An Evaluation Framework for Design-Time Context-Adaptation of Process Modelling Languages

Jing Hu, Ghazaleh Aghakhani, Faruk Hasić, Estefanía Serral

An Evaluation Framework for Design-Time Context-Adaptation of Process Modelling Languages

Jing Hu, Ghazaleh Aghakhani, Faruk Hasić and Estefanía Serral

Leuven Institute for Research on Information Systems (LIRIS), KU Leuven
{jing.hu; ghazaleh.aghakhani}@student.kuleuven.be
{faruk.hasic; estefania.serralasensio}@kuleuven.be

Abstract To enhance the performance and efficiency of business processes, it is essential to take the dynamics of their execution context into account during process modelling. This paper first proposes an evaluation framework that identifies the main requirements for supporting the modelling of context-adaptive processes. Using this framework, we analyse four popular business process modelling languages: Coloured Petri Nets (CPN), Business Process Modelling and Notation 2.0 (BPMN), Yet Another Workflow Language (YAWL), and Unified Modelling Language Activity Diagrams (UML AD). The analysis is carried out by evaluating how the respective language notations fulfil the identified requirements in several real-life scenarios. Lastly, a comparative analysis of the languages focussed on their support for modelling context-adaptive business processes is provided.

Keywords: Context adaptation, business process modelling, process notation

1 Introduction

The central focus of business processes is on the way the activities are performed within an organization. The modelling of business processes has the goal of mapping the current process of the organization ("as-is") to the future process ("to-be"). Successful modelling eventually leads to favourable results in Business Process Management (BPM), which incorporates the improvements. Therefore, business process modelling is the most critical step in the BPM lifecycle [1, 2].

Essentially, a business process is designed to achieve certain objectives and goals. Nevertheless, it may fail to do so due to the divergence caused by the dynamic changes and unexpected events during the lifetime of the business process [3]. Being unable to adapt accordingly to the ever-changing business environment results in a high failure rate of BPM projects [4]. To tackle this problem, [5] suggests that one of the critical principles for success is the principle of context adaptation, which means the practitioners must understand and identify relevant context factors when designing a business process, hence enabling flexibility to deal with internal and external contingencies. Therefore, it is of paramount importance to employ a process language that supports context-adaptation. Such language must provide high expressiveness in terms of control flow, as well as in capturing the significant execution contexts.

Considerable number of approaches have been proposed to support the above principle in terms of both modelling perspectives and business languages. Yet, neither of these approaches provide a solid universal way to construct a Context-Adaptive Business Process (CABP), nor are the proposed different modelling languages compared in this aspect. Hence, this paper aims to compare the most popular business process languages (CPN, BPMN, YAWL and UML AD) to answer the following research question: *"How do business process modelling languages support CABPs?"*

Based on the standing point of CABP modelling, a framework for comparing modelling notations in terms of supporting context adaptation is proposed. This framework is constructed by a comprehensive list of requirements towards the modelling of CABP. This list of requirements is the result of both repetitive review of relevant literature and eleven CABP scenarios from empirical studies. By applying this framework, the selected business languages are examined and their modelling support towards context adaptation is compared against these requirements.

The rest of the paper is structured as follow. Section 2 handles the related work. Section 3 details the proposed framework. Evaluation of business notations regarding their fulfilment of the framework in terms of used scenarios is elaborated upon in Section 4. Finally, Section 5 delivers the conclusions and discusses future work.

2 Background and Related work

A business process cannot adapt to the changing situations without being aware of the situational context in which it is executed. Many studies, such as [6–11], make contributions to define context in order to formulate a more precise definition of CABP. Despite all these efforts, there is no uniquely adopted definition so far. This is because each of these definitions is formulated from different perspectives or based on different process design approaches. In this paper, we refer to the definition provided by [3, 12–14]. The context towards context-aware business process is therefore defined as: *"Context is any information that is relevant to and might affect the execution of a business process. This information includes aspects of the process itself, the business environment in which it is embedded as well as any other entities that interact with the process."*

A context-adaptive business process, on the other hand, is defined as: "*A business process with the ability to recognize relevant contexts, integrate the relevant knowledge and information into the execution."*

In terms of constructing context-aware business processes, [3] examines the role-based modelling approach where assignment of relationships between roles and functions can be changed according to the dynamic context, given that roles and functions are more static. [15] proposes an extended CPN tool with ontology-based context models to support context adaptation. [16] explains how Worklet, an extended tool of YAWL, supports dynamic workflow and exception handling. In terms of extended tools of BPMN, [17] introduces uBPMN to support ubiquitous processes while [18] proposes to integrate BPMN with Internet of Things (IoT) devices and its native software components to support IoT dynamism.

Some analyses provide a comparative analysis between the most commonly used traditional modelling notations, such as [10, 19–25]. Yet, none of them concern context-adaptive BPs. They aim at normal control flow capability or specific applications like Web of Things (WoT) [1, 26]. Though the targets and perspectives of these comparative analyses are different from the purpose of this paper, their research sheds light on the methodology in terms of properties and requirements for comparison criteria.

3 Framework

To obtain the essential requirements that will compose the framework, a literature survey is performed where roughly 50 relevant papers have been analysed. Firstly, these papers are categorized based on their relevant aspects, including definition of context/context-awareness/context adaptation, context-aware or context-adaptive scenarios, CABP modelling, and CABP modelling with extended languages and tools. The second step is to identify scenarios related to CABP and extract all potential requirements. This step iterates until the most fitting scenarios are found, and a comprehensive list of CABP requirements is identified.

3.1. Relevant scenarios

From the literature survey, eleven relevant scenarios are extracted based on their representativeness. These scenarios are listed and briefly described below; for a full description of the scenarios, source papers are presented.

- **Scenario 1 - Ambient Assisted Living (AAL) system** [27]: The AAL system consists of three parts: the Set Top Box (STB), the remote computer, and the smart phone. Each enclose an individual process and interact with the other in terms of exchanging data and information, yet executed independently by different actors. The STB process makes use of the sensors on the patients to detect their location. It interacts with the smart bio-medical devices on patient to collect the patients' health measurement data when the patient is at home. When it is detected "not at home", STB requests the smartphone to take-over and perform the monitoring and reporting of the health data. All the health data is eventually sent to the remote computer process in the hospital for analysis. In case of abnormal data, (e.g. high or low blood pressure), on-site doctors are notified and the corresponding actions take place.
- **Scenario 2 - E-business shop process** [19]: The entire online shopping process is executed by two resources, customer and the shop, all interacting. Depending on the customer's action, activities from the shop are triggered and executed. This requires flexible sequence order among the activities of the shop process.
- **Scenario 3 - Dynamic pricing process in supermarket** [18]: Price of orchid is automatically reduced if the temperature rises as it affects the quality. The process includes a temperature sensor which estimates the quality of orchid and Electronic Shelf Label (ESL) which indicates the price.
- **Scenario 4 - Online banking system for international transfer** [28]: The online banking for international transfer process is performed via an internet

banking application. It starts with a user logging into the online banking system; the user location is automatically detected by the device-installed GPS sensor at the same time. A feasibility check-up is then activated based on the detected location. If the check-up returns true, the user must manually input the destination country, which again is checked for validity by the system. In the case of an invalid country, an exception arises, and the process flow is channelled to display error message. Otherwise, the system will determine the corresponding transaction fee and exchange rate based on the predefined business rules and situational information, respectively.

- S**cenario 5 - Vehicle repair process** [19]: Vehicle repair process is triggered by the reception of a vehicle, and completed after handing over the vehicle back to the customer. Within this process, the vehicle does not receive any repair until a diagnosis is confirmed. Furthermore, the repair types are determined according to the diagnosis.
- **Scenario 6 - Post-production phase of film** [16]: The post production process can be broken into three major stages, pre-edit, edit and post-edit. The following tasks are identified across the three stages: prepare film/tape for edit, video effect production (VFX), offline task, edit sound and music, online task, film finishing task, finishing tape/disk task, and release printing. In addition, in the pre-edit stage, post-edit stage, and the high-resolution editing, more than one task is involved. Execution of involved tasks is required to be flexible according to the context or to be skipped.
- **Scenario 7 - Product promotion process in department store** [8]: The product promotion process normally starts with sales assistant being notified with the promotion event by the manager, and ends with closing a sales deal. To increase the chances of successful sales and be more efficient, it is essential to identify potential buyers rather than to approach every customer. The potential buyers' identification is context sensitive, and involves large number of deterministic context information and facts, which are necessarily integrated into the process.
- **Scenario 8 - Health checking process** [19]: The health checking process starts with the admission of the patient, and ends by discharging the patient. Within this process, preconditions are held for the execution of some examination activities, such as MRT test, puncture, etc. This gives rise to different variants of the process, which require adaptation of the initial process.
- **Scenario 9 - Process of handling incoming call in insurance company** [10]: The process of handling incoming calls for insurance claims is impacted by the environmental elements. Different approaches are required depending on the weather factors to ensure smooth workflow, for example, hiring external staff during storm season.
- **Scenario 10 - Air ticket reservation and check-in process** [10] : The process specifies the foreseen exceptions and the corresponding strategies during ticket reservation and check-in process in an airline company. For example, weather condition or holiday season may lead to more traditional check-ins and special arrangement is required.
- **Scenario 11 - Environmental monitoring system** [29]: Numerous sensors are installed for measuring various environmental index. Once there is a deviation, alert will be sent to the central system to trigger adaptation of the overall process.

3.2. Identified Requirements

From the literature and the scenarios, we have extracted a set of requirements that are classified into three main categories: support for context modelling, support for flexibility, and support for scalability. The reason for such classification is that each category plays an indispensable role in supporting the construction of CABP as is next detailed. Table 1 lists the three categories of the identified requirements along with their sub-requirements. Among the sub-requirements, support for creating sub-processes and support for decision logic can be classified into both scalability and flexibility as they apply in both cases. All three categories and the sub-requirements are explained in detail in what follows. Furthermore, among the eleven scenarios, scenario 1, 4, 6 and 7 are selected to evaluate the support of the languages towards the requirements for CABP. The selection is based on the fact that these scenarios demand together all the specified requirement. This is also demonstrated in Table 1.

Requirements	Scenarios			
	1	**4**	**6**	**7**
Support for Context Modelling				
Support for interoperability [11]	X			X
Support for use of external context [30, 31]		X		
Support for context fusion [31, 32]	X	X		
Support for time constraint [33]	X			
Support for Flexibility				
Support for sub-processes [8, 10, 34]	X	X	X	X
Support for flexible activity execution [33]			X	
Support for decision logic [8]		X	X	X
Support for routing logic [10, 19, 36]	X	X	X	
Support for exception handling [16]	X	X		
Support for Scalability				
Support for Sub-processes [36]	X	X	X	X
Support for Separation of Concerns [37]		X	X	X

Table 1: CABP requirement framework & Selected scenarios with involved respective requirements

3.2.1. Support for Context Modelling

The ability to represent and manage the context data is crucial for dynamic processes. A dynamic environment generates massive context data at any point in time that needs to be expressed in the model. These context data may affect the execution of relevant activities in a process and its overall outcome. The following four sub-requirements are identified to support context modelling.

Support for interoperability. Interoperability means heterogeneous systems involved in one business process can interact with each other in real-time mode [11]. More precisely, when activities or processes within a system or among different systems can make use of data flow between different applications or users, this process or system is interoperable.

Support for use of external context. In the domain of IoT, hardware sensors and virtual sensors are usually used to capture the external contexts by which process activities are triggered. Nevertheless, regardless of the domain, there are other sources that can provide usable contextual information [30]. Irrespective of the ways the external context is acquired, a modelling language is expected to support real-time data retrieval from the physical world to enable dynamic business processes.

Support for context fusion. Context fusion refers to the aggregation of data being collected from multiple sources prior to being propagated for use. Especially in cases where data from single sources may not carry sufficient information to understand the full situation [32]. Therefore, it is necessary to fuse data from multiple sources to achieve more accurate information [31]. However, data may be acquired in different forms, such as data captured by sensors and contextual data like business rules and relevant regulations. Depending on the data forms, different fusion methods apply.

Support for time constraint. Within a workflow, the result or execution of one activity may influence others. Such influence may closely be linked with time-related aspects like activity duration, time constraints between activities (such as deadline, interval, temporality [33]), and their feasibility (i.e. time constraints do not contradict each other). Time constraints play a crucial role in designing and managing processes. It ensures the consistent state of workflow instances. If it is violated, an exception error must arise during the execution. In addition, in case of redesigning an existing business process, the information provided by time constraints about the actual time consumption of activity execution is indispensable to ensure business process improvement [38].

3.2.2. Support for Flexibility

Process flexibility is essential to deal with variations caused by a changing environment, and occurring during run time A flexible business process has the ability to alter itself, or the parts affected by the context, at runtime, to support execution alternatives that deal with foreseen and unforeseen changes [10, 39]. The following five sub-requirements are identified to support flexibility.

Support for creating sub-processes. A business process usually aims at achieving one ultimate hard-goal. Use of sub-processes allows to decompose a process into several soft adjustable sub-goals, represented by their respective sub-processes. In this way, a better understanding of the relevant contexts is ensured by linking them to each sub-goal, and corresponding context variables are modelled into the sub-processes. Eventually, the chance of failing to achieve the ultimate hard-goal is reduced by mitigating the risk of being affected by context changes via sub-processes. Moreover, sub-processes enable the modelling of situations without detailing the relevant context information in the master process. Whenever a situation arises, the relevant sub-process is activated to adapt to the context changes, while the master process remains unaffected [34].

Support for flexible activity execution. A dynamic business process allows to determine the execution sequence of activities at run time [33]. This implies that predefined activity sequences may weaken flexibility. Therefore, a sequential process modelling language is not expected to profoundly support the situation that certain activities are required to be executed in arbitrary orders, multiple times or even be skipped.

Support for decision logic. Decisions are essential and unavoidable in business processes. It is deterministic in the path of process execution. Hence, modelling the decision logic, including decision points, conditions and branching paths, is crucial. Moreover, there are two types of decision points, data-driven and event-driven. The former decides on the branch it takes based on the data output from the precedent activity, while the latter makes use of data from an external event. Regardless of the types, context-adaptive modelling languages are expected to provide corresponding constructs to model decision logic.

Support for routing logic. Routing logic is a representation of "workaround" in a process. It refers to the use of gateways to model the variability of a business process, hence, execution patterns of the process vary depending on the available input data or the relevant context [19].

Support for exception handling. Executions of processes may deviate due to the exceptions that greatly differ from the norm. According to [16], it is important to provide a comprehensive mechanism to define exceptional process behaviour and proper ways of handling that behaviour to achieve a complete and adequate CABP. With such mechanism, exceptions raised from changing contexts are handled without calling off the entire process or modifying the design of the whole specification. However, handlers are only specified for foreseen exceptions [40]. Unforeseen exceptions are considered as a black box, i.e. they are *non-existent* until their appearance. In this case, a flexible business process is expected to deal with these black box exceptions to ensure and maintain a proper process execution. From the modelling perspective, in addition to exception handlers, additional constructs or means are required to incorporate the support for exception handling.

3.2.3. Support for Scalability

Scalability of a process refers to its capability to expand and accommodate with growing workload due to added resources [41]. Integration of contexts into processes makes them grow rapidly. Thus, scalability is required to support the rapid growth of the models. Scalability can be split in the following two sub-requirements.

Support for creating sub-processes. Sub-processes are used to support process scalability by detailing complex activities. By using sub-processes, a simple business process can be extended to higher degree of complexity, while the main process remains abstract and clean. Sub-processes are also more reusable due to the fact that they can cover a more limited functionality, can be separated from the parent process, and can be triggered when a certain condition is satisfied or called from other processes. [29] points out that decomposition or decentralization of processes helps to increase the scalability and performance of business processes.

Support for Separation of Concerns. Complex BPs designing involves various concerns apart from major workflow. This is due to the ever-increasing process complexity and dynamic changes. Separation of these concerns decomposes the complex process such that to avoid crosscut influence of different concerns during execution. On the other hand, the process must be scalable to integrate these concerns back into the execution and further expand to include added resources. Extraction of decision logic and data modelling are examples of such concerns, for both are contextual. Hardcoding of both into the core process behaviour description may result in hyper-complex process flow and potential execution errors [37].

4 Process Modelling Notation Evaluation using the Framework

In this section, the proposed framework is applied to evaluate to what extent BPMN, CPN, UML AD, and YAWL provide support for modelling CABPs. As such, process models of each of the four selected scenarios are constructed using each modelling notation. Figure 1 demonstrates the process model of AAL system in BPMN.

Considering that business people are the targeted end-users in the design phase, the evaluation of business notations values more the graphical representation ability, simplicity and understandability of the constructed models.

4.1 Support for Context Modelling

Support for Interoperability. YAWL supports interoperability via task decomposition, while it is limited to internal data sharing at task level. **BPMN** 2.0 supports the interoperability by using lanes and pools representing both internal and external resources. Data sharing and information exchange happen via message flows, message events, data objects and data storages. As in Figure 1, the message sent by task "*send request for patient data*" in the STB process is received as a trigger of the smartphone process. Similar example in the remote computer system, which starts with a message start event, triggered by the reception of patient's health data. A similar mechanism as BPMN is applied in **UML AD**, where partitions are acting as lanes, accept/sent actions and data pins represent data and message flows. **CPN** does not support interoperability due to the lack of constructs.

Support for Use of External context. Except for **BPMN** 2.0, which makes use of lanes and data objects & storages to support the use of external context, none of the other three notations support this sub-requirement. In Figure 1, the three parts of the AAL system correspond to the three resources, which are modelled as different roles participating in the process execution, represented by different pools. This also makes the process interoperable. In addition, data objects and storages are used to enable plugin of externally stored patient health data for executions of analysis tasks.

Support for Context Fusion. The fusion of contexts can be simply modelled using parallel join structure, namely, AND join which is supported by all four languages. Moreover, **YAWL** supports the fusion of context also through task decomposition. However, the limited-to-internal-data character as in interoperability applies. **BPMN** 2.0 also makes use of business rule task to support the fusion of more complicated contextual data, such as business rules. In the AAL process, the measurement of health condition is performed by a set of activities being executed simultaneously, which are enabled by using AND split. The results of these activities are synchronized by an AND join (Figure 1).

Support for Time Constraints. Time constraints are supported by all the four languages, yet to different degrees. **YAWL** editor offers a task timer element for constructing time constraint, while it is only applicable to time-out and delay constraints on atomic manual tasks and automated tasks, respectively. **BPMN** 2.0 offers the options to model various types of time constraints through timer events, including time-out, delay, interval, and duration. Figure 1 shows different types of BPMN timers used to model time interval of ten minutes between every measurement of the health condition, as well as a one-minute time constraint before reporting the patient's abnormal health data. Moreover, a timer start event is used to trigger the event-based sub-process “patient data management”, which runs every four hours. **UML AD** deploys the *hourglass* symbol to represent time events. By applying time stamps on tokens and a global clock for the whole net, **CPN** can support all types of time constraint.

4.2 Support for Flexibility and Scalability

Support for Creating sub-processes. YAWL creates sub-processes by using composite tasks, there is no distinction between types or functions of sub-processes. In addition to support for creating simple standard sub-processes, **BPMN** 2.0 extends its support for different types of sub-processes, such as compensation, loop, multiple instance and event-based sub-processes. In Figure 1, “*Patient Data Management*” is an independent process from the main control flow, and it is modelled using an event-based sub-process, which is triggered when the condition is satisfied. In **UML AD s**ub-processes are created using call behaviour action and activities. Hierarchies are used in **CPN** to decompose processes into modules to represent different levels of a CPN process.

Support for Flexible Activity Execution. To enable flexible activity execution, **YAWL** and **BPMN** use OR gateways as a solution. However, this leads to complexity and difficulty of maintenance. **BPMN** 2.0 makes use of ad-hoc sub-process to replace OR gateways and thus to avoid process errors caused by the latter [43]. However, standard **YAWL** does not provide any alternatives. In **UML AD**, there are two ways to enable flexible activity execution, either via interruptible activity region or by formulating OR gateway functions with XOR and AND gateways which results in complex process models. **CPN** does not directly support this requirement, but partial and indirect support is found through the combination of parallel and decision patterns – represented by places and transitions.

Support for Decision Logic. Simple decisions are perfectly supported by the four languages in forms of XOR-split in **YAWL** and **BPMN**, decision nodes in **UML** and formulated XOR functionality using places and transitions in **CPN**. Figure 1 shows the use of XOR-split in BPMN representing data-driven decision on which process to perform the monitoring based on the detected location. Regarding more complicated decisions which involve business rules, BPMN provides the mechanism of fusing contexts via business rule tasks. The aggregation is done in form of a decision table, which is then applied to execute complex decisions. While the definition of decision tables is not supported by BPMN 2.0 directly, compatible support is provided through the Decision Model and Notation (DMN) standard [44, 45]. In **UML AD**, for complex decisions, Object Constraint Language (OCL) is used to formulate pre- or post- conditions, constraining the possible actions of activities.

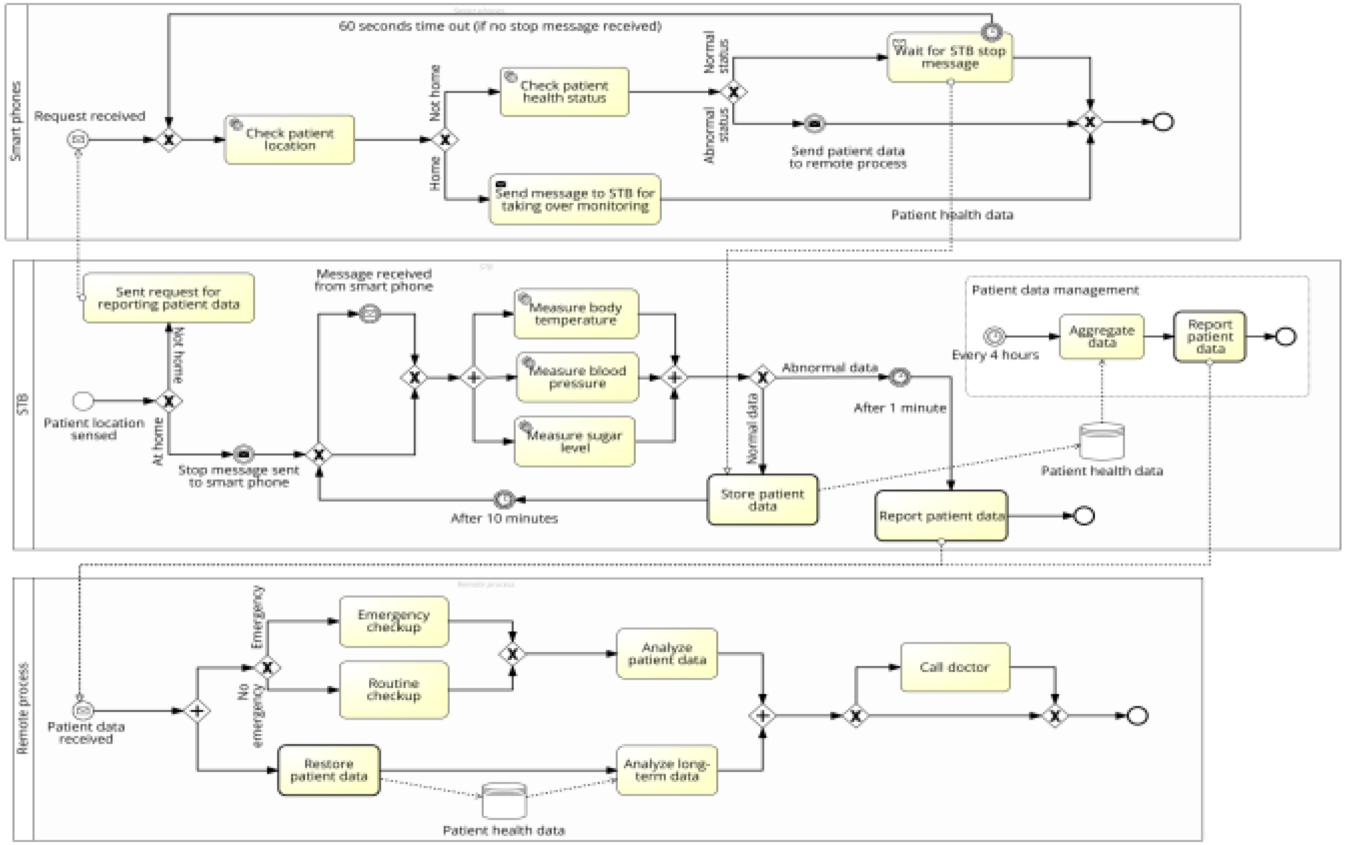

Figure 1: Process model of AAL system in BPMN

Category	Sub-requirement	YAWL	BPMN 2.0	UML AD	CPN
Support for Context Modelling	**Support for Interoperability**	• Task decomposition (internal)	• Message flow • Message events • Lanes & pools • data object & storage	• Accept event & send signal action • Swimlanes • Object node & input/output pins	-
	Support for Use of External context	-	• Lanes • Data object & storage	-	-
	Support for Context Fusion	• AND join • Task decomposition (internal)	• AND join • Business rule task	• Fork and join node	• Parallel activities structure
	Support for Time Constraint	• Task timer	• Timer events	• Timer event action	• Timestamp & Global clock
Support for Flexibility	**Support for Creating Sub-Processes**	• Composite task	• Sub-process marker	• Call behaviour action • Activity	• Hierarchy
	Support for flexible activity execution	• OR gateway	• OR gateway • Ad-hoc sub-process • Events	• Interruptible activity region • Events	• Combination of places and transitions
	Support for Decision Logic	• XOR split • Condition	• XOR split • Business rule task	• Merge and decision node • OCL<localPrecondition> & <localPostcondition>	• Decision Structure
	Support for Routing Logic	• Task decoration	• Gateways • Events	• Merge and decision node • Fork and join node • Events	• Parallel activities structure • Decision structure
	Support for Exception Handling - Foreseen	• Cancellation set	• Intermediate boundary events	• Exception handler • Interruptible activity region • Action events	• Extra transitions and places with CPN ML
Support for Scalability	**Support for Creating Sub-processes**	• Composite task	• Sub-process marker	• Call behaviour action • Activity	• Hierarchy
	Support for Separation of concerns	• Data separation vis XML standards	• Decision separation via business rule task	• Decision separation via OCL	-

Table 2: **Evaluation of process languages on context-adaptiveness**

Support for Routing Logic. In general, there are three types of routing logics, AND, XOR and OR. **YAWL** and **BPMN** both support the three types of routing behaviours. Moreover, **BPMN** provides additional gateway types with specified functionalities, like event-based gateways and complex gateways. For **UML AD**, routing logic falls into two types, the usage of either fork and join nodes or merge and decision nodes. Either type, they refer to AND and XOR routing logic. For OR structures, the formulation applies (see 4.2 Support for Flexible Activity Execution). There are no specific constructs to directly model either of the three routing behaviours in **CPN**. However, places and transitions can be used to form corresponding routing patterns.

Support for Exception Handling. YAWL handles foreseen exceptions by implementing a cancellation set. Yet, it only covers simple exceptions like time-out of a manual task and error. The intermediate boundary events in **BPMN** 2.0 not only expand the types of possible exceptions during the execution, but also give options of handling types, interrupting or non-interrupting. Exception handlers and interruptible regions are used to provide the support for exception handling in **UML AD**. Use of extra task pattern and CPN ML language, provide **CPN** with partial support for exception handling as well. However, limitations to model exception cases in **CPN** make the language the least favourable in this requirement. Noteworthy is that in terms of support for unforeseen exceptions, none of the four languages is competent.

Support for Separation of Concerns. UML AD and BMPN support the separation of concerns regarding the decision aspect. However, only YAWL supports data separation, as it deploys the XML-based standards for data management that can be linked to the processes.

5 Conclusion and Future work

This paper is motivated by the growing prevalence of business process management in a rapid changing environment and its high failure rate. It focuses on the research question *"How do business process modelling languages support context-adaptive business processes?"* To answer this question, we conducted an intensive CABP related literature review, which lays a solid basis for the construction of the proposed framework. The framework consists of three categories, including in total nine sub-requirements, which are applied to evaluate the support of business process notations towards modelling context-aware processes. Moreover, this framework is evaluated through the application of different approaches in real-world scenarios using four commonly used standard business languages. An evaluation matrix is generated based on the framework to compare the support of BPMN, CPN, UML AD, and YAWL during the modelling of scenario processes. It is found that BPMN 2.0 fulfils most of the requirements among the four languages due to the wide range of elements and constructs it provides and thus it appears to be the most suitable of the four considered languages for modelling context-adaptive BP during the design phase. However, none of the other notations provide full support for modelling CABPs.

For future work, it is suggested to put more focus on the topic of context modelling in BPM since it is considered as the most important requirement in dynamic process modelling. Numerous real case studies are appearing especially due to the emergence of IoT applications. Furthermore, on top of the traditional languages, the limitations of languages towards modelling context adaptation give rise to the study of their extensions and tools, therefore, the notation extensions and the supporting tools are also worth exploring. Finally, it is worthwhile to include a more detailed empirical comparison to examine how different business languages facilitate process modelling in terms of context-adaptability.

References

1. Lu, R., Sadiq, S.: A survey of comparative business process modeling approaches. Proceedings of the 10th International Conference on Business Information Systems (BIS2007). 82–94 (2007).
2. Dumas, M., La Rosa, M., Mendling, J., Reijers, H. a.: Fundamentals of Business Process Management. (2013).
3. Saidani, O., Nurcan, S.: Context-awareness for adequate business process modelling. In: Proceedings of the 2009 3rd International Conference on Research Challenges in Information Science, RCIS 2009. pp. 177–186 (2009).
4. Brocke, J. vom, Schmiedel, T., Zelt, S.: Considering Context in Business Process Management- The BPM Process Framework. BPTrends. (2015).
5. vom Brocke, J., Schmiedel, T., Recker, J., Trkman, P., Mertens, W., Viaene, S.: Ten principles of good business process management. Business Process Management Journal. 20, 530–548 (2014).
6. Dey, A.K.: Understanding and Using Context. Journal of Personal Ubiquitous Computing. 5, 4–7 (2001).
7. Coutaz, B.Y.J., Crowley, J.L., Dobson, S., Garlan, D.: Context key. 48, 49–53 (2005).
8. de la Vara, J.L., Ali, R., Dalpiaz, F., Sanchez, J., Giorgini, P.: Business Process Contextualisation via Context Analysis. Proceedings of the 29th International Conference on Conceptual Modeling (ER 2010). 6412, 471–476 (2010).
9. Rosemann, M., Recker, J., Flender, C., Ansell, P.: Understanding Context-Awareness in Business Process Design. Proceedings 17th Australasian Conference on Information Systems. 2006, (2006).
10. Rosemann, M., Recker, J., Flender, C.: Contextualization of Business Processes. International Journal of Business Process Integration and Management. 3, 47–60 (2008).
11. Li, S., Xu, L. Da, Zhao, S.: The internet of things: a survey. Information Systems Frontiers. 17, 243–259 (2015).
12. Henricksen, K., Indulska, J., Rakotonirainy, A.: Modeling Context Information in Pervasive Computing Systems. 167–180 (2002).
13. Saidani, O., Nurcan, S.: Towards Context Aware Business Process Modelling. 8th Workshop on Business Process Modeling, Development, and Support. 9 (2007).
14. Kröschel, I.: On the Notion of Context for Business Process Use. In: ISSS/BPSC. pp. 288–297 (2010).
15. Serral, E., De Smedt, J., Vanthienen, J.: Extending CPN tools with ontologies to support the management of context-adaptive business processes. Lecture

Notes in Business Information Processing. 202, 198–209 (2015).

16. Adams, M.., Ter Hofstede, A.., Russell, N.., van der Aalst, W.. B.: Dynamic and context-aware process adaptation. Handbook of Research on Complex Dynamic Process Management: Techniques for Adaptability in Turbulent Environments. 104–136 (2009).
17. Yousfi, A., Bauer, C., Saidi, R., Dey, A.K.: UBPMN: A BPMN extension for modeling ubiquitous business processes. Information and Software Technology. 74, 55–68 (2016).
18. Meyer, S., Ruppen, A., Magerkurth, C.: Internet of things-aware process modeling: Integrating IoT devices as business process resources. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 84–98 (2013).
19. Ayora, C., Torres, V., Pelechano, V.: BP Variability Case Studies Development using different Modeling Approaches. 1–31 (2011).
20. BÖrger, E.: Approaches to modeling business processes: A critical analysis of BPMN, workflow patterns and YAWL. Software and Systems Modeling. 11, 305–318 (2012).
21. Jian, H.Y., Shi, X.S., Wen, S., Li, J.W.: Formal semantics of BPMN process models using YAWL. Proceedings - 2008 2nd International Symposium on Intelligent Information Technology Application, IITA 2008. 2, 70–74 (2008).
22. Lerner, B.S., Christov, S., Member, S., Osterweil, L.J., Bendraou, R., Kannengiesser, U., Wise, A.: Exception handling patterns for process modelling. 36, 162–184 (2010).
23. Ramadan, M., Elmongui;, H.;, Hassan, R.: BPMN Formalisation using Coloured Petri Nets. Proceedings of the 2nd GSTF Annual International Conference on Software Engineering & Applications. (2011).
24. Geambasu, C.V.: BPMN vs. UML activity diagram for business process modeling. Accounting and Management Information Systems. 11, 637–651 (2012).
25. Vasko, M., Dustdar, S.: A view based analysis of workflow modeling languages. Proceedings - 14th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2006. 2006, 293–300 (2006).
26. Meyer, S., Sperner, K., Magerkurth, C., Pasquier, J.: Towards Modeling Real-World Aware Business Processes. Proceedings of the Second International Workshop on Web of Things. 1–6 (2011).
27. Dar, K., Taherkordi, A., Baraki, H., Eliassen, F., Geihs, K.: A resource oriented integration architecture for the Internet of Things: A business process perspective. Pervasive and Mobile Computing. 20, 145–159 (2015).
28. Ann, M., Kerryann, M.: Context-aware Process Design: Exploring the Extrinsic Drivers for Process Flexibility. 26, 423–436 (2005).
29. Haller, S., Magerkurth, C.: The real-time enterprise: Iot-enabled business processes. IETF IAB Workshop on Interconnecting Smart Objects with the Internet. 1–3 (2011).
30. Alves, P., Ferreira, P.: Distributed Context-Aware Systems. (2011).
31. Serral, E., De Smedt, J., Snoeck, M., Vanthienen, J.: Context-adaptive Petri nets: Supporting adaptation for the execution context. Expert Systems with

Applications. 42, 9307–9317 (2015).

32. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context Aware Computing for The Internet of Things. IEEE Communications Surveys & Tutorials. 16, 414–454 (2014).

33. Olegas Vasilecas, Toma Rusinaite, D.K.: Dynamic Business Processes and Their Simulation: A Survey. Databases and Information Systems IX. (2016).

34. Smanchat, S., Ling, S., Indrawan, M.: A survey on context-aware workflow adaptations. Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia - MoMM '08. 414 (2008).

35. Rosemann, M., Recker, J., Flender, C.: Contextualisation of business processes. International Journal of Business Process Integration and Management. 3, 47 (2008).

36. Hermann, E., Ávila, R.B., Navaux, P.O.A., Denneulin, Y.: Lecture Notes in Computer Science. (2007).

37. Vanthienen, J., Caron, F., De Smedt, J.: Business rules, decisions and processes: five reflections upon living apart together. Proceedings SIGBPS Workshop on Business Processes and Services (BPS'13). 76–81 (2013).

38. Eder, J., Panagos, E., Rabinovich, M.: Time Constraints in Workflow Systems. Advanced Information Systems Engineering, CAiSE '99. 286–300 (1999).

39. Schonenberg, H., Mans, R., Russell, N., Mulyar, N., van der Alst, W.: Process flexibility: A survey of contemporary approaches. Lecture Notes in Business Information Processing. 10, 16–30 (2008).

40. Russell, N., Van Der Aalst, W., Ter Hofstede, A.: Exception Handling Patterns in Process- Aware Information Systems. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 4001 LNCS, 288–302 (2006).

41. Nielsen, C., Lund, M.: The Concept of Business Model Scalability. Social Science Research Network. 1–20 (2015).

42. Sehic, S., Li, F., Nastic, S., Dustdar, S.: A programming model for context-aware applications in large-scale pervasive systems. In: International Conference on Wireless and Mobile Computing, Networking and Communications. pp. 142–149 (2012).

43. Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Seven process modeling guidelines (7PMG). Information and Software Technology. 52, 127–136 (2010).

44. OMG: Decision Model and Notation (DMN) 1.1. (2016).

45. Hasić, F., Devadder, L., Dochez, M., Hanot, J., De Smedt, J., Vanthienen, J.: Challenges in refactoring processes to include decision modelling. In: Business Process Management Workshops (2017).