Comparing Business Process Ontologies for Task Monitoring

Amina Annane, Mouna Kamel and Nathalie Aussenac-Gilles
IRIT - Paul Sabatier University, Toulouse, France

Keywords: Business Process Monitoring, Business Process Modeling, Ontology, Industry 4.0.

Abstract: Business process (BP) modelling is an active area of research due to its multiple applications. For systems that support/monitor operators to perform their tasks (i.e., tasks of a given BP), a formal representation is essential. Various BP ontologies are available to formally represent BP. In this paper, we review and compare a set of nine BP ontologies according to their ability to represent process specification and process execution in a fine-grained way to enable task monitoring. The comparison shows that, on the one hand, ontologies developed from scratch establish a clear distinction between process specification and process execution, but do not allow to represent workflow constraints required for process execution. On the other hand, most of the ontologies, that are ontological versions of existing BP modeling languages, focus only on process specifications but do not represent process execution, or mix the representation of BP specification and execution.

1 INTRODUCTION

A business process (BP) is a set of related tasks that have to be performed in a specific order, by some organization units and actors, to reach well defined business goals (Dumas et al., 2013). To maintain a competitive position, industrial companies look permanently for improving their BPs. A research area has emerged from this need, known as Business Process Modeling (BPM) (van der Aalst, 2013). It aims at providing models and tools to analyze, improve, simulate and automate business processes (BPs). *Business Process Models* provide a more precise definition of BPs by representing the enterprise activities and work-flows at an abstract level, including the description of the roles and responsibilities of its organization units for different tasks as well as task dependencies and used resources (goods, data and knowledge). In order to better evaluate BPs, and to assist their execution and management, information technologies require formal representations of BP models. More recently, the Industry 4.0 vision assumes an effective and high-quality communication between systems (interoperability), which also relies on a formal representation of BPs (Vogel-Heuser and Hess, 2016). In this context, ontologies, defined as formal representation of domain knowledge at a conceptual level, are a good means to provide standard and formal representations, and by this way, to reinforce interoperability and communication between systems (humans and machines) that use these BP models. Moreover, ontology formal languages allow to perform automatic reasoning on the knowledge that they represent. This capability contributes to check knowledge integrity and consistency (Rospocher et al., 2014). A deeper discussion about the benefits of representing BPs using ontologies is available in (Lautenbacher et al., 2008).

Beyond BP description, another perspective can be to check BP execution and to perform reasoning on this execution, for instance to diagnose failures and to capture recovery procedures. This requires additional concepts that can keep track of each process run. In this paper, we examine the needs to be satisfied by an ontology that would support the execution of BPs and keep track of their executions.

By definition, ontologies should be shared and easily reusable models that facilitate knowledge modelling and to increase model interoperability (R. Gruber, 1995). So we examine the literature in order to find reusable ontologies that could contribute to define a relevant BP ontology that would support the representation of both process specifications and process executions. We propose a review of nine of the most popular BP ontologies, and we evaluate their ability to explicitly represent BPs' specification and their execution. The goal of this analysis is to guide us in selecting the most relevant ontologies among the existing ones.

This work is motivated by our participation in the

Annane, A., Kamel, M. and Aussenac-Gilles, N.
Comparing Business Process Ontologies for Task Monitoring.
DOI: 10.5220/0008978706340643
In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020) - Volume 2*, pages 634-643
ISBN: 978-989-758-395-7; ISSN: 2184-433X

AVIREX project that fits in Industry 4.0. AVIREX partners aim at developing an *intelligent* virtual assistant that should: (i) assist/monitor operators in the step-by-step execution of BPs; (ii) answer the operator's questions about the process execution; (iii) keep all related details to process execution, included the context in which an anomaly (i.e., unexpected event) may occur (so as to support the diagnosis process); (iv) save the solution if the operator succeeds in solving the anomaly. To accomplish its mission, the virtual assistant relies on a formal and a fine-grained representation of the whole BP, hence the need for an ontology that enables such a representation. Furthermore, the ontology should be generic to be specified and instantiated with the BPs of any company. In the two use cases given by the project industrial partner, the BPs describe how to assemble electronic, digital and physical components.

The following of the paper is organized as such: Section 2 identifies key concepts that should be covered by an ontology to represent BP execution and specification; they will serve as comparison criteria of the nine studied ontologies sketched in Section 3. From this comparison and requirements, we selected several modelling options and discuss them in Section 4 before we conclude in Section 5.

2 REQUIREMENTS

We studied several knowledge sources: (i) research papers and technical documents, and (ii) competency questions, to identify the key concepts that should be covered by a core BP ontology.

2.1 Knowledge Sources

Research Papers and Technical Documents: We studied a set of research paper that we cite and discuss throughout this article. Moreover, we analyzed two corpora provided by AVIREX industrial partners with the help of their experts. One corpus includes 20 documents describing real BPs, and the other one contains 28 documents describing anomalies (or feedback experiences) with/without their solutions.

We are interested in internal and executable BPs of industrial companies. They are imperative processes to be performed by human agents (i.e., operators). The role of the virtual assistant is to monitor and support operators to improve their efficiency and effectiveness when performing these processes.

Figure 1 shows a sub-process (made anonymous for confidentiality reasons) to give an idea of what a BP looks like. As we can observe, the instructions ask the operator to perform a set of specific activities ("execute", "verify", "note"), on a specific place ("MMA station"), in a sequential order. Moreover, some activities are controlled by conditions ("if the execution has generated an alarm"), and require some resources ("the software SOFT").

```
On the MMA station:
1.  Execute the script MUXF.exe
2.  Using the software SOFT, verify
whether the script execution has
generated an alarm
3.  If the execution have generated an
alarm, note the error code
```

Figure 1: Excerpt of a real Business Process.

Competency Questions: Competency questions are recognized to be a good means to identify ontology specifications (Grüninger and Fox, 1995). Thanks to our meetings with experts and related works (Hepp and Roman, 2007), (Abdalla et al., 2014), we collected a set of 26 competency questions: "What are the sub-activities of a given process?", "Which activities must precede a given activity?", "Which resources are required for/produced by a given task?", "Who can perform/has performed a given activity?", "What is the context (i.e., task\agent\resource\..) of a given anomaly?", etc.

2.2 Key Concepts

Based on the studied knowledge sources, we identified the following key concepts:

Process Specification refers to the model that describes the process and how it should be performed. It includes the following items:

a. *Activity Decomposition*: Processes are complex procedures that are usually decomposed into atomic activities and complex activities (i.e., sub-processes).

b. *Workflow*: It specifies the order (sequential or parallel) in which the activities should be performed.

c. *Preconditions/Postconditions* (PP): Beginning or ending some activities may be subject to conditions. Preconditions control the beginning of activities, while Postconditions control their ending. Postconditions usually are the preconditions of the next activities. PP depend on the availability of (i) data or (ii) resources (agents, machines, raw materials, etc.), (iii) the execution of other activities, (iv) time constraints, or (v) the occurrence of events. An Event is something that "happens" during process execution e.g., a call phone or the arrival of an e-mail.

d. *Place*: It specifies where an activity should be performed: manufacturing facilities and the resources located at each of them.

e. *Anomaly*: Unexpected values or/and errors may occur during process execution, giving rise to an *anomaly* to be tracked in the ontology. At the specification level, expected anomalies are represented with their repair activities.

Process Execution is about representing the actual execution of a given process specification (or process model), the activities actually performed, who (i.e., actual agent) performed them, and in which *place*. It also represents the *events* that triggered the execution of activities, and the actual *resources* involved, consumed or produced during this execution. Human agents may perform some activities that are not specified in the process model and which it is essential to keep a trace of. A process specification may correspond to several executions. Process/Activity execution is mainly characterized by different *states* (e.g., ready, completed, aborted, ...) and *time intervals* (i.e., execution-start-time and execution-end-time). To help experts in diagnosing anomalies, every piece of information related to the anomaly occurrence context should be saved. Moreover, if the operator succeeds in solving it, the solution should also be saved for re-use when the same anomaly appears again.

Organizational Model and Resources define the hierarchy of professional roles and their relations, in other words, it defines who is subordinated by whom in the organization. This is important in validation workflows for example. Since both activity specification and execution refer to resources, it is essential to represent the different types of *organizational resources* held by the enterprise, and that are involved in BPs.

3 STUDIED BP ONTOLOGIES

In this section, we briefly introduce a set of nine ontologies that deal with the semantic representation of BPs. Here, we try to give the reader a clear idea about the origin, the context of development, and the representation of the key concepts. If some key concepts (see Section 2) are not discussed for a given ontology, it means that they are not covered by this ontology. This is to avoid redundancy, and keep the overview of each ontology short.

To select relevant ontologies, we searched the *Google Scholar* database with various combinations of the key words: ”ontology”, ”business process”, ”formal model”, ”formal representation”. The search results were checked through a title and abstract screening to identify relevant works. Only publications with a number of citations higher than 20 (see Tab. 1), and that explicitly mention the development of an ontology for representing BPs were considered as relevant. We used the number of citations to assess the popularity of ontologies. The search in the database was aborted when we noticed a significant amount of repetitions and loss of precision. In addition, we performed a cross reference search. An ontology is mainly composed of a set of concepts (i.e., class of things), a set of relations among these concepts, and a set of axioms that are used to formalize domain knowledge. For the sake of readability, we use the SMALL CAPITAL LETTERS font when we refer to the actual concepts of an ontology.

Table 1: Number of citations of the studied ontologies according to google scholar.

Ontology	Citations
EO	1353
PSL	210
COBRA	113
SMPO	104
IMAMO	23
Petri-Net ontology	109
EPC ontology	69
BPMN 2.0 ontology	70
BPMO ontology	65

Among the studied ontologies, we distinguish (i) ontologies developed from scratch, and (ii) ontologies that formalize BPM languages.

3.1 Ontologies Developed from Scratch

The Enterprise Ontology (EO). The enterprise ontology (Uschold et al., 1998) was developed within the Enterprise Project that aimed at designing a modeling framework that uses executable process models to help users perform their tasks. EO is composed of five sub-ontologies: (1) Meta-ontology and Time, (2) Activities and processes, (3) Organization, (4) Strategy, and (5) Marketing. In this paper, we consider only the three first sub-ontologies since they are related to the key concepts that we have identified.

The sub-ontology of activities and processes is concerned with the representation of process specification and execution; the central concepts are ACTIVITY SPECIFICATION and ACTIVITY.

Process specification is limited to the representation of the process decomposition, but the workflow is not represented: neither concepts, nor relationships are provided to represent the order in which the activities should be performed.

Organization sub-ontology is concerned with the representation of agents (human or software), and their organizational hierarchy.

Organization sub-ontology is linked to the activity sub-ontology via CAPABILITIES and AUTHORITIES concepts. On one hand, EO represents the capabilities and authorities of each agent. On the other hand, it represents the capabilities and authorities required for performing each activity.

An ACTIVITY is an execution of an ACTIVITY SPECIFICATION. It is performed by a DOER in a given TIME INTERVAL. In EO, an ACTIVITY may use/consume RESOURCES, be decomposed into SUB-ACTIVITIES, have PRECONDITIONS and EFFECTS (i.e., outcomes). EVENTS are considered as a specific type of ACTIVITY.

EO is formalized in Ontolingua which is an old formal language and is available at https://tinyurl.com/y26su33t.

Process Specification Language Ontology (PSL). PSL (Gruninger and Menzel, 2003) is an ISO standard for representing manufacturing processes (Pouchard et al., 2005) that was created at the National Institute of Standards and Technology (NIST[1]). It was designed to facilitate correct and complete exchange of process information among manufacturing systems. ACTIVITY concept denotes a specification of an activity. ACTIVITY-OCCURRENCE denotes the actual execution of one ACTIVITY. Except the decomposition of an activity into sub-activities, the representation of process specifications is made by applying constraints on the instances of ACTIVITY-OCCURRENCE (Grüninger, 2009). Indeed, using PSL, it is not possible to define process specifications (e.g., order constraints, events, activity preconditions/postconditions) independently of process execution (i.e., activity occurrences).

PSL offers a set of concepts and relations to represent for a given activity-occurrence (i.e., at the execution level) : sub-activity occurrences, beginning and ending time, states, preconditions and postconditions, and the resources involved in its execution. There is no elaborated representation of anomalies, but the concept ATTEMPT (i.e., an activity occurrence that has been interrupted for some reason) may be a good starting point for adding an extension.

PSL is formalized in OWL and is available at https://tinyurl.com/y4annulm. In (Grüninger, 2009), the authors explain how to represent process specifications using PSL with first order logic sentences.

Core Ontology for Business pRocess Analysis (COBRA). The SUPER (Semantics Utilized for Process management within and between EnteRprises) project[2] is a European research project aimed at defining a set of ontologies for Semantic Business Process Management (SBPM). In the context of this project, Hepp et al. (Hepp and Roman, 2007) have proposed eight ontologies (e.g., Upper-level Process Ontology (UPO)) and clarified their scopes with competency questions. Unfortunately, we were not able to access the SUPER ontologies except for COBRA and BPMO. COBRA (Pedrinaci et al., 2008) is a core ontology that aimed at enhancing BP analysis. The core concepts of COBRA are BUSINESS ACTIVITY and BUSINESS ACTIVITY REALIZATION that represent process specification and process execution, respectively. COBRA represents the composition of activities at the specification and execution levels. Furthermore, it reuses the time ontology[3] to represent the beginning and ending time-points of BUSINESS ACTIVITY REALIZATIONS. A taxonomy of monitoring events has been defined to capture the different events related to process execution. These events change process or activity states (e.g., started, aborted, completed, etc.). However, at the specification level, events are not represented. COBRA allows to specify AGENTS or ROLES responsible for the execution of each BUSINESS ACTIVITY, as well as the actual AGENT that has performed a given BUSINESS ACTIVITY REALIZATION. Different relations are provided by COBRA to represent the resource/data requirements and produced of a given BUSINESS ACTIVITY or BUSINESS ACTIVITY REALIZATION such as *uses*, *consumes*, *produces* and *provides*. These relations link activities to PERSISTENT ENTITY that may be a PHYSICAL ENTITY or a NON-PHYSICAL ENTITY. The categorization of these entities is inline with top level ontologies such as DOLCE (Masolo et al., 2003). COBRA has been formalized in OCML but it has no accessible link.

Software Maintenance Project Ontologies (SMPO). To decrease the efforts and costs of the software maintenance projects, an extended software engineering environment MANTIS has been implemented including the semi-formal ontology (SMPO) (Ruiz et al., 2003). This ontology deals with the representation of the static and dynamic aspects of software maintenance projects. SMPO is composed of three sub-ontologies: (1) Maintenance, (2) Workflow, and (3) Measurement. We do not discuss the measurement ontology since it represents metrics for assessing the quality of products and processes, which is out the scope of this paper.

Maintenance sub-ontology (i.e., the static aspect of the maintenance process) in turn includes four sub-ontologies describing: (i) PRODUCT and their composition; (ii) AGENTs, their ROLEs and their orga-

[1] https://www.nist.gov/

[2] https://cordis.europa.eu/project/rcn/105285/factsheet/en

[3] https://www.w3.org/TR/owl-time/

nizational model; (iii) ACTIVITY, its sub-types (i.e., MAIN ACTIVITY, SUPPORT ACTIVITY, etc.), its inputs/outputs, and the used RESOURCEs; and (iv) PROCESS ORGANIZATION allows to represent procedures (activity specification) that may be used by activities. The content of these procedures is not represented. A maintenance Activity is triggered by a MAINTENANCE REQUEST (i.e., an internal document that describes the anomaly) communicated by an AGENT. Then, the maintenance request is manually studied by experts that produce an INVESTIGATION REPORT that points out the CAUSE of the anomaly. The corrective actions are included in the maintenance request report, but are not represented.

Workflow sub-ontology (i.e., the dynamic aspect) deals with the representation of activity decomposition, temporal constraints between activities and their execution order. It is surprising to observe that specifications such as temporal constraints between activities are represented at the execution level and not at the specification level, in the process organization sub-ontology. SMPO does not establish a clear distinction between process specification and process execution. The same concept ACTIVITY is used to represent both of them. Indeed, ACTIVITY is defined as *the description of the work to be performed*. At the same time, it comprises the STATE attribute that refers to execution states (e.g., not started, in execution, etc.). SMPO has been semi-formalized in UML.

Industrial MAintenance Management Ontology (IMAMO). IMAMO (Karray et al., 2012) has been developed within the scope of the SMAC (Semantic MAintenance and lifeCycle) project in collaboration between academic and industrial organizations from France and Switzerland. The SMAC project aimed at providing a semantic platform of industrial maintenance ensuring the capitalization and reuse of knowledge. Hence, it was essential to represent maintenance process and its related concepts.

Process specification is represented mainly with the PROCESS PATTERN concept that is composed of a set of STEPs. A process workflow is described through several concepts (e.g., STEP, TRANSITION, etc.) and relations (e.g., *has first step*, *next step*, *next transition*, etc.). For more details, see the processes view in (Karray et al., 2012).

RESOURCEs are well covered in IMAMO since they are the central element in an industrial maintenance process, especially material resources. Indeed, many details about material resources are represented such as their composition, their functional requirements, etc. In addition, IMAMO provides a rich taxonomy of industrial resources. However, for a given activity specification (i.e., PROCESS PATTERN or STEP), the required resources are not represented except agents who should perform the activity.

Key concepts for process execution are PROCESS and ACTIVITY. They are characterized by the attribute "State", related to a PERIOD (start and end dates), and an ACTOR who performs the activity. A PROCESS refers to a PROCESS PATTERN and composed of a set of ACTIVITY. An ACTIVITY is the execution either an ACTION or a STEP. ACTION denotes a task specification that aims at resolving a TROUBLE. TROUBLE concept refers to the Anomaly concept. IMAMO represents different types of anomalies, their causes, and the actions that may be performed to resolve them. Only the events triggering anomalies are represented.

IMAMO has been formalized in OWl and is accessible at https://tinyurl.com/yyxjsryp

3.2 Ontologies Developed from BP Modeling Languages

A *process modeling language* provides appropriate syntax and semantics to precisely specify BP requirements, in order to support automated process verification, validation, simulation and process automation (Lu and Sadiq, 2007). To take advantage of an ontological representation of BPs, formal ontological descriptions of some BP languages have been proposed:

Petri-Net Ontology. In (Koschmider and Oberweis, 2005), the authors have proposed an ontology that formalizes Petri-Net elements for BP representation. Petri-Net is a mathematical and graphical modeling language that is utilized for modelling workflows and simulating/analyzing their enactments. A Petri-Net is a directed graph that mainly consists of two different nodes, PLACES and TRANSITIONS. PLACES represent states, while TRANSITIONS represent events and activities. The abstract concept *"token"* is used to simulate the move of the execution flow through the process graph. It is possible to represent the different workflow patterns (e.g., AND-Split, OR-Split, Loops, etc.) (van der Aalst et al., 2003) with Petri-Net. Time constraints, data and resource requirements are not supported naively, but they may be represented with additional or customized tokens (e.g., colored tokens) (van der Aalst, 2015). Although this solution can be handled by computers, it requires a high expertise and generates complex Petri-Net graphs. Petri-Net ontology does not offer more elements, hence it offers the same representation characteristics as Petri-Net. Moreover, petri-Net ontology does not offer a vocabulary for the domain of BP, which is a key element for a BP ontology. This is explained by the fact that BP representation is just an application for Petri-Nets. Its strength is its mathematical model that has clear execution semantics. In addition, several process analysis algorithms exist (e.g. detecting deadlocks, reachability, etc.) and may be reused. The Petri-Net Ontology has been formalized in OWL, but we were not able to find its OWL file.

Event-driven Process Chain (EPC) Ontology. EPC has been developed in a joint effort between SAP[4] and the Institute of Information Systems of Saarbrücken in the context of the Architecture of Integrated Information Systems (ARIS) framework (Keller et al., 1992). EPC is a graphical BP modeling language; its major strength is its simplicity and easy-to-understand notation. Originally, EPC does not include a formal definition, but several works have proposed formalized ontologies with or without extension over time (van der Aalst, 1999),(Thomas and Fellmann, 2007). EPC has been developed to represent enterprise workflows. It has five key concepts (Scheer et al., 2005): (i) FUNCTION that represents an activity specification, (ii) EVENT to represent preconditions and post-conditions of FUNCTION, (iii) CONTROL-FLOW that refers to a transition from one EPC element to another, (iv) LOGICAL CONNECTOR, such as OR, AND, and XOR, that connects at least three FUNCTIONs, and finally (v) RESOURCE. A LOGICAL CONNECTOR can either join or split FUNCTIONs. EPC does not support process execution. EPC is formalized in OWL but it is not available.

Business Process Model and Notation (BPMN) Ontology. BPMN is a standard[5] widely adopted by companies that has been developed by the Object Management Group (OMG)[6]. BPMN has two main versions BPMN 1.0 and BPMN 2.0 (OMG, 2011). The first version has been published in 2008 while the second one in 2011. An ontology has been developed from each version in (Rospocher et al., 2014) and (Natschläger, 2011), respectively. In the following, we consider only the BPMN 2.0 version since it is the most recent and complete one. Indeed, it offers an execution logic for its elements and a mapping to the BP execution language (BPEL[7])

BPMN meta-model offers a fine grained representation of process specification. A PROCESS is represented as a container that includes four types of elements (or FLOW ELEMENTS): (i) ACTIVITY: an activity specification that may be atomic (TASK) or complex (SUB-PROCESS), (ii) EVENT: something that happens, (iii) GATEWAY: a flow control for synchronization, and (iv) SEQUENCE FLOW: transitions between the previous elements to complete the representation of the workflow. Transitions may be controlled with conditions. BPMN supports almost all workflow patterns (e.g., Simple Merge, Exclusive choice, loop) (van der Aalst et al., 2003), and represents advanced BP modeling paradigms like exception handling, transactions, and compensation. In addition, it allows to represent data requirements (i.e., DATA INPUT and DATA OUTPUT), and the agents or roles that are responsible of a given activity. However, it is not possible to define organizational or resource models with BPMN.

BPMN offers specific events to deal with anomalies such as the ERROR EVENT for interrupting errors and ESCALATION EVENT for non interrupting errors. In both cases, BPMN allows to represent the process to be performed when these events occur.

Process execution is represented only through a set of additional attributes for instances of process-specification concepts such as *State* (e.g., ”ready”, ”completed”, ”aborted”, etc.) for PROCESS, ACTUALOWNER – the actual actor that is performing the activity – for ACTIVITY, etc. Thus, no actual distinction is made between process specification and process execution. Nevertheless, BPMN defines the life-cycle model of activity instances, with the events enabling transitions between them (OMG, 2011) (pp.428). To the best of our knowledge, the OWL version of BPMN2.0 (Natschläger, 2011) is not available to the community. However, BPMN meta-model is available in BPMN specification (a document with more than 500 pages) as UML class diagrams (OMG, 2011). An XML version of this meta-model is also available. However, BPMN specification includes a number of constraints (or specifications) in natural language, that are represented neither in the UML diagrams, nor in the XML file.

Business Process Modeling Ontology (BPMO). BPMO was developed in the context of the SUPER project to represent highlevel BP workflow models, abstracting from existing BP notations and languages (Cabral et al., 2009). BPMO supports only the representation of process specification. A PROCESS has a WORKFLOW composed of WORKFLOW ELEMENTS: TASKs, EVENTs, BLOCK PATTERNS inspired from BPEL, and GRAPH PATTERNS inspired from BPMN 1.0. BLOCK PATTERNS and GRAPH PATTERNS are both control flows representing workflow decision points (van der Aalst et al., 2003). Block-structured control flows are defined similar

[4]Systems, Applications & Products in Data Processing (SAP): a German multinational software corporation that makes enterprise software to manage business operations and customer relations.

[5]https://www.iso.org/standard/62652.html

[6]https://www.omg.org/about/index.htm

[7]https://tinyurl.com/3xntz3

to existing programming languages by using block-structures such as ”if” or ”while”. Conversely, graph-oriented language defines control flows through explicit links between activities. For more details about the difference between BP graphical and block languages, please refer to (Kopp et al., 2009). BPMO has no graphical language of its own. BPMO is formalized in WSML-Flight which is an old formal language. We did not find an accessible link for BPMO.

Table 2 sums up the previous descriptions where the symbols +, $-$, $\approx$ have the following meanings: (i) $+$: the ontology supports the element representation, (ii) $-$: the ontology does not support the element representation, and (iii) $\approx$: the key concept is partially represented e.g., $\approx$ (agent) means among resources only agents are represented. Furthermore, for anomaly specification, we used the $\approx$ symbol for Petri-Net, EPC and BPMO because it is possible to represent anomalies as events and activities, but there is no specific concepts/relations for representing them in these models. On the Data values line, $\approx$(as PP) means that data values can be represented as preconditions/ postconditions but not with a specific concept.

4 DISCUSSION

These ontologies and the project requirements guided our modeling choices, in particular for the representation of process specification and execution.

4.1 Process Specification

As we can observe in Tab. 2, among the ontologies developed from scratch, PSL is the one that covers the most BP specification elements. However, as explained earlier, in PSL, the definition of specifications depends on the execution occurrences.

Ontologies obtained from existing BP languages offer a richer representation of process specifications. This may be explained by the maturity of BP languages and their evolution over the years, in particular BPMN, the most recent and complete BP language. Indeed, even if Petri-Net, EPC, BPMO, and BPMN cover almost the same elements, BPMN stands out for its expressiveness. Compared to Petri-Net and EPC, BPMN offers much more concepts and relations to represent the BP domain. For instance, while ”event” is represented in EPC by a single general concept with no formal semantic other than its annotation with a term, BPMN offers a set of 48 concepts representing different event types with specific attributes and clear semantics (e.g., timer event, start event, error event, message event, etc.) that are classified into: (i) catching/throwing events, and (ii) interrupting/non-interrupting events.

4.2 Process Execution

Ontologies developed from scratch (except SMPO) establish a clear distinction between process specification and process execution. As we can see in Tab 2, the time perspective is covered by these ontologies, which allows to keep trace of the different executions. COBRA is distinguished by the fine grained representation of monitoring events, the different states and possible transitions between them within the lifecycle of processes/activities.

Ontologies extracted from BPM languages either do not represent process execution such as EPC and BPMO, or mix between process execution and process specification (i.e, no clear distinction) like Petri-Net and BPMN. Indeed, Petri-Net execution consists on the movement of the token over the net (i.e., the graph representing the process). Hence, the represented activities are considered as activity-specification and activity-execution at the same time. The BPMN developers propose to instantiate the process model than to add some attributes such as *state* to the instances (activity instance, process instance, etc.) without considering the time interval of any instance. Process-specification is an informational entity that describes what and how to do, while process-execution is a temporal entity that has a time interval (start-date; end-date). Thus, we believe it is semantically more correct to differentiate these two concepts.

4.3 Vocabulary

One promising role of ontologies was to provide, for each domain, a shared vocabulary with precise and formal definitions that should improve the communication between different actors of these domains (either humans or machines) (Gruber, 1991). However, we observed a high terminological heterogeneity in the studied ontologies. On the one hand, different terms are used to label the same concepts. For instance, the main concept denoting activity-execution is labeled differently from one ontology to another: (EO: ”Activity”), (PSL: ”Activity occurence”), (COBRA: ”Process instance”), and (IMAMO: ”Process”). Another example is the terms used to label the concept representing the move of the execution flow from one element to another: (IMAMO: ”Transition”), (BPMN: ”Sequence flow”), and (BPMO: ”Control flow connector”), etc. On the other hand, the same terms are used to label non-equivalent concepts e.g., the term ”Process” is used to label activity-

Table 2: BP ontologies vs. key concepts.

Key concepts	Comparison criteria	Ontologies from scratch					Ontologies from BPM languages			
		EO	**PSL**	**COBRA**	**SMPO**	**IMAMO**	**Petri-Net**	**EPC**	**BPMN 2.0**	**BPMO**
Specification	Process decomposition	+	+	+	+	+	+	+	+	+
	workflow	-	≈	-	+	+	+	+	+	+
	PP - Data	-	≈	+	+	-	≈	+	+	+
	PP - Resource	≈(agent)	≈	+	+	≈ (agent)	≈	+	≈ (agent)	-
	PP - Activity dependency	-	≈	-	-	+	-	-	-	-
	PP - Time constraints	-	≈	-	-	-	≈	+	+	+
	PP - Event	-	≈	-	-	-	≈	+	+	+
	Anomaly description	-	-	-	-	-	≈	≈	+	≈
	Anomaly solution	-	-	-	-	-	≈	≈	+	≈
	Place	-	-	-	-	-	-	-	-	-
Execution	Activities	+	+	+	+	+	-	-	≈	-
	Time	+	+	+	+	+	-	-	-	-
	State	-	+	+	+	-	+	-	+	-
	Events	+	+	+	-	≈ (anomaly)	+	-	≈	-
	Data values	≈(as PP)	+	+	+	-	≈	-	≈	-
	Involved resources	+	+	+	+	≈(agent)	≈	-	≈ (agent)	-
	Place	-	-	-	-	-	-	-	-	-
	Anomaly description	-	-	-	+	+	-	-	≈	-
	Anomaly solution	-	-	-	-	+	-	-	≈	-
Organizational resources		-	-	-	-	+	-	-	-	-
Organizational model		+	-	-	+	-	-	-	-	-
Formal language		Ontolingua	OWL	OCML	UML	OWL	OWL	OWL	OWL/UML	WSML
Accessibility of the formal version		+	+	-	-	+	-	-	-/+	-

specification in COBRA, but activity-execution in IMAMO. Such a high terminological heterogeneity reflects the conceptual differences and makes it not feasible to automatically align these ontologies using tools such as AML (Faria et al., 2013). Reusing and combining these ontologies requires to manually study the ontologies one by one in order to establish correspondences between equivalent/related concepts. In the future, it would be interesting to define a standard vocabulary that could be used as a pivot to interconnect existing BP ontologies.

4.4 Synthesis

Based on the elements discussed previously, none of the studied ontologies covers all the requirements of the core ontology that we need, but as we may notice they have common and complementary fragments (see Tab.2). BPMN seems to be the most relevant one for our project. In particular, because of its expressiveness that allows to capture the most of details related to process specification, which is essential in our case since we aim at monitoring process execution step by step. Unfortunately, BPMN ontology is not available. Hence, we will start by translating the BPMN meta-model into an ontological model using (OMG, 2011) and (Natschläger, 2011) .

Despite the richness of BPMN, it has some limitations: (i) It does not represent the organizational model and resources. Furthermore, it does not allow to specify the resource requirements (other than agents) of an activity. (ii) There is no way to represent the place where the activities should be performed (absent in all models). (iii) There is no explicit concept to represent process execution. To overcome these limitations, we decided to reuse fragments from the other ontologies such as the resource taxonomy of IMAMO, relations that represent the resource requirements such as *consumes*, *uses* from COBRA and the organizational model from EO (with the definition of agent authorities and capabilities). Reusing together BPMN and these fragments is harder than expected due to (i) terminological heterogeneity and differences in the formal languages, and (ii) the lack of availability of the reused ontologies.

Furthermore, we will add concepts to represent process execution such as "Activity instance", "Process instance" and "time:interval" (from the Time ontology) as shown in Fig 2. This enrichment is inspired from the COBRA ontology.

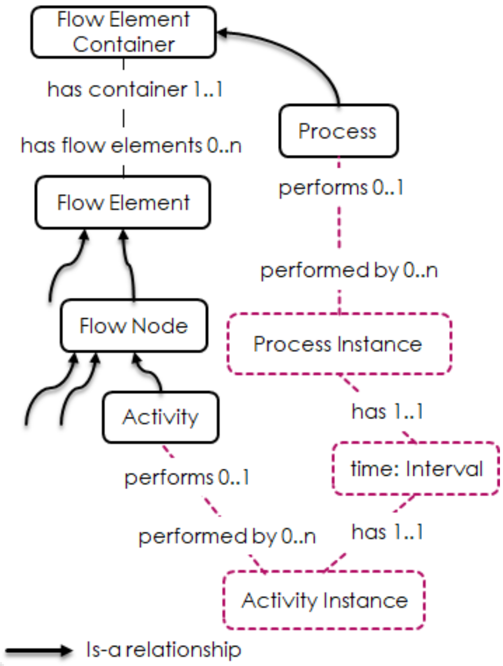

Figure 2: Example of enrichment with new concepts (those with dashed line).

5 CONCLUSION

In this paper, we highlighted the main concepts that should be covered by a BP core ontology, then we presented and compared nine popular business-process ontologies. In particular, we focused on how these ontologies deal with the representation of process specification and process execution.

Regarding our project, BPMN seems the best candidate to start with. However, BPMN alone does not cover all the requirements, which leads us to reuse fragments from the other ontologies.

ACKNOWLEDGEMENTS

This work within the AVI-REX project is funded by a grant from region Occitanie, the FEDER-FSE Midi-Pyrénées and Garonne 2014-2020 program under the label 2017-AVI-REX-IRIT-READYNOV INDUSTRIE DU FUTUR.

REFERENCES

Abdalla, A., Hu, Y., Carral, D., Li, N., and Janowicz, K. (2014). An Ontology Design Pattern for Activity Reasoning. In *5th WS on Ontology and Semantic Web Patterns (WOP)*, pages 78–81, Riva del Garda, Italy.

Cabral, L., Norton, B., and Domingue, J. (2009). The business process modelling ontology. In *4th Int. WS on semantic business process management*, pages 9–16.

Dumas, M., La Rosa, M., Mendling, J., and Reijers, H. A. (2013). *Introduction to Business Process Management*, pages 1–31.

Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I. F., and Couto, F. M. (2013). The agreementmakerlight ontology matching system. In *On the Move to Meaningful Internet Systems OTM, Graz, Austria*, pages 527–541.

Gruber, T. R. (1991). The role of common ontology in achieving sharable, reusable knowledge bases. In *2nd International Conference on Principles of Knowledge Representation and Reasoning (KR). Cambridge, MA, USA*, pages 601–602.

Grüninger, M. (2009). Using the psl ontology. In *Handbook on Ontologies*, pages 423–443. Springer.

Grüninger, M. and Fox, M. S. (1995). The Role of Competency Questions in Enterprise Engineering. In *Benchmarking Theory and practice*, pages 22–31.

Gruninger, M. and Menzel, C. (2003). The process specification language (psl) theory and applications. *AI magazine*, 24(3):63–74.

Hepp, M. and Roman, D. (2007). An ontology framework for semantic business process management. In *8th Internationale Tagung Wirtschaftsinformatik (WI), Karlsruhe, Germany*, pages 423–440.

Karray, M. H., Chebel-Morello, B., and Zerhouni, N. (2012). A formal ontology for industrial maintenance. *Applied ontology*, 7(3):269–310.

Keller, G., Nüttgens, M., and Scheer, A. (1992). Semantische prozeßmodellierung auf der grundlage ”ereignisgesteuerter prozeßketten (epk)”. *Verö ffentlichungen des Instituts für Wirtschaftsinformatik*, (89).

Kopp, O., Martin, D., Wutke, D., and Leymann, F. (2009). The difference between graph-based and block-structured business process modelling languages. *Enterprise Modelling and Information Systems Architectures*, 4(1):3–13.

Koschmider, A. and Oberweis, A. (2005). Ontology based business process description. In *The Open Interop Workshop on Enterprise Modelling and Ontologies for Interoperability*, pages 321–333.

Lautenbacher, F., Bauer, B., and Seitz, C. (2008). Semantic business process modeling - benefits and capability. In *AI Meets Business Rules and Process Management, 2008 AAAI Spring Symposium, Technical Report SS-08-01, Stanford, Cal., USA*, pages 71–76.

Lu, R. and Sadiq, S. (2007). A Survey of Comparative Business Process Modeling Approaches. In *Business Information Systems*, pages 82–94.

Masolo, C., Borgo, S., Gangemini, A., Guarino, N., Oltramari, A., and Schneider, L. (2003). The wonderweb library of fundational ontologies and the DOLCE ontology. *The WonderWeb Library of Fundational Ontologies and the DOLCE ontology. WonderWeb Deliverable D18, Final Report (vr. 1. 0.)*.

Natschläger, C. (2011). Towards a BPMN 2.0 ontology. In *3rd Int. WS on Business Process Modeling Notation*, pages 1–15, Lucerne, Switzerland.

OMG (2011). Business Process Modeling Notation, v2.0 - Specification. Technical report, Object Management Group.

Pedrinaci, C., Domingue, J., and de Medeiros, A. K. A. (2008). A core ontology for business process analysis. In *5th European Semantic Web Conference, ESWC, Tenerife, Canary Islands, Spain*, pages 49–64.

Pouchard, L. C., Cutting-Decelle, A., Michel, J.-J., and Grüninger, M. (2005). Iso 18629 psl: A standardised language for specifying and exchanging process information. *IFAC Proceedings Volumes*, 38(1):37–45.

R. Gruber, T. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5-6):907–928.

Rospocher, M., Ghidini, C., and Serafini, L. (2014). An ontology for the Business Process Modelling Notation. In *8th International Conference on Formal Ontology in Information Systems (FOIS)*, pages 133–146.

Ruiz, F., Garcia, F., Piattini, M., and Polo, M. (2003). Environment for managing software maintenance projects. In *Advances in software maintenance management: technologies and solutions*, pages 255–291.

Scheer, A.-W., Thomas, O., and Adam, O. (2005). Process modeling using event-driven process chains. *Process-aware information systems*, pages 119–145.

Thomas, O. and Fellmann, M. (2007). Semantic EPC: enhancing process modeling using ontology languages. In *WS on Semantic Business Process and Product Lifecycle Management (SBPM), Innsbruck, Austria*.

Uschold, M., King, M., Moralee, S., and Zorgios, Y. (1998). The enterprise ontology. *The knowledge engineering review*, 13(1):31–89.

van der Aalst, W. M. (1999). Formalization and verification of event-driven process chains. *Information and Software technology*, 41(10):639–650.

van der Aalst, W. M. (2013). Business process management: a comprehensive survey. *ISRN Software Engineering*, 2013:1–37.

van der Aalst, W. M. P. (2015). Business process management as the “killer app” for petri nets. *Software & Systems Modeling*, 14(2):685–691.

van der Aalst, W. M. P., ter Hofstede, A. H. M., Kiepuszewski, B., and Barros, A. P. (2003). Workflow patterns. *Distributed and Parallel Databases*, 14(1):5–51.

Vogel-Heuser, B. and Hess, D. (2016). Guest Editorial Industry 4.0 Prerequisites and Visions. *IEEE Transactions on Automation Science and Engineering*, 13(2):411–413.