators to form a foundation for these patterns. The spatial operators bear a resemblance to those introduced in [36], which contains concepts for two explicit distance patterns, an *extreme distance pattern* (where one entity is either the nearest or farthest from another entity) and an *exact distance pattern* (which binds the current distance to a variable).

For the following patterns, we need different operators that we can use for each of them. They can be mainly split into two categories: *temporal-independent spatial operators*, where we select one entity from a list of entities based on their

Table 2 Comparison of different techniques for localization

Technology	Technique	Positioning	Application area	Range	Accuracy	Cost			Energy-usage	Scalability	Complexity
						Base-station	Terminal/tag	Operation			
Barcode	PS	R	I	10 m	1 cm – 10 m	⦿	⊙	⊙	⊙	⦿	⊙
Infrared	PS	R	I	5 m	1 cm – 5 m	⦿	⊙	⊙	⊙	⦿	⊙
RFID	PS	R	I	0,5 – 10 m	1 – 5 m	◉	⊙	⊙	⊙	⦿	⊙
NFC	PS	R	I	10 cm	1 – 5 m	⦿	⊙	⊙	⊙	⦿	⊙
WiFi	PM	R	I	on the network (< 100 m)	1 m	⦿	◉	⦿	⦿	◉	⊙
BLE	PS	R	I	10 m	0,5 – 2 m	⦿	⦿	⦿	⊙	◉	⊙
UWB	A	R	I	50 m	0,5 m	⦿	◉	◉	◉	⊙	⊙
LPWAN	PM	R	I/O	on the network (< 40 km)	1 m	●	⊙	⦿	⦿	◉	◉
Radar	A	R	O	1000 km	15 m	●	●	●	●	⦿	◉
cellular	PS	A	O	worldwide	50 m – 1 km	–	⦿	⦿	⦿	⦿	◉
sensor-based	PM	A	I/O	on the network	10 cm – 1 m	◉	⦿	⊙	⊙	⦿	◉
visually	PM	A	I/O	–	1 cm – 1 m	⦿	◉	⊙	⊙	⊙	◉
GPS/GSM	PS	A	I/O	worldwide	6 – 10 m	–	◉	⊙	●	⊙	◉

Rating: ⊙: low, ⦿: medium, ◉: high, ●: very high
Techniques: **A**: Angulation, **PS**: Proximity Sensing, **PM**: Pattern Matching
Positioning: **R**: Relative, **A**: Absolute
Application area: **I**: Indoor, **O**: Outdoor

current location, and *temporal-dependent spatial operators*, where we examine one single entity and check for noteworthy events and its movement over time.

Consider the following sets: $\mathcal{E}$ representing the positions of all possible entities involved in a process, divided into static ($\mathcal{S}$) and dynamic positions ($\mathcal{D}$) with $\mathcal{E} = \mathcal{S} \cup \mathcal{D}$ and the set $\mathcal{A} \subset \mathcal{D}$ representing the positions of actors and let $f_d(x, y)$ be any distance function that returns a distance[8] between locations x and y.

Temporal-independent Spatial Operators

- To find the *minimum distance* between the location of any one entity and the location of another entity $e_1, e_2 \in \mathcal{E}$ with $e_1 \neq e2$, you can use the following expression:

$$\min f_d(e_1, e_2) \tag{1}$$

- Similarly, to find the *maximum (longest) distance* between an entity's location and all other entities from $e_1, e_2 \in \mathcal{E}$ with $e_1 \neq e2$, you can use:

$$\max f_d(e_1, e_2) \tag{2}$$

These expressions represent the minimum and maximum distances, denoted as $d_{\min}$ and $d_{\max}$, respectively, between all current locations entities $\mathcal{E}$ and the location of exactly one entity $e \in \mathcal{E}$.

Temporal-dependent Spatial Operators The calculation of whether an entity is inside or outside of a geofence is independent of the actual distance function here (this is mainly the case for geofencing, where we draw a circle or polygon around a location and want to check if an entity moves inside or leaves the area). We can either use the distance around a specific location as a threshold, leading to $i(x, y)$ being the result of the location update (making the geofence a circle), or more proficiently use the ray-casting algorithm that can be used for arbitrary polygon geofences, and returns whether or not a point is located within a polygon.

$$i(x, y) := f_d(x, y) \text{ op } d_{\min}$$
$$i(x, y) := \text{boolean interrelation between } x \text{ and } y$$

where

$$\text{op} \in \{<, \leq, =, \geq, >\}$$

The *location update* describes the fact that the location of an dynamic entity $d \in \mathcal{D}$ changed at least/exactly/at most $d_{\min}$ between $t-1$ and t.

$$f_d(d_{t-1}, d_t) \text{ op } d_{\min} \tag{3}$$

To accurately detect an *entity entering a geofenced area*, both of the following conditions must be satisfied:

- The first condition ($i(d_{t-1}, e_{t-1}) =$ false, with op being $<$) ensures that the entity a was initially located outside the geofenced area just before the time interval t. This condition is necessary to prevent false positives where the entity is already inside the geofence at the beginning of the interval.[9]
- The second condition ($i(d_t, e_t) =$ true) confirms that the event has indeed entered the geofenced area at time t. This condition is essential to validate that the event has crossed the geofence boundary during the specified interval.
  Thus, we can define the enter geofence operator as follows:

$$i(d_{t-1}, e_{t-1}) = \text{false} \wedge i(d_t, e_t) = \text{true} \tag{4}$$

  Similarly, to detect an *entity leaving a certain area*, we reverse the truth values. a is inside the geofence at time $t-1$ and outside of it at t. Meaning:

$$i(d_{t-1}, e_{t-1}) = \text{true} \wedge i(d_t, e_t) = \text{false} \tag{5}$$

Using those operators, we can concisely express different operations using the location of entities. For example, the specific elements referenced in [53] can also be expressed using these operators. The operator *position achieved* is the inequality $f_d(x, y) \leq \varepsilon$ (with ε being a small positive number, optimally zero), *position update* can be expressed using the operator above, and *the conditional position* is entering or leaving a geofence each for $a \in \mathcal{A}$ and $s \in \mathcal{S}$.

4 Patterns for location-aware business process execution

The following sections present different possibilities for using spatial information within executed business processes and how it can influence the execution of mobile distributed processes. These patterns generally describe the use of location in business processes. To facilitate comprehension and to make the abstract patterns more accessible, we align the patterns directly along commonly known elements and

[8] Different distance metrics can be calculated or queried (e.g., as-the-crow-flies, driving distance)

[9] While it may not be practical to define a geofence around a moving entity (d), we do not impose restrictions on the operator solely using static or non-changing locations (s)

Table 3 Patterns for the use of location-based information in executed processes

Conceptual use	Task	Involved BPMN concept			
		Gateway	Boundary event	Single-instance	Multi-instance
Automatic start of process instances	○	○	○	●	○
Decision making through location-based data	○	●	○	●	○
Location-based task allocation	●	○	○	●	●
Location-based monitoring of tasks	●	○	●	●	●
Location-based event dispatching	●	○	●	●	○
Location-based automatic completion of tasks	●	○	●	●	○
Location-based automation of tasks	●	○	○	●	●
Location-based worklist reordering	●	○	○	●	●
Location-based constraining of execution	●	○	○	●	●

paradigms of BPMN [18] (for a short introduction on the basics of BPMN, see Section A). Nevertheless, the patterns are agnostic to the process modeling standard or paradigm and even more so to the implementation of the execution engine.

Using a BPMS, the task's execution can be automated and supported by IT. At the same time, a business process model describes the schema of a process; typically, different instances of any process run concurrently. While most of the focus of research and implementation support is on executing and optimizing single instances ("the process model as it is"), looking at multiple concurrently running processes leads to a completely different set of restrictions, and the interplay leads to other possible synergies that are not depictable in standard process models.

Table 3 shows a non-exhaustive list of the identified conceptual uses for location-aware business processes together with a classification of BPMN patterns that we will explain in more detail in the following sections. The following table also differentiates between single-instance and multi-instance, the first including patterns that can be used on a single execution instance of a process. In contrast, the latter implies the existence of at least two process instances (multi-instance), as described above. Utilizing or integrating location data within a task or facilitating its execution constitutes another use of such data. However, it is essential to note that location is just one component within a more extensive collection of necessary or utilized data points compared to the other patterns, which will be mentioned in Sect. 4.10.

4.1 Location-based automatic start of process instances

Location data can be used to start process instances automatically, e.g., starting an externally modeled process for a specific task when an actor arrives at a specific location. As explained above, primary context information, i.e., the single location of an entity alone, is insufficient to start a new process instance. Only when the location of at least one other entity or a different location for the same entity is available can metrics be calculated to initiate a process instance, e.g., the event distance location with the place of fulfillment for the job being the event, meaning, we need at least two different data points to use our operators. Fundamental to that is the availability of location data outside the bounds of the process instance for all involved entities.

Depending on the calculation of the used metric, logic is required to be included within an event start (in BPMN: a conditional start event), meaning that the calculation has to happen "outside" of the process instance's lifecycle, implying that the start of the process instance is not bound to the BPMS but has to be triggered externally. The triggering event of a process could also be modeled as part of a message start event, relying on the viewpoint and the modeling choice, thus losing semantic information during the modeling. Coming back to the presented context levels of [25]: spatial information of the involved entities are primary data, as they are gained from sensors (or more specifically mobile IoT devices) and cannot be used by themselves, whereas the distance between two entities is derived, thus secondary context, and can be used to start a process instance.

4.2 Decision making through location-based data

Decisions are a fundamental aspect of any modeling language and enable the structuring of the process by influencing the process flow [1]. Decisions can be implemented in all parts of the system architecture depending on the frequency and availability of required location data. These decisions range from using an expression that gets evaluated directly within the BPMS to employing lifecycle hooks and execution listeners attached to the BPMS. Alternatively, decisions can be deferred to the middleware or processed as a stream using CEP, thus closing the abstraction gap between IoT data and human-executable processes while incorporating all three spatial context classes.

As the most basic concept, decision-making based on location information is the most general application. Every conceivable data point can be included in a business process decision. Depending on the level of sophistication, decisions can be made at a basic level using the operators above for simple decisions or incorporated into larger decision tables where the location of an involved entity is just a tiny part.

4.3 Location-based task allocation/assignment

Another noteworthy pattern involves allocating or assigning one or multiple actors to an activity or subprocess. While the terms *Allocation* and *Assignment* may initially appear synonymous, their usage has been distinguished in the previous literature. Various authors have coined their definitions for these terms, sometimes exhibiting overlaps or contradictions. For this discussion, we will employ the term "allocation" to signify assigning a specific task to a single actor during the process execution. Conversely, "assignment" encompasses a broader scope, encompassing candidate users and groups during process modeling and programmatically filtering and modifying them during process execution.

It is important to note that allocation establishes a direct connection between a single actor and a given activity or subprocess. At the same time, an assignment can result in none, one, or multiple candidates being associated, independent of the current process status. However, both activities address the interaction between actors and activities and will be jointly explained.

A task allocation to a process participant specifies the responsibility for a single task or a group of tasks to a concrete user. During process modeling, tasks can get assigned to a particular user group whose members are candidates for execution, meaning the allocation happens deferred during the process execution [1, 54]. Although earlier approaches considered the use of location data for process models (cf. Section 2.2), the research gap of allocating concrete users to tasks based on location data has been neglected.

The realization of a location-based task allocation is primarily a task of a smart cross-system implementation. Let us explain this directly using a small application scenario: Imagine a crafts business where exactly one actor is working at one customer location for one specific process instance; this subprocess of concrete manual work starts with the allocation of the current process instance to exactly one worker. The start of the tasks triggers either the calculation or querying of distances and the allocation itself. Because the BPMS itself does not store real-time data and because of the separation of the backend and the data-handling middleware, the BPMS has to either actively get the information or delegate the task to the middleware. Depending on the actual need, both possibilities are feasible. Even more, we can either try to get the current location of all possible entities by directly querying the sensor and using the most current stored data as a fallback or solely rely on the most current known location. After calculation, a list of distances with unique identifiers for the corresponding actors is returned, which needs to be sorted and then used for the allocation. All possible logical expressions/operators can be applied to the data based on the locations and the application area.

One significant issue must be addressed is the allocation or assignment span, also known as the allocation lifecycle. Depending on the process and the modelers' needs, this pattern has to include a possibility of involving more than just one task visually and in the implementation. For this reason, both a single task and a subprocess, including multiple tasks, can be used for allocation and assignment. After the task or subprocess is finished, the used actor is available again, and all relations are removed.

Ultimately, the allocation gets sent to the BPMS, including the current distance to the target location, which can then be used further to support the actors and increase the transparency of the allocation process. Depending on the actual implementation and the BPMS choice, we can either explicitly model the allocation/assignment as a script task or append it to the lifecycle of a complete subprocess (again, for BPMN, cf. [12]). As in decision model and notation (DMN), where the goal is a decision between multiple process flows [55], the allocation of a single actor to a task based on location data can be understood as a Boolean decision for one and against all other actors, whereas the assignment filters the candidate users for a task or list of tasks.

4.4 Location-based event dispatching

Following the classification of spatial context data from [26], we can divide all events that occur during the execution of processes into three classes, which then can be used during process execution to react to them and trigger additional functionality in the process. Based on the operators above, we can define different events that can occur during the execution of a task based on the locations of the involved entities. One crucial possible application is the use for notifying customers or actors based on the current location of an entity (e.g., a worker driving to a customer, where a new task is triggered when the worker enters a certain distance around the customer or automatically dispatching the refilling of resources when the supplier dropped them off at the facility).

Within BPMN, the supervision of tasks is conceptualized with the use of boundary events so that different reactions to events occurring during the execution of a single task can be modeled [1]. These events can split the process flow while executing a single task or subprocess. BPMN defines conditional non-interrupting boundary events to trigger a separate flow based on a condition happening while one task is executed [18], which spatial data can enrich.

4.5 Monitoring process execution

The inclusion of spatial information also eases the monitoring of process execution, improving transparency by displaying all involved entities on a map and visualizing the physical environment where tasks are going to be performed, including the temporal properties of the process instance [34]. Furthermore, multiple instances running in parallel, including current and historical data, could be used to calculate the median execution time of a single task or a group of tasks or the maximum distance traveled to a customer among the most recent *n* jobs. In addition, the quality of task execution can be enhanced by providing the current location data of all participating entities to mobile task workers. The display of all involved entities on a map offers a visualization of the physical environment where tasks are going to be performed that includes temporal properties of the process instance [34], represented by the current position of an actor with a possible trail to show the path taken.

Conceptually, either the BPMS or the middleware has to keep calculating the distance of the allocated actor to the customer's location in the background and either push the result as a process variable to the BPMS or periodically evaluate the given expression. Depending on the company's needs, a fitting granularity of updates and data accuracy has to be chosen.

All in all, because spatial data are nothing more than contextual information, its use for monitoring the execution of processes can lead to improved transparency and show possible areas of improvement for the following improvement of the processes (business process improvement (BPI)).

4.6 Location-based automatic completion of tasks

The process flow defines the fundamental sequence of all parts within a process model, where each element (excluding the start and end events) has at least one predecessor and one successor [1]. Upon the completion of an element, all directly connected elements in the model are triggered. This process can also be automated using location-based data. The update frequency and the accuracy of the location information influence the precision of automatically completing the current task and proceeding to the following flows. It is important to note that user tasks are not automatically executable in a BPMS and, therefore, do not implement any logic [1]. However, BPMSs typically provide lifecycle hooks as listeners that can be used to execute logic when specific states of the task's lifecycle, such as *create*, *assignment*, or *complete*, are reached.

Compared to dispatching events using location data, we automatically finish one task in this approach, triggering the following tasks. When the first task splits the process flow, and the base task is not finished due to the non-interrupting nature of the boundary event, using the task's lifecycle allows us to complete tasks based on the location of involved entities. While implementing the latter approach requires decisions and calculations to be made outside of the BPMS, for the former approach, expressions can be evaluated within the BPMS itself.

4.7 Location-based automation of tasks

Location data can enhance the execution of tasks that have to be done by a human but can be supported by technology (user tasks in BPMN) and can be used to completely automate non-value-adding tasks like the documentation of work and support tasks for value-adding processes. Due to the simple fact that most BPMS typically save records of historic execution data, this information can be used on a multi-instance basis to calculate and verify metrics like start and finish time, as well as custom location-specific data of tasks like comments made by field workers for specific entity locations (Imagine a heating engineer storing location-specific information assigned to one customer). These comments can be used to support location-based information for future projects. Within our running example, the last task is creating an invoice that can be automated or supported using the execution times of the current process instance tasks split into time driving and working on-site at the customer, simplifying and automating the billing process while removing non-value-adding documentary work.

Beyond the benefits mentioned above, location-based automation can also improve workforce management. For instance, real-time tracking of field workers' locations allows for efficient dispatching, ensuring that the nearest available technician is assigned to a job, minimizing response times, and improving customer satisfaction (as already explained in Sect. 4.3). Moreover, the integration of location data can lead to predictive analytics. By analyzing historical data related to task execution times at specific locations, businesses can forecast future workloads more accurately, enabling proactive resource allocation and optimized scheduling (see Sect. 4.6). This predictive capability streamlines operations and aids in budget planning and resource allocation, ultimately contributing to cost savings and improved operational efficiency.

Furthermore, as mentioned earlier, utilizing location-specific comments can play a pivotal role in quality assurance and customer engagement. These comments can be analyzed to identify recurring issues or customer preferences associated with particular locations. This valuable information can guide process improvement initiatives and allow businesses to tailor their services to specific customer needs, ultimately leading to higher customer satisfaction and loyalty.

4.8 Location-based worklist reordering

Taking a step back from a single executed process instance, location-based information can also be used to reorder the work list of an actor based on location information, e.g., the actual distance of a process participant to the point of interest like a customer location, i.e., entity and possibly event distance location. In contrast to the concept of work list reordering in [56], which investigates the influence of random reordering tasks on overall temporal performance based on historical data, recommender systems can be used in combination with BPM to build suggestions for the next task to be executed by a human actor. Following this approach, any metric calculated based on an actor's location or a combination with other metrics can be used to rearrange tasks within a work list of a specific user. [39] have already touched on the topic within their local resource management, adding the work preferences of human resources and application-specific resource types and states to a decision matrix while differentiating between filtering possible tasks or prioritizing them. Within the running example, imagine multiple open jobs, i.e., tasks from different process instances, at different customers yet to be allocated to a single actor. Each available actor gets a list of those tasks on their mobile user interface sorted by distance to the corresponding jobs and can then choose their next task based on this information.

Compared to allocation and assignment, this broader perspective enables organizations to dynamically allocate tasks to actors based on proximity to customers and other strategic considerations, such as optimizing resource utilization or minimizing transportation costs. Furthermore, integrating recommender systems and BPM opens avenues for intelligent task allocation. Organizations can leverage historical event logs and machine learning algorithms to provide actors with task recommendations that align with location-based metrics and individual preferences. This personalized approach can result in improved task execution times and enhanced user satisfaction.

4.9 Location-based constrained execution

One of the most common patterns in earlier research (as seen in Table 1) involves explicitly constraining the execution of specific tasks within a business process. This constraint ensures that individual activities must or must not be performed at specific locations. These constraints can be categorized as direct or indirect, as defined by [33]. Direct constraints are known and established during the modeling phase of the process, while indirect constraints are dynamically evaluated at runtime.

In practical terms, the constraint mechanism we have discussed can be a powerful tool. For example, with the right business process management system (BPMS), you can maintain a list of all entities already participating in the process instance. This list can then be used to restrict their future involvement, essentially creating a "deny list" of actors for specific activities. This approach is an extension of the earlier pattern of assigning a task to an actor based on a deny list but with the added refinement of further reducing the candidate group not only by distance but also by considering "has already worked on" criteria.

However, it is crucial to note that while the patterns introduced earlier are independent of the actual implementation and rely solely on essential BPMN elements and functionalities, back-referencing to earlier tasks is not a standard feature of BPMN. This can lead to implementation dependencies and requires careful consideration when designing location-based constraints in processes.

4.10 Other uses of location data in business processes

As explained above, location data can also be part of a bigger collection of data points within business processes. The focus of the patterns here is on the direct use of location data to influence the flow of processes, which could also result from using location data within a task.

Within single tasks, the use of location data is possibly endless, from personalizing marketing and sales campaigns where the customer's location is used in combination with their preferences and behavior to tailor more relevant and timely promotions to improving travel and tourism services by providing recommendations or enabling location-based payments and transactions for preventing fraud and enhancing security by using the customer's location and their usual location.

Comparing those approaches, we can quickly identify a difference: While the above patterns primarily focus on executing tasks, other uses focus on modeling, analyzing, and improving the structure and logic of process models.

Consider the conceptual implementation of location-based task allocation. This approach allows the process model to dynamically assign or allocate actors to tasks based on their geographical position rather than relying on static or predetermined assignments. This dynamic allocation can significantly enhance process execution flexibility and efficiency, potentially boosting actor satisfaction and motivation. This innovative use of location data in business processes can inspire new ways of thinking about process management.

As mentioned in the related works part, augmented process modeling includes spatial information combined with processes. Compared to the above patterns, this almost always implies the need for additional tools or services. Here, for example, we can display the actual fulfillment location of tasks on a map, transforming the classic tasklist into a task map and improving the visualization of the link between multiple tasks that the same actor may execute. Another possibility could be to include virtual and augmented reality or mobile devices to provide needed information based on the current location of the actor. This information can include currently open tasks for a specific location (e.g., workspace or construction site) or specific data (e.g., all current sensor values for the machine in front of an actor or contextual information for a customer's location).

Finally, another conceptual use of location can be for personalizing marketing and sales campaigns. This approach allows the process model to incorporate location data to tailor the content and timing of the campaigns to the customer's location, preferences, and behavior. The potential benefits are significantly improved effectiveness and relevance of the process outcome and enhanced customer loyalty and retention. This use of location data in business processes offers a promising avenue for improving customer relationships.

5 Extending BPMN for location awareness

In the following section, we describe an approach for BPMN specifically as an exemplary modeling approach for the actual use of the conceptual patterns for the use of location information in business processes. For this, we introduced a BPMN extension for location-awareness in [12]. Using the guidelines in [57], we deliver the missing piece by extending the BPMN metamodel to enable a visual and understandable implementation of said patterns. The extension for location-aware BPMN (LABPMN) contains three main elements: *location data objects* (referring to the used locations), *location events* (all different standard types of event (throwing, catching, interrupting, non-interrupting) but for the explicit use with location—based on the operators described earlier), and finally an extension for *subprocesses and tasks* for dynamically assigning and allocating actors to a task based on their and the task's current location.

Using those BPMN elements, we can model all the patterns presented above to correctly represent the functionality within the process model itself. The extension of the BPMN metamodel to explicitly include location awareness and specific elements itself needs a complementary implementation based on a BPMS and a possibility to model said elements.

Providing the extension for BPM includes extending the BPMN metamodel conceptually and graphically while extending the BPMS core functionality to keep the process model executable. The Internet of Things is a pivotal enabler for new approaches to using contextual information in business process modeling and execution while relying on the de facto standard BPMN, which enables the correct depiction, transparent execution, and use of data.

6 Visibility and interaction of location data

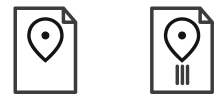
Between identifying and presenting the different patterns for the use of location data for BPM and a short introduction of the implementation, we integrate the presented patterns into the well-established data patterns defined in [14]. This integration helps explain how data can be used and exchanged, which is a fundamental step in bridging the gap between conceptual solutions and prototypical rational implementations.

Within the context of BPM, data points from various sources can be defined, stored, and utilized at different levels. Values can be made accessible within a defined part of the process or at any point for every element of the process. The precise specification of the scope and visibility of the data is crucial for addressing existing problems or adhering to best practices, such as managing limited resources, managing traffic, ensuring security, and addressing privacy aspects [58] related to IoT devices and their use while optimizing data handling within the BPMS.

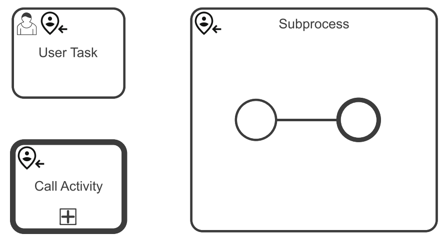
The different classes described in Table 4 show the visibility of the data within the BPMS, ranging from the most specific (task data) to the most general use of the data (environment data). Minimizing network traffic and efficiently managing data requires consideration for the implementation. For instance, when creating an invoice for a customer, access to the execution time of the current process instance's tasks is essential. However, these data will no longer be needed once the invoice is created. On the other hand, customer information, such as their location, should be implemented as folder data since a single customer could require multiple repairs and thus be associated with multiple process instances; not saving these data for future tasks is not practicable in the long run. However, data utilization and handling often depend on specific implementations, system architectures, and the current process, making it challenging to provide further conceptual descriptions beyond their given definitions in [54].

Furthermore, [54] defines various patterns for external data interaction, primarily distinguishing between push and pull communication. Push communication involves variables within the processing framework (e.g., IoT middleware) that are saved and can be processed into events. These events are then pushed into the BPMS, where they must be referenced within tasks. In this scenario, the processing framework sends data to a specific task, case, or workflow without explicitly requesting it from the BPMS. To illustrate, when a task becomes active, the middleware transmits data at regular intervals or when the value changes to the BPMS. This approach can reduce processing time and latency, as the data

Fig. 3 Graphical representation of all elements added to the BPMN meta model



(a) Graphical extension for Location-Aware Data Objects



(b) Graphical extension for Location-Aware Assignment/Allocation of Tasks

(c) Graphical extension for the different types of Location-Aware events

are immediately available for processing. However, it may result in the transmission of unnecessary data.

In contrast, *pull communication* operates vice versa; the BPMS actively requests data from the middleware when needed, such as for making decisions or using it within a service task. This approach can result in longer processing times if the middleware waits for the BPMS to request data before transmitting it. The trade-off between push and pull communication must be carefully considered for each scenario to minimize network traffic while ensuring the necessary responsiveness, especially considering the limited resources in IoT devices. It is also essential to consider the relevance of the data: not every sensor value or processed event is required for every process component.

The visibility types defined above and their use for the patterns vary greatly depending on the choices. While some exemplary location-dependent data points seem easy to categorize, many can be used differently depending on a company's needs, the available data, and the chosen method of implementation (both the implementation itself and the deployed and used BPMS).

7 Implementation

After presenting the patterns for using location and a brief overview of its potential applications with BPMN, the objective is to establish a fully executable and system-driven approach based on location data using existing standard process management technology. This section outlines the system architecture based on Business Process Management (BPM) and the frameworks involved in a prototype implementation.

In order to make use of the different patterns, we developed a multi-level system architecture for integrating contextual location information in BPM (Fig. 4). With this approach, we can enhance process execution by utilizing location data from mobile devices used by individuals involved in the process. The implementation also provides real-time information on current tasks that need to be performed and includes a back-office application that offers contextual information about actors, customers, and other process-related resources. This tool is an initial approach to dynamically adapt mobile applications for individuals working at different locations solely based on the modeled and executed business process.

The backend employs a conventional layered model-view-controller (MVC) architecture with Spring Boot[10] as its technological foundation. It offers a REST interface for the mobile application and HTML web pages for human users. These interfaces support common use cases, including storing and retrieving tools, locations, projects, and user tasks. For data storage, we use a SQL database (PostgreSQL[11]) with a geographic data extension (PostGIS[12]). To accommodate a wide range of mobile devices (Android and iOS, with the

[10] https://spring.io/projects/spring-boot

[11] https://www.postgresql.org/

[12] https://postgis.net/

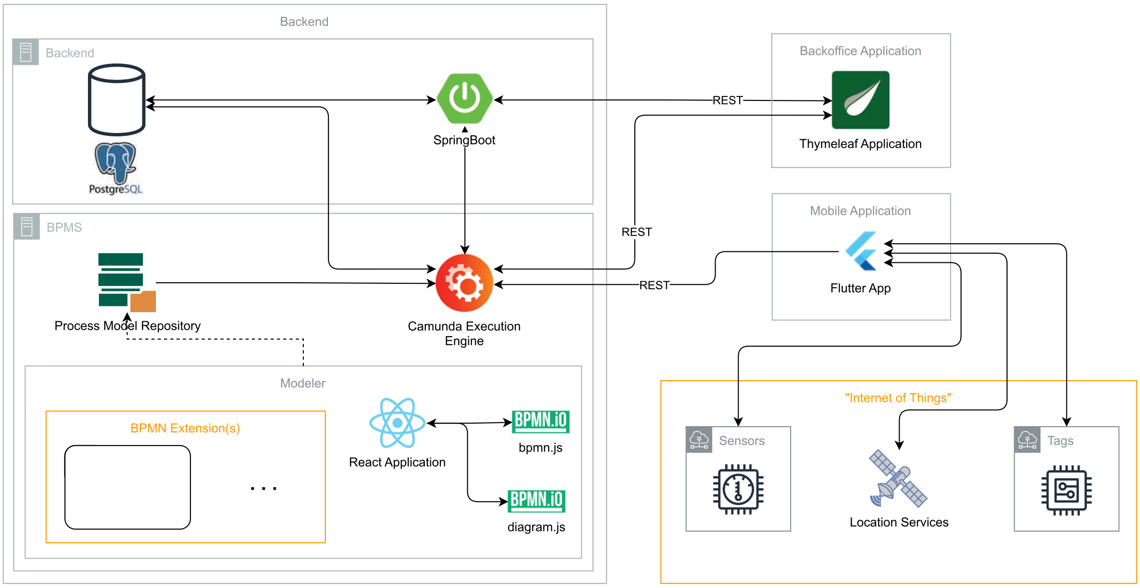


Fig. 4 Software architecture for the use of IoT data in BPM

option of deploying as a web application) and enable access to native device features such as Bluetooth communication and location services, we have opted for Flutter as the mobile development framework.

The first application is based on the BPMN extension published in [12]. The implementation of the modeling aspect of the extension is built upon the open-source implementation *bpmn.js*,[13] which seamlessly integrates with BPMS. Camunda is responsible for executing the process model and providing the necessary logic.

The various introduced elements can be selected either directly from the context menu or after assigning a specific location-aware event type or distance type to the element within the properties panel (see Appendix Section B, Page 41).

To enhance efficiency and effectiveness in process execution, we also provide a mobile application for actors responsible for carrying out the tasks of the business process (Fig. 5). This application includes functionalities such as accessing current tasks (both assigned and unassigned), initiating and completing tasks, accessing attached forms, and working with them. It also supports initiating process instances and provides easy access to the actor's current location through the mobile device's geolocation services. Additionally, it can connect to other IoT devices, such as reading sensor values or accessing Bluetooth Low Energy (BLE) tags. Next to mobile devices like smartphones, tablets, and smartwatches, because of the standard implementation approach, all different kinds of devices could be added as

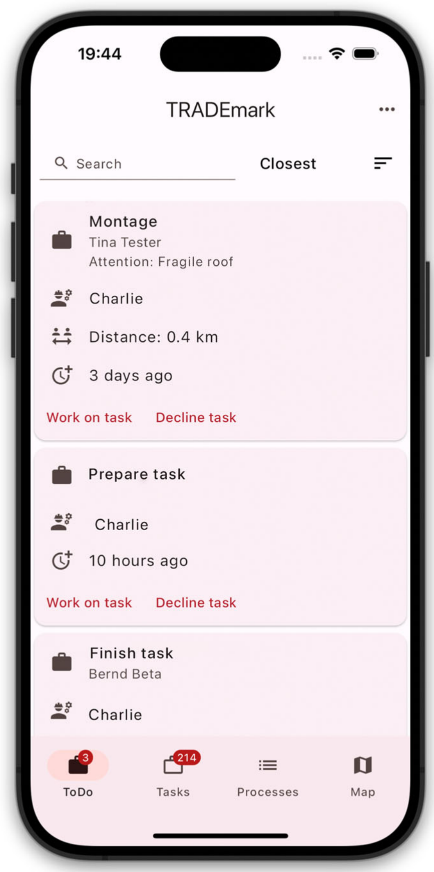


Fig. 5 Mobile application showing currently available and assigned tasks sorted by current distance, including customer-specific context information

[13] https://bpmn.io/toolkit/bpmn-js/

Table 4 Data patterns defined by [14] and exemplary data depending on location

Visibility type	Definition of data visibility	Exemplary location dependent data
Task data	Within the context of individual execution instances of specific tasks	Time spent on-site for equipment maintenance at various customer locations
Block data	Within all components of corresponding subprocess	Geographic coordinates of all tasks within a field service subprocess
Scope data	By a subset of tasks in a case	–
Multiple instance data	Tasks that are executed multiple times within a single case with instance-specific data	Distance traveled for each delivery task within a logistics case
Case data	Specific to process instance (case)	Location-specific details for project milestones in a construction project
Folder data	By multiple cases selectively, accessible to all bound cases	Regional distribution centers accessible to multiple logistics cases
Workflow data	By all components in every case of the process ("process context")	Availability of on-site staff for emergency response across cases
Environment data	Data elements existing in the external operating environment	Real-time traffic conditions affecting delivery routes

data sources (e.g., single-board computers (SBCs) with SIM cards or network connection).

The backend comprises the BPMS (in this case, Camunda[14]), a Spring Boot application that integrates and manages all components, and the mandatory database. This component provides access to all standard BPMS features and extends its functionality with location-specific features. This includes displaying the customer locations and the current locations of items such as tools on a map (Fig. 6). It also supports the incorporation of additional contextual information for existing customers.

After modeling the location-aware process using the included modeler, we need the BPMS to access the process model. Because we extended the meta-model of BPMN in our location-aware extension, the business process model adheres to the standard XML format and could be used with a different BPMS, provided there is an implementation for the location-aware events. After providing the necessary parameters, such as event type, candidate user/group, and geofence areas (depending on the patterns used), we can use BPMS (in this case, Camunda) or mobile application to initiate a new process instance as usual.

Depending on the deployed process model, the next step involves providing access to various location information. This could involve including a database with known customer locations or accessing third-party data through a standard interface. In the implementation presented here, all expressions are evaluated within the BPMS, meaning that all operators are implemented within the BPMS itself. However, for real-world applications, additional components could be considered. For example, middleware for preprocessing and collecting data could be added, and another system could be used to outsource the required logic. This could reduce the computational load on the BPMS, especially for complex calculations.

The implementation follows the well-established data patterns described in [54], which explain the usage and exchange of data in implementing process models. These patterns are a fundamental bridge between conceptual solutions and well-reasoned prototypical implementations (see Sect. 6).

Following this basic implementation of the presented patterns, in [12], we develop a BPMN-extension for location-awareness (*LABPMN*, cf. Section 5) that substitutes the used script tasks by metamodel-conform extension elements without changing much of the application visible to the user. In [59], we provide a short overview of the updated implementation of the BPMN extension by extending the modeler and implementing support for the execution in a BPMS. Coming back to the formal description of the spatial operators (cf. Section 3.2), Fig. 11 (Page 41) shows the added parameters needed for the execution that were previously hidden from

[14] https://camunda.com

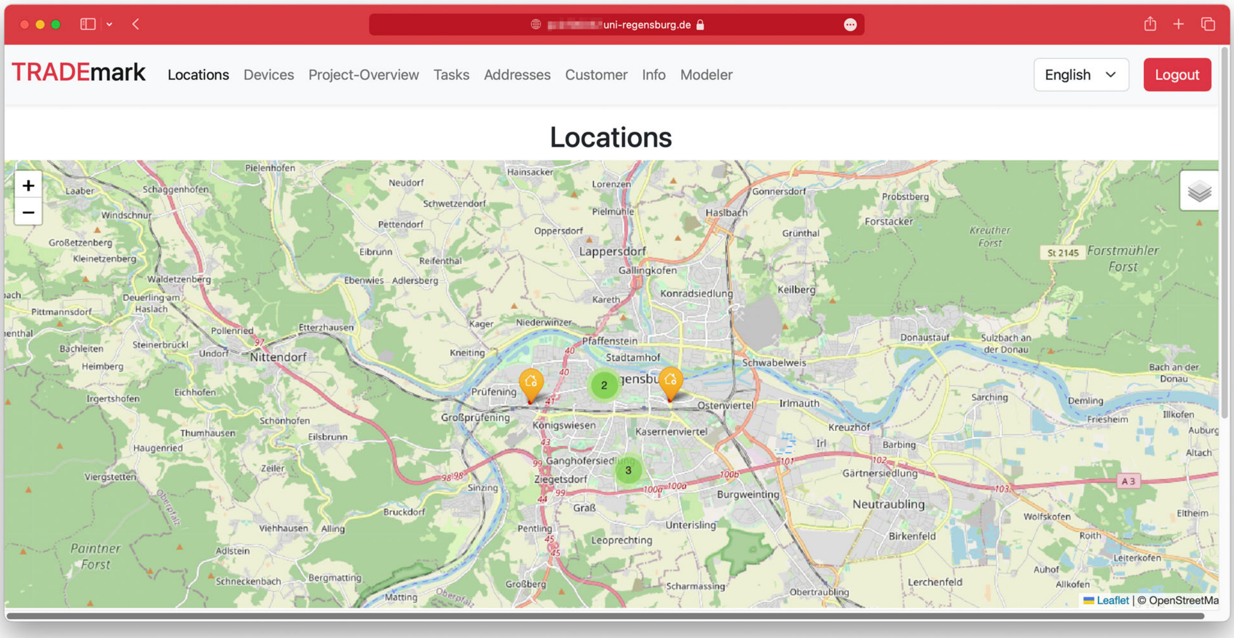


Fig. 6 Back-office application showing customer locations and the locations of current construction sites, the foundation of location handling, and deeply integrated with the BPMS

the modeler as well as process executors in the tasks getting automatically executed by the BPMS.

Besides better support and more precise visual depiction of the use of location data within the process model (a location event that is attached to a task is more expressive than a generic signal or conditional event and explicitly implies the use of location data or the allocation of a task to an actor, where the reason for the allocation is directly visible within the process model), using LABPMN as an exemplary implementation for the use of presented patterns also enables more specific handling of, for example, the allocation. Here, a subprocess element in BPMN can be used to control and depict the allocation duration.

8 Location-aware business process applications

This section thoroughly evaluates the proposed patterns by examining their applicability in four distinct real-world application domains. The first three provided use cases are actual processes from our research project currently used in operational business and are only slightly adjusted for academic publication. At the same time, the last one is adapted from [15] to show the transferability and generalizability to different application areas (here: emergency services). Our analysis encompasses the following scenarios:

1. *Conventional repair process employed by a crafts business*
   The first application scenario demonstrates the effectiveness of the proposed patterns in a conventional repair process. The evaluation showcases how these patterns enhance the operational efficiency and adaptability of the repair workflow within craft businesses, where tasks are allocated based on the customer and actor's location, and additional information is provided based on historical data on the customer's location.
2. *Delivery of important objects between branches of a company*
   In the second scenario, we delve into a use case involving the surveilled delivery of objects between the branch and the main office of an industry partner. Here, the incorporation of the current location of the parcel and additional sensor data into the operational framework is critical for traceability and documentation.
3. *Rental process enhancement through location information*
   The third scenario demonstrates the use of location information to enhance the rental process in a crafts business. In this case, the customer can automatically obtain and return their device without requiring personnel from the company. This improvement improves customer satisfaction and ease of use by automating and supporting the complete process based on the device's location.

4. *Emergency Incident Management*
   The last scenario uses the process for incident response management in [15]. Similar to the previous processes, the distributed character of the process, in which an ambulance gets dispatched and drives to the patient, already implies the high importance of location data. Both the decision for the type of needed service and the allocation of the emergency to one (or possibly multiple) emergency units include location data. Different regulatory requirements demand regular documentation of the current status, and depending on the location, the central call center can treat the patient accordingly.

The following cases illustrate a diverse range of industry-specific scenarios, demonstrating the versatility and effectiveness of the presented patterns. Each scenario demonstrates the patterns in action and provides a unique implementation perspective, shedding light on their architectural utility. DSRM requires knowledge to be generalizable and transferable, which we will show with these location-aware processes. We classify the different applications according to the used *technology* (cf. Section 3.1), the applied *operators* (cf. Section 3.2), and the included *patterns* (cf. Section 4).

It is important to note that the last two patterns, worklist reordering, and constrained execution, are not explicitly incorporated into the following exemplary applied processes. Worklist reordering is based on considering different parameters from multiple concurrently running instances and is thus not easily depictable within a singular process model. Similarly, constrained execution and augmented process modeling approaches, although applicable to location data, face visualization limitations within standard BPMN elements. Consequently, implementing these patterns must be separated from the business process management system (BPMS).

8.1 Location-aware repair process (P1)

Technology: BLE-Tags, Cellular, GPS
Operator: Minimum distance, geofence, location update
Pattern: Decision making, task allocation, event dispatching, process monitoring, automatic completion of tasks (implied worklist reordering[15])

Driven by a real-life application from our research project, we explain the use of spatial information within a mobile distributed business process from craft businesses, where workers drive to customers to replace broken heaters. The BPMN model of the process is depicted in Fig. 7.

The process starts with an incoming job that includes the customer's location (either by entering it using the given form or by querying already known customers). Because we want our company to act sustainably, we check if any currently working actors are reasonably far away, i.e., using all currently known locations and checking if any actor satisfies the condition of being inside a specific distance from the customer. If not, we inform the customer about a delay and finish the process instance. For this example, if we have at least one possible candidate, the process proceeds, and we use the worker's location and the customer's location again to allocate the closest available actor to the process. The location-aware subprocess is used to show and implement the lifespan of the allocation, as already explained above; all enclosed tasks are to be executed by the same actor.

Our allocated actor immediately starts driving toward the customer, periodically updating their location. Based on their current location and the expected location, we can notify the customer if the actor is behind schedule or automatically adjust following tasks from different process instances; on the other hand, we also check for the actor entering a geofence around the customer, where we automatically inform the customer about the imminent arrival. When the actor arrives at the customer, a location-aware event is thrown to update the actor's status and provide information for invoicing in another process lane. After repairing the heater at the customers' location, the actor drives back to our depot to get their stock refilled, which is again triggered using a throwing-location event, caught by a location-start event for a different business process of another department.

Looking at the process model, it becomes evident that location information is directly visible for the allocation of the actor and the triggered notifications for the customer. These notifications are initiated by entering geofences around the customer's location. Furthermore, the two intermediate location events enable the employees' current location (including assigned and used tools) to be used by other parts of the business. Finally, we can use information collected from both the BPMS, information gained by sensors, and location data to fully automate or at least support the creation of an invoice for the order.

8.2 Location-aware transport process of logistics company (P2)

Technology: Barcode, RFID, visual
Operator: Geofence, location update
Pattern: Start of processes, task assignment, event dispatching, completion of tasks

Figure 8 illustrates an enhanced process model representing a second real-world application that demands complete transparency in the delivery of goods between two distinct branches of an industry partner manufacturing orthopedic medical products. Analogously to the preceding process, integrating location information is pivotal in overseeing the

[15] Not visible within the process model, as explained above

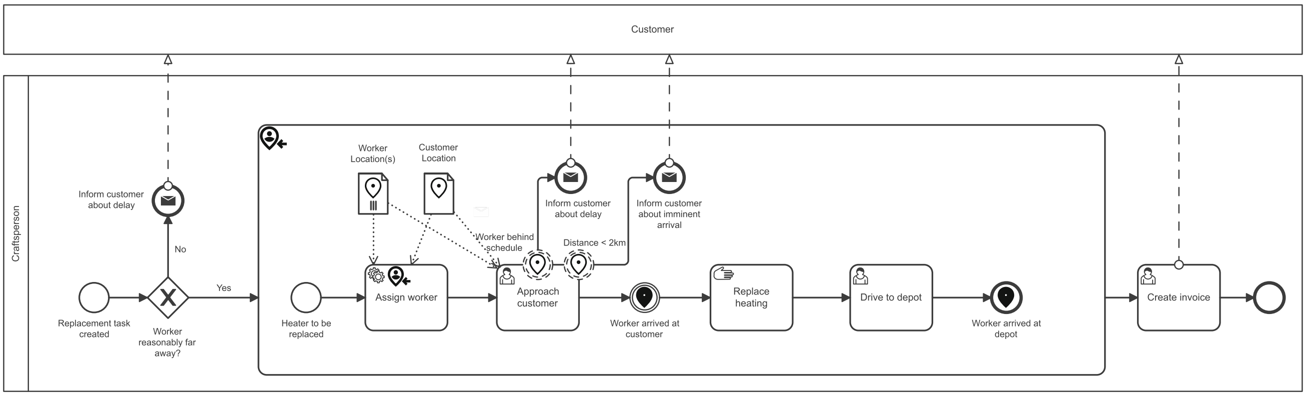


Fig. 7 Exemplary real-life process from a craft business enhanced with location data

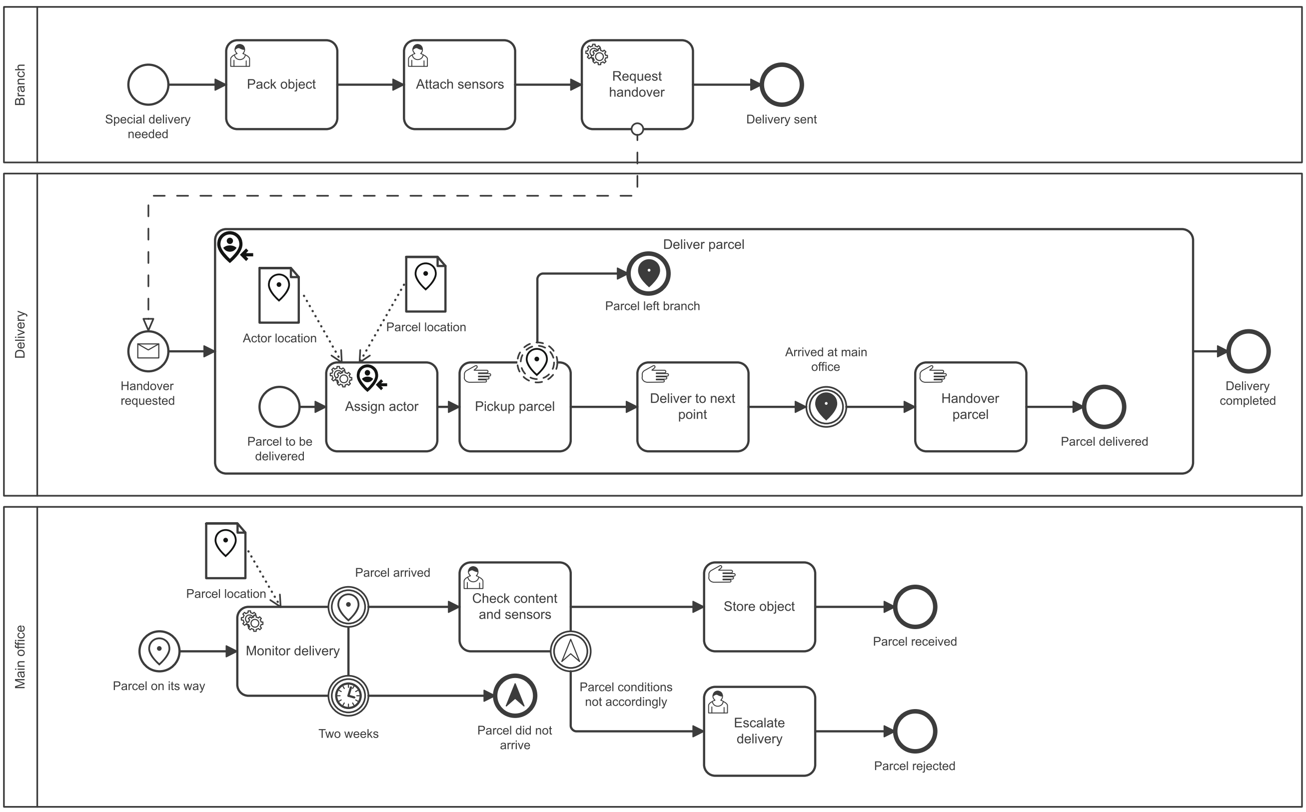


Fig. 8 Exemplary real-life process for parcel delivery between branches enhanced with location data

seamless delivery of a parcel from one of the company's branches to the central office.

The process begins at a branch of the industry partner, where an object is packed, including sensors capturing data such as humidity and temperature. Subsequently, the branch initiates a request for handover from the company's delivery department. Upon notification of the impending delivery, a selection process ensues within the delivery department, akin to the previous process. The chosen actor for delivery is determined based on contextual information encompassing their skill matrix, external specifications, and the current location, with the latter being contingent on the nature of the object to be delivered.

Following the selection process, the designated actor retrieves the parcel, initiating monitoring at the main office. The parcel is then transported to its destination. On arrival, a handover event is modeled for the main office. At this point, the package is received, subject to examination, and its condition evaluated. If the condition is deemed satisfactory, the object is securely stored, and the process is concluded. Conversely, if an issue is identified, the delivery undergoes escalation, triggering an alternate process in the real-world scenario. This intricate system ensures that the delivery process is transparent and equipped to handle deviations from the norm, enhancing overall operational robustness.

Similar to the previous process, the location of the parcel is the root of the process, enabling traceability through modeling elements that ease documentation tasks during execution while simultaneously increasing transparency during the modeling.

8.3 Location-aware rental process (P3)

Technology: GPS, WiFi
Operator: Geofence, location update
Pattern: Start of processes, event dispatching, monitoring execution, automation of tasks

The last process is based on a rental process (see Fig. 9). As part of its core business, a crafts company rents out mobile heaters for emergencies or during renovation/construction. The process here starts with a concluded contract that triggers the initiation of the rental process with the craftspeople. They prepare the unit and then either plan and transport it to the customer or have the customer pick it up themselves. The unit is automatically registered when it leaves the company premises and the rental period begins. During the rental period, the unit is automatically remotely monitored with access from the back office through a dashboard.

Fitting the rented objects with sensors for location can also provide additional access to information that can be used for troubleshooting or predictive maintenance. Again, directly using the location of the involved entities in the process model provides additional information that can then be used, increasing transparency in execution while making the spatial context of the process visible at first glance.

At any given point or the end of each month, the customer can decide to return the unit, triggering the return delivery (similar to the delivery to the customer but including a final examination after the return). This, in turn, automatically triggers the final invoice creation when the unit is back in the company. At the end of each month, the system automatically generates an invoice based on the current location and usage parameters.

8.4 Emergency incident management (P4)

Technology: Cellular, GPS, Emergency Services Software
Operator: geofence, location update
Pattern: Decision making, task allocation, event dispatching, process monitoring, automatic completion of tasks

Another application area that comes to mind for location services is emergency services, where the location of the victim or patient is of utmost importance. Compared to the three other processes that stem from a current research project and are running processes and business partners, the process shown in Fig. 10 is a case study with Australian emergency services presented in [15]. Here, the authors already include location-specific events in a non-standard conform way to describe the process.

The agent begins by determining the appropriate response to the incoming emergency call. This may involve dispatching an ambulance (or, in the original example, airborne emergency services, which we excluded for better clarity). Once the dispatch request is received at the ambulance station, a unit is dispatched automatically based on its current location and other factors. Upon arrival at the scene, the unit checks for treatment. If so, the patient is approached. Each location update of the mobile unit must be recorded and saved to the record management system. During the patient's treatment, the emergency call center provides specialist medical advice based on the most recent observations. If a patient requires transportation to a hospital, they are loaded and transported to the hospital, where they are dropped off at the emergency department (ED). The arrival at the hospital is broadcast to the call center agent as a status update. Once the emergency has been completed, the computer-aided dispatch software is cleared, and the dispatch is finished for both the mobile unit and the agent in the call center. During the dispatching process, an event may occur that necessitates a recall or retasking of the unit, interrupting both subprocesses and immediately concluding the dispatching.

In addition to the various status updates that are recorded and utilized to facilitate the process at the call center, the patient's location is also employed to determine the appropriate response and necessary service and dispatch the specific unit. This process involves decision-making based on location data, the allocation of tasks, and the dispatching of events, in addition to monitoring tasks with location information and completing tasks based on an actor's location. In contrast to the process presented in [15], the process model's direct use of location patterns is evident. Assuming the external data sources and sinks provide appropriate interfaces, the process would also be directly executable if the implementation detailed above were to be used.

8.5 Potentials enabled through the presented patterns

Integrating location data into business processes offers a large number of potentials that can significantly impact real-world applications across diverse industries. One of the foremost advantages lies in the real-time optimization and execution of business processes. Companies can dynamically enhance supply chain efficiency by incorporating location data into their processes. This is achieved by adjusting routes in response to the current location of goods, tools, actors, and other involved entities that directly impact operational effectiveness (*P1, P2, P3, P4*).

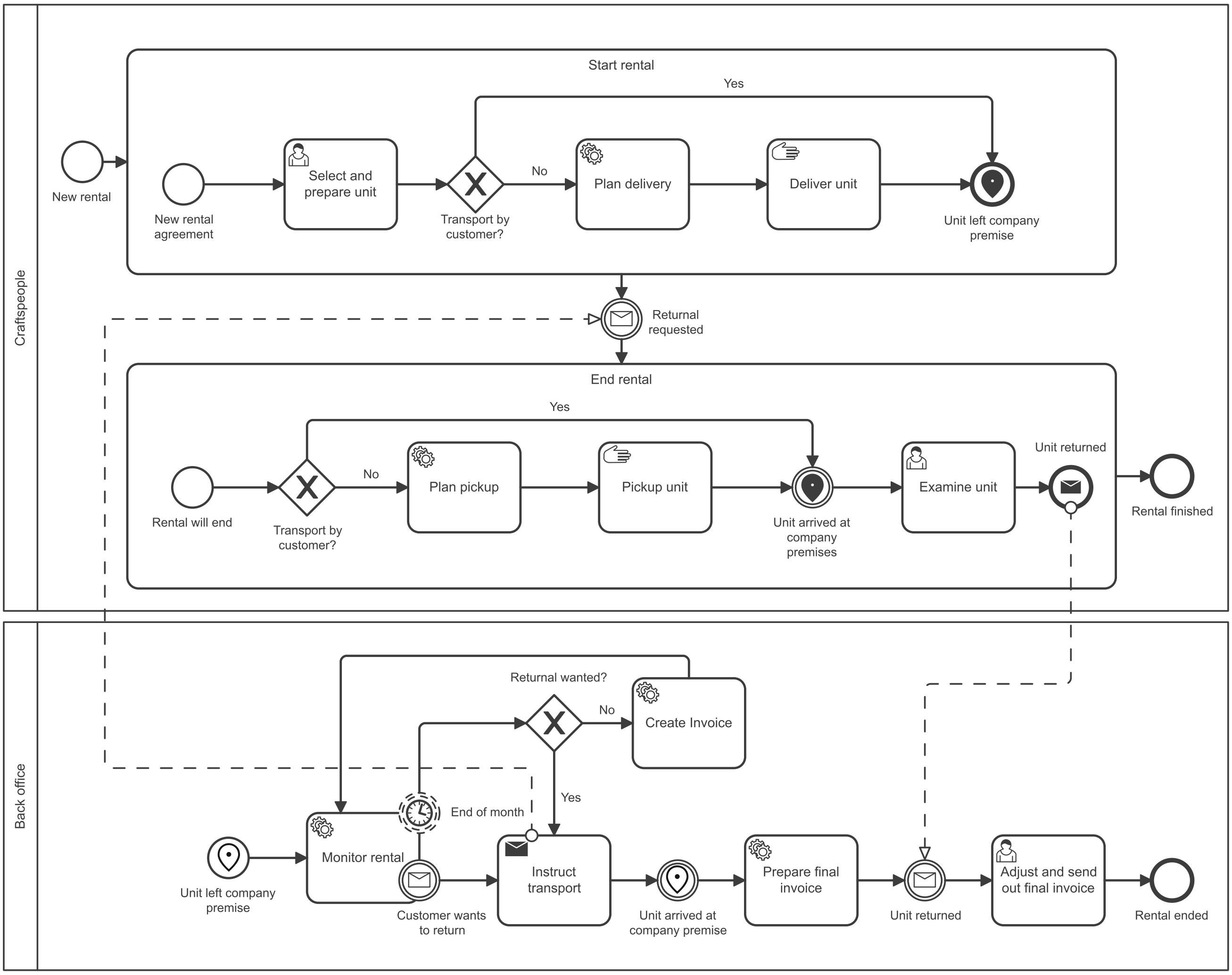


Fig. 9 Exemplary real-life process for the rental of mobile heating units enhanced by location data

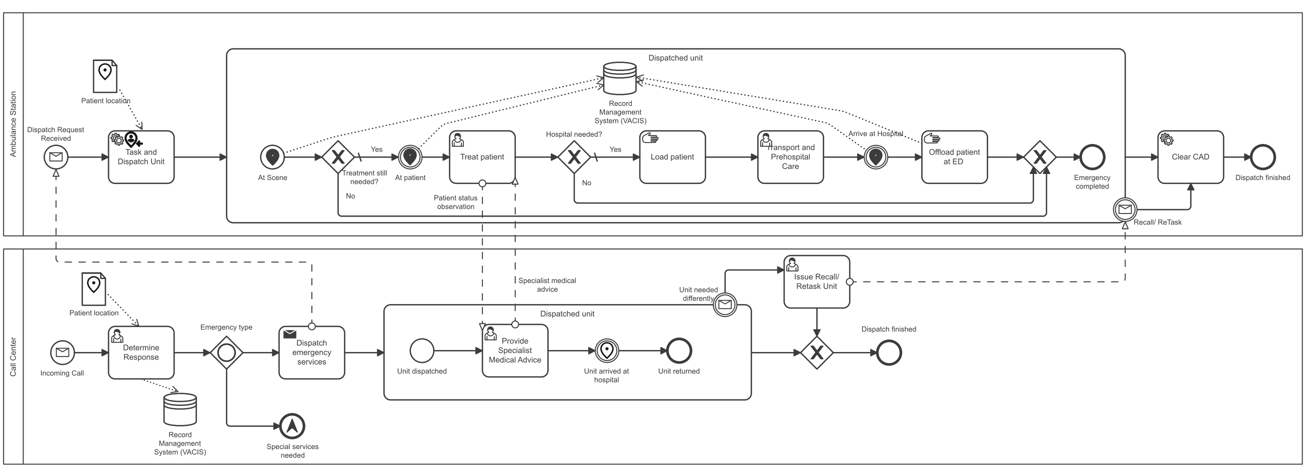


Fig. 10 Emergency services process from [15] enhanced with location-aware patterns

Automated decision-making based on location information represents a robust potential. Configuring the BPM platform to respond to specific location conditions automatically ensures streamlined processes. For instance, tasks can be automatically assigned, or processes can be triggered when an IoT device reaches a predefined location, demonstrating a sophisticated utilization of location data in optimizing workflow (*P1, P3, P4*).

In industries where risk management is critical, the integration of location data proves invaluable. Sectors such as logistics and healthcare can adapt processes in real-time to respond to unexpected events in specific locations, thus improving overall risk mitigation strategies (*P2, P4*).

The customer-centric approach is further advanced through the use of location data. Companies can tailor their services precisely to customer needs by using location information (*P1, P2, P3*).

Integrating location data allows companies to optimize resources such as personnel deployment and inventory management more effectively, improving their efficiency through resource optimization. This optimization contributes to streamlined operations and resource utilization (*P1, P2, P3*).

Real-time monitoring of assets is a critical application in manufacturing and logistics. Location data ease the continuous monitoring of assets like machines, vehicles, dispatched units, and products. This capability enables proactive maintenance, avoids bottlenecks, and enhances overall operational efficiency in these industries *(P1, P2, P4)*.

Lastly, location data play a pivotal role in industries where regulatory compliance and safety standards impose strict requirements for documentation. The integration of location information (*P2, P4*) promises to facilitate the assurance that business processes adhere to required policies.

9 Conclusion and future work

As seen in the related works, using spatial information to facilitate locally distributed work within a factory hall, or even mobile work, is not new and has been part of research for nearly 20 years. Both are different from the combination of spatial information and business processes. By using IoT devices as enablers and coming from a vastly different real-world application area rooted in different executable processes at its base, this paper closes the gap between conceptual approaches for using location in BPM and the real world.

In this work, we summarize different ways of using contextual location information for the process based on the location of different entities, facilitating process execution and enabling further automation of process execution. Compared to earlier approaches, the exemplary implementation enables efficient execution using Camunda as the BPMS. It can be easily extended because of the strict separation between the different systems (the BPMN-extension based on the patterns identified here (cf. [12]) further develops the usability by reintroducing the logic directly into the BPMS based on the BPMN meta-model).

All main parts of single processes can be enriched using the location as contextual information during execution, from simply using the distance between two locations in an expression for logic gateways or start and end events (starting and finishing a process) and tasks (allocating a task to the user closest to an entity) up to terminating tasks based on the spatial information and dispatching events based on an entities location or the distance between two entities. Considering multiple concurrently running process instances, we can use location data to intelligently reorder the task list of all actors, use multiple different actors as candidates for the allocation of tasks, or gain insights through monitoring their execution. By first presenting the different identified patterns while providing an implementation for their use during modeling and execution, we answer both research questions motivating this publication. The presented list of possibilities shows the potential of combining IoT and BPM by using contextual information in BPM to improve efficiency and the workflow of workers working within mobile distributed processes.

While the focus of the included patterns is on actively using location data for different appliances, future research could broaden this definition to include the use of location for dynamically adjusting the price of a service or to use the location as part of a more extensive set of parameters for the selection of for example an advertisement. Centering around this use of location data would vastly change the perspective on the business processes but could provide additional insights into the topic. On the other hand, for the deployment using BPMN, we already introduced an extension for depicting the location and using it within business process models, supporting the patterns presented here. Staying at BPMN as one way of implementing the patterns, different new challenges can arise by showing more interdependencies between different parts of the process models, and while these are possibly also existent in the real world, the inclusion in the model as a simplified representation could decrease readability and understandability.

Using context information in BPM is a promising approach for new ways of orchestrating company business processes. While this work focuses on location information, future endeavors could take a more comprehensive look at contextual information and its utilization in BPM. For instance, contextual information can be leveraged during the execution of otherwise unsupported manual tasks enabled through the IoT.

Redesigning and implementing new processes with research partners is time-consuming. Nevertheless, we also plan a quantitative evaluation based on classic process metrics such as throughput and cycle times while considering improved customer satisfaction and advancements through transparency and business process improvement. The exact interaction and connections of pull- and push-communication, data patterns, and appliances are tied to specific implementations and decisions, leaving their generalizability for future assessment. The design decision regarding the level or system within which specific calculations and decisions occur is closely related.

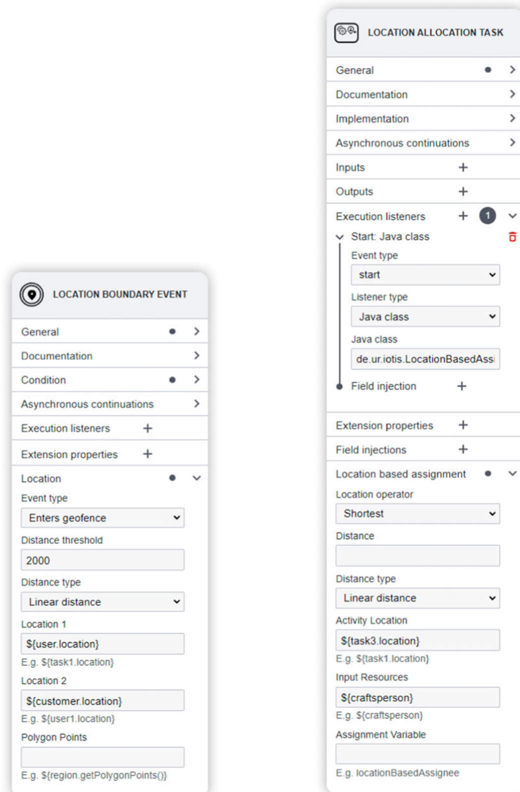
Context information, such as the location of entities (actors, customers, and other involved entities), can be utilized in business processes to facilitate and enhance a company's value-creation process. The capacity to map relationships between locations of process entities opens up a range of new implications and possibilities for analysis and improvement. This can lead to the development of a more powerful and expressive process management.

Appendix A Business process modeling notation (BPMN)

Business process modeling notation (BPMN) provides a "standardized bridge for the gap between the business process design and process implementation" [18] by providing a graphical representation of business processes and a blueprint for the supported execution by business process management systems (BPMS). As a short introduction to the following patterns for the location-aware business process execution, which is tied to but not dependent on BPMN, different technical terms have to be introduced: The primary purpose of describing a business process is to document and standardize the sequence of tasks and decisions made to contribute to the businesses value creation [1].

BPMN describes different types of tasks, from script tasks that happen fully automated and supported by the BPMS to user tasks that are mainly executed by human actors but supported by the BPMS and connected systems to manual tasks that are entirely ignored by the BPM and completely executed by human actors. The flow between those tasks can be directed using gateways that divide, redirect, or join the process flow (the most used gateways are the "AND"-Gateway as well as the "OR"-Gateway). Next to using gateways within a flow, business process analysts can use events attached to tasks to react to specific changes during task execution (so-called boundary events).

Appendix B Location-aware parameters for modeling



(a) Location parameters for location-aware events (b) Location parameters for allocation and assignment

Fig. 11 Extension of the modeler with parameters for location-awareness

Acknowledgements This work is funded by the "Bayerische Forschungsstiftung (BFS)" within the project *IoT-basiertes Daten- und Prozessmanagement im Handwerk (TRADEmark)*.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of business process management. Springer, Berlin Heidelberg (2018)
2. Compagnucci, I., Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F.: Modelling notations for IoT-aware business processes: a systematic literature review. In: Business process management workshops, pp. 108–121. Springer International Publishing, New york (2020)
3. Gershenfeld, N., Krikorian, R., Cohen, D.: The internet of things. Sci. Am. **291**(4), 76–81 (2004). https://doi.org/10.1038/scientificamerican1004-76
4. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: Handheld and ubiquitous computing, pp. 304–307. Springer, Berlin Heidelberg (1999)
5. Huebscher, M.C., McCann, J.A.: Adaptive middleware for context-aware applications in smart-homes. In: Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing, pp. 111–116. ACM Press (2004)
6. Dörndorfer, J., Seel, C.: A framework to model and implement mobile context-aware business applications. In: Schaefer, I., Karagiannis, D., Vogelsang, A., Méndez, D., Seidl, C. (eds.) Modellierung 2018, pp. 23–38. Gesellschaft für Informatik e.V, Bonn (2018)
7. Behkamal, B., Pourmasoumi, A., Rastaghi, M.A., Kahani, M., Motahari-Nezhad, H.R., Allahbakhsh, M., et al.: Geo-enabled business process modeling. In: 18th international conference on business process management (2020). https://doi.org/10.48550/ARXIV.2204.08063
8. Hallerbach, A., Bauer, T., Reichert, M.: Context-based configuration of process variants. In: 3rd international workshop on technologies for context-aware business process management (TCoB 2008), pp. 31–40 (2008)
9. Rosemann, M., Recker, J., Flender, C.: Contextualisation of business processes. Int. J. Bus. Process Int. Manage. **3**(1), 47 (2008). https://doi.org/10.1504/ijbpim.2008.019347
10. Wieland, M., Kopp, O., Nicklas, D., Leymann, F.: Towards context-aware workflows. In: CAiSE07 proc. of the workshops and doctoral consortium. vol. 2, p. 78 (2007)
11. Beynier, A., Chevaleyre, Y., Gourvès, L., Harutyunyan, A., Lesca, J., Maudet, N., et al.: Local envy-freeness in house allocation problems. Auton. Agents Multi-Agent Syst. **33**(5), 591–627 (2019). https://doi.org/10.1007/s10458-019-09417-x
12. Poss, L., Dietz, L., Schönig, S.: LABPMN: Location-Aware Business Process Modeling and Notation. In: Sellami, M., Vidal, M.E., van Dongen, B., Gaaloul, W., Panetto, H. (eds) Proceedings of the international conference on cooperative information systems (CoopIS) 2023 (2023)
13. Poss, L., Schönig, S.: A generic approach towards location-aware business process execution. In: van der Aa, H., Bork, D., Proper, H., Schmidt, R. (eds.) Enterprise, business-process and information systems Mmodeling. Springer International Publishing, Cham (2023)
14. Russell, N., ter Hofstede, A.H.M., van der Aalst, W.M.P., Mulyar, N.: Workflow control-patterns - A Revised View. BPM reports. 2006;0622
15. Andrews, R., Wynn, M., Vallmuur, K., ter Hofstede, A., Bosley, E., Elcock, M., et al.: Leveraging data quality to better prepare for process mining: an approach illustrated through analysing road trauma pre-hospital retrieval and transport processes in Queensland. Int. J. Environ. Res. Public Health **16**(7), 1138 (2019). https://doi.org/10.3390/ijerph16071138
16. vom Brocke, J., Zelt, S., Schmiedel, T.: On the role of context in business process management. Int. J. Inf. Manage. **36**(3), 486–495 (2016). https://doi.org/10.1016/j.ijinfomgt.2015.10.002
17. Weske, M.: Business process management. Springer, Berlin Heidelberg (2007)
18. OMG.: Business process model and notation (BPMN), Version 2.0.2. Available from: http://www.omg.org/spec/BPMN/2.0.2/
19. Zarour, K., Benmerzoug, D., Guermouche, N., Drira, K.: A systematic literature review on BPMN extensions. Bus. Process Manage. J. **26**(6), 1473–1503 (2019). https://doi.org/10.1108/bpmj-01-2019-0040
20. Braun, R., Esswein, W.: Classification of domain-specific BPMN extensions. In: Lecture notes in business information processing, pp. 42–57. Springer, Berlin (2014)
21. OMG.: Meta object facility (MOF) core specification, Version 2.5.1. Available from: https://www.omg.org/spec/MOF/2.5.1
22. de Morais, R.M., Kazan, S., de Pádua, S.I.D., Costa, A.L.: An analysis of BPM lifecycles: from a literature review to a framework proposal. Bus. Process Manage. J. **20**(3), 412–432 (2014). https://doi.org/10.1108/bpmj-03-2013-0035
23. Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Di Ciccio, C., et al.: The internet of things meets business process management: a manifesto. IEEE Syst. Man Cybernet. Mag. **6**(4), 34–44 (2020). https://doi.org/10.1109/msmc.2020.3003135
24. Schönig, S., Ackermann, L., Jablonski, S., Ermer, A.: IoT meets BPM: a bidirectional communication architecture for IoT-aware process execution. Softw. Syst. Model. **19**(6), 1443–1459 (2020). https://doi.org/10.1007/s10270-020-00785-7
25. Küpper, A.: Location-based services - fundamentals and operation. Wiley, Hoboken (2005)
26. Etzion, O.: Event processing in action. Manning (2011)
27. Soffer, P., Hinze, A., Koschmider, A., Ziekow, H., Ciccio, C.D., Koldehofe, B., et al.: From event streams to process models and back: challenges and opportunities. Inform. Syst. **81**, 181–200 (2019). https://doi.org/10.1016/j.is.2017.11.002
28. Page, M.J., McKenzie, J.E., Bossuyt, P.M., et al.: The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. BMJ (2021). https://doi.org/10.1136/bmj.n71
29. Compagnucci, I., Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F.: A systematic literature review on IoT-aware business process modeling views, requirements and notations. Softw. Syst. Model. (2022). https://doi.org/10.1007/s10270-022-01049-2
30. Torres, V., Serral, E., Valderas, P., Pelechano, V., Grefen, P.: Modeling of IoT devices in business processes: a systematic mapping study. In: 2020 IEEE 22nd conference on business informatics (CBI). IEEE (2020)
31. Webster, J., Watson, R.T.: Analyzing the past to prepare for the future: writing a literature review. MIS Q. **26**(2), 13–23 (2002)
32. Stürzel, P.: Modellierung und Ausführung von workflows unter Berücksichtigung mobiler Kontextinformationen. KIT Scientific Publishing, Karlsruhe (2018)
33. Decker, M., Che, H., Oberweis, A., Stürzel, P., Vogel, M.: Modeling mobile workflows with BPMN. In: 2010 ninth international conference on mobile business and 2010 ninth global mobility roundtable (ICMB-GMR). IEEE (2010)
34. de Leoni, M., Adams, M., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Visual support for work assignment in process-aware information systems: framework formalisation and implementation. Decis. Support Syst. **54**(1), 345–361 (2012). https://doi.org/10.1016/j.dss.2012.05.042
35. Tchemeube, R.B., Amyot, D., Mouttham, A.: Location-aware business process management for real-time monitoring of a cardiac care process. CASCON. **13**, 230–244 (2013)
36. Schönig, S., Zeising, M., Jablonski, S.: Towards location-aware declarative business process Management. In: Business infor-

mation systems workshops, pp. 40–51. Springer International Publishing, New york (2014)
37. Chiu, H.H., Wang, M.S.: Extending event elements of business process model for internet of things. In: 2015 IEEE international conference on computer and information technology; ubiquitous computing and communications; dependable, autonomic and secure computing; pervasive intelligence and computing. IEEE, pp. 783–788. IEEE (2015)
38. De Loof, J., SAP, C.M., Meissner, S., Nettsträter, A., CEA, A.O., SAP, M.T.: Internet of things-architecture iot-a deliverable d1 5-final architectural reference model for the iot v3. 0. IoT-A **257521**, 1–499 (2013)
39.) Jeng, J., Huff, K., Hurwitz, B., Sinha, H., Robinson, B., Feblowitz, M.: WHAM: supporting mobile workforce and applications in workflow environments. In: proceedings tenth international workshop on research issues in data engineering. RIDE 2000. IEEE Comput. Soc (2000)
40. Aoumeur, N., Fiadeiro, J., Oliveira, C.: Towards an architectural approach to location-aware business process. In: 13th IEEE international workshops on enabling technologies: infrastructure for collaborative enterprises. IEEE Comput. Soc (2004)
41. Stephan, P.: System architecture for using location information for process optimization within a factory of things. In: proceedings of the 3rd international workshop on location and the web - LocWeb '10. ACM Press (2010)
42. Gopalakrishnan, S., Sindre, G.: Diagram notations for mobile work processes. In: Lecture notes in business information processing, pp. 52–66. Springer, Berlin (2011)
43. Zhu, X., Recker, J., Zhu, G., Santoro, F.M.: Exploring location-dependency in process modeling. Bus. Process Manage. J. **20**(6), 794–815 (2014)
44. Zhu, X., van den Broucke, S., Zhu, G., Vanthienen, J., Baesens, B.: Enabling flexible location-aware business process modeling and execution. Decis. Support Syst. **83**, 1–9 (2016). https://doi.org/10.1016/j.dss.2015.12.003
45. Shinde, S., Portier, B. (2012) Geo-location improves business process management human task assignment. IBM Acad. TechNotes. 3 (6)
46. Grambow, G., Hieber, D., Oberhauser, R., Pogolski, C.: A context and augmented reality BPMN and BPMS extension for industrial internet of things processes. In: Business process management workshops, pp. 379–390. Springer International Publishing, New york (2022)
47. Hevner, M., Park, R.: Design science in information systems research. MIS Q. **28**(1), 75–105 (2004). https://doi.org/10.2307/25148625
48. Johannesson, P., Perjons, E.: An introduction to design science. Springer International Publishing, Berlin (2021)
49. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. J. Manage. Inform. Syst. **24**(3), 45–77 (2007). https://doi.org/10.2753/mis0742-1222240302
50. Khudhair, A.A., Jabbar, S.Q., Sulttan, M.Q., Wang, D.: Wireless indoor localization systems and techniques: survey and comparative study. Indones. J. Electr. Eng. Comput. Sci. **3**(2), 392 (2016). https://doi.org/10.11591/ijeecs.v3.i2.pp392-409
51. Pascacio, P., Casteleyn, S., Torres-Sospedra, J., Lohan, E.S., Nurmi, J.: Collaborative indoor positioning systems: a systematic review. Sens. basel. **21**(3), 1002 (2021). https://doi.org/10.3390/s21031002
52. Asaad, S.M., Maghdid, H.S.: A comprehensive review of indoor/outdoor localization solutions in IoT era: research challenges and future perspectives. Comput. Netw. **212**, 109041 (2022). https://doi.org/10.1016/j.comnet.2022.109041
53. Kozel, T.: BPMN Mobilisation. In: proceedings of the European conference of systems, and European conference of circuits technology and devices, and European Conference of communications, and European conference on computer science. ECS'10/ECCTD'10/ECCOM'10/ECCS'10, pp. 307–310. World scientific and engineering academy and society (WSEAS), Stevens Point, Wisconsin, USA (2010)
54. Russell, N., Aalst, W.M.P.V.D., Hofstede, A.T.: Workflow patterns - the definitive guide. MIT Press (2015)
55. Taylor, J., Fish, A., Vanthienen, J., Vincent, P.: Emerging standards in decision modeling. In: intelligent BPM systems: impact and opportunity. iBPMS Expo; Chicago (US), pp. 133–146 (2013)
56. Pflug, J., Rinderle-Ma, S.: Analyzing the effects of reordering work list items for selected control flow patterns. In: 2015 IEEE 19th international enterprise distributed object computing workshop. IEEE (2015)
57. Stroppi, L.J.R., Chiotti, O., Villarreal, P.D.: Extending BPMN 20: method and tool support. In: Lecture notes in business information processing, pp. 59–73. Springer, Berlin (2011)
58. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., Zhao, W.: A survey on internet of things: architecture, enabling technologies, security and privacy, and applications. IEEE Int. Things J. **4**(5), 1125–1142 (2017). https://doi.org/10.1109/jiot.2017.2683200
59. Poss, L., Jasinski, R., Market, M., Schönig, S.: TRADEmark - Using location data for mobile distributed processes in BPM. In: CoopIS'23: Cooperative Information Systems (2023)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Leo Poss is a research assistant with the Department of Information Systems at the University of Regensburg in Germany. He received his master's degree in Information Systems from Otto-Friedrich-Universität Bamberg. His research interests lie in the human aspects of process management and the Internet of Things (IoT). Based on his research, he has published scientific papers in international conferences and journals.

Stefan Schönig is Full Professor for Information Systems with the Department Information Systems at University of Regensburg in Germany. He received both the master's degree (with honours) in Applied Computer Science (Engineering/Computer Science) and the doctoral degree from University of Bayreuth. Before, he held a position as a tenured assistant professor at University of Bayreuth. He was a post-doctoral researcher with the Institute for Information Business at WU Vienna (Vienna University of Economics and Business). He has an established background in BPM/Process Mining and IoT research and has been working in this field for over 12 years. He published extensively in the research area of BPM and information systems, both in international conferences and journals.