Towards an ontological analysis of BPMN

Emilio M.Sanfilippo[1,2], Stefano Borgo[2], and Claudio Masolo[2]

[1] Institute of Industrial Technologies and Automation, ITIA-CNR, Italy
emilio.sanfilippo@itia.cnr.it
[2] Laboratory for Applied Ontology, ISTC-CNR, Italy

Abstract. BPMN is a de-facto standard with more than 70 commercial tools that currently support its use. However, its the semantic ambiguities jeopardize its implementation. We perform an ontological analysis of important constructs of BPMN like activities and events to highlight their implicit commitments.

Keywords: Ontological analysis, BPMN, State, Event, Activity

1 Introduction

Business process (BP) modelling concerns the analysis and representation of the activities by which companies coordinate their organisation and work, produce goods, interact with each others and customers. The goal is a common conceptual modelling language that can be easily understood to facilitate business communication. The Business Process Model and Notation (BPMN) [9] is one of such BP languages and is a OMG standard with more than 70 commercial tools that currently supports its use[3]. BPMN focuses on graphical constructs and lacks formal semantics. Thus, it presents conceptual ambiguities regarding the interpretation of its metamodel and the supporting software tools are not guarantee to interoperate. We use ontological analysis to dwell into the backbone elements of BPMN like activity and event. The goal is to investigate whether the standard is (possibly implicitly) committed to some (coherent) ontological perspective. The remainder of the paper is organized as follows: Section 2 describes the state of the art about the analysis of BPMN. Section 3 looks at a BPMN process diagram to highlight some problematic issues. Sections 4 gives the ontological analysis of our target notions.

2 State of the art

The literature about BPMN focuses on three types of analysis: (*i*) the *syntactic* analysis, (*ii*) the *behavioral* analysis, and (*iii*) the *ontological* analysis. The syntactic analysis aims at defining the structural constraints that BPMN-models

[3] This work is part of a larger study [11] that will be presented at the International Conference on Formal Ontology in Information Systems (FOIS 2014). In this version we assume some familiarity with BPMN.

must satisfy. [4] presents the BPMNO meta-ontology implemented in OWL [7]. BPMNO allows reasoning with semantically annotated processes and it enriches BPMN with, e.g., temporal information. Similarly, [1] provides a BPMN ontology encoded in the WSML language [2]. The behavioral analysis looks at what can happen during the execution of a well-formed BPMN model like, e.g., the existence of deadlocks or livelocks [3]. This (static) analysis of a process model considers the semantics underlying the schema only for procedural information and is not relevant for our work in this paper. Finally, the ontological analysis focuses on the characterization of the primitives of the language. [8] discusses the OPAL reference framework and characterizes the specific kinds of activities or events present in a given model, but it lacks a characterization of the general difference between BPMN-activities and BPMN-events. [5] uses the ontology ABDESO to distinguishes actions from other events. The authors find quite a few ambiguous and redundant elements, as well as missing concepts in BPMN. Finally, [10] looks at the mutual relationship between BPMN and the Bunge-Wand-Weber (BWW) foundational ontology. The paper highlights some ontological shortcoming in the first release of the standard with respect to ontological completeness, construct overload, construct excess and construct redundancy.

In this paper we are interested in the ontological analysis of BPMN with the aim of clarifying how one can understand the notions of activity and event. We carry out our study in two ways: first by ontologically analyzing the information provided by the BPMN standard, and then by characterizing our findings on these concepts with the DOLCE ontology [6]. We are not proposing an evaluation of BPMN with respect to an ontology; we rather use the ontology to find (possibly implicit) commitments of the standard, identify business related elements that the standard does not address, and to highlight the variety of interpretations that are consistent with these constraints.

3 Activities and Events in BPMN

The BPMN diagram in Fig. 1 represents a process with four participants: *Purchaser*, *Service provider A*, *Service provider B* and *Service provider C.* The process starts when the event type *None* in the *Purchaser* pool happens. This is followed by the execution of task *Request Quotes* which ends with the sending of a message to each service provider participant in the process. Once a service provider receives the message, it starts its own process consisting in sending a quote to the *Purchaser.* After this, the service provider process ends. When the *Purchaser* receives at least two quotes, it executes the *Assess the Quotes* task after which the process ends for the *Purchaser* provided the condition *Sufficient reserve amount?* is satisfied. Otherwise, the process is back to the *Request Quotes* and flows again.

The reason why some parts of the process are marked as activities and others as events is not immediately clear. In the diagram below, messages are exchanged between the process participants by using tasks of type *Message send.* However, in BPMN one could also use a *Message* as throw event (not showed in Figure

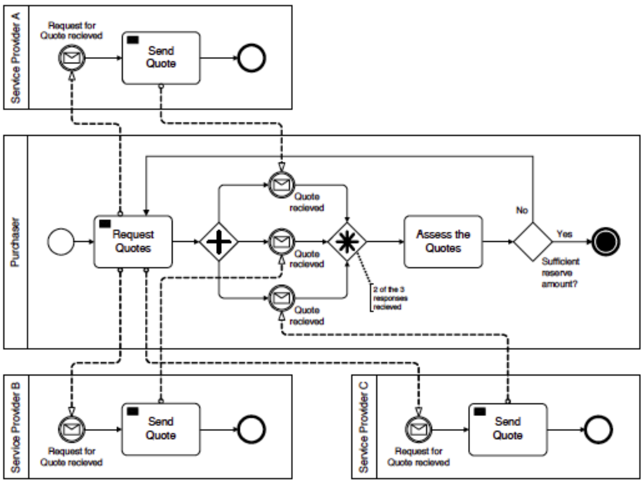

Fig. 1. BPMN process diagram, taken from [9, p.362]

1), which models the sending of a message as well. The meanings of these two different constructs – that seem to model the same thing – is not immediately clear. The BPMN specification [9] provides little help in clarifying the distinction between activity and event. The BPMN Quick Guide[4] states that "[...] an *Event* maps to a time point on a time line [while] a *Task* [activity] maps to a time interval". This seems to mean that events are instantaneous while activities last in time, which implies that temporal atomicity is a discriminating property. Nevertheless, BPMN does not commit to a theory of temporal points or intervals, thus every reference to time remains vague. Another possibility is to understand activities and (at least some) events in terms of endogenous vs exogenous entities: the first are happenings controlled within the poll the latter are out of the control of the pool.

4 Ontological analysis of BPMN events and activities

Events and activities in BPMN are connected to other events and activities in the same (in a different) pool by solid (dashed) arrows; these arrows mark execution precedence and thus temporal dependences. This reveals the temporal nature of events and activities in BPMN. From the ontological viewpoint, we can classify them as (some type of) perdurants or occurrents, in the sense of entities that extend in time possibly by accumulating temporal parts. In the following we use the DOLCE taxonomy of perdurants—mainly the concepts of *Achievement*, *Accomplishment*, *State*, and *Event*—to discuss and ontologically characterize the difference between *Activities* and *Events*, and between *Catch-* and *Throw-events*.

[4] http://www.bpmn.org

While we find helpful to use a foundational ontology like DOLCE, we remark that the analysis could be based on other ontological systems provided they include a taxonomy of perdurants.

Activities and tasks: We have seen that BPMN activities are not instantaneous, thus they take time to execute. In addition, BPMN distinguishes between two types of activities: *tasks*, i.e., atomic activities and *sub-processes*, i.e., non-atomic activities. The relationship between *being instantaneous* and *being atomic* is not trivial given that a task can have a positive temporal extension.

In some ontological theory [12] it is assumed that perdurants extend in time by having different temporal slices at different times. This would rule out BPMN tasks because, by extending in time, they *necessarily* have (temporal) proper parts, where 'necessity' is here used in the ontological sense, namely tasks have temporal parts in all possible worlds. According to this perspective, a task like *Request Quotes* is necessarily *anti-atomic* and *anti-homeomeric*, i.e., all its instances have parts that do not belong to *Request Quotes.* The anti-homeomericity is evident for BPMN sub-processes, whose structure is explicitly included in the BPMN-model.However, it might be suggested that BPMN models the world at some *granularity*, in the sense that tasks are *considered to be atomic* in the context of the model even though they have temporal parts in the actual world. In this case, tasks could be *conceived as* atomic or non-atomic depending on the context, granularity, or perspective on the world. We talk in this case of conceptual modality, because the ontological status of tasks relies not on how they are, but rather on how they are conceived within a certain conceptual framework.

We observe also that the mereological sum of two instances of a task like *Send quote* is not an entity of the same type. This is consistent with the assumption that BPMN activities represent units of work with the purpose of achieving given business goals; they thus *culminate* with the achievement of the goal.

We can then conclude that by considering a strict ontological modality activities are anti-atomic and anti-cumulative, i.e., they can be mapped to DOLCE *accomplishments.* Vice versa, by assuming a conceptual modality, only sub-processes may be mapped to accomplishments. More generally tasks would be mapped to DOLCE *events*, i.e., as anti-cumulative types of perdurants with no commitment on atomicity and homeomericity.

Catch events: We saw that events are instantaneous, consequently they are temporally atomic, that is, they cannot extend over or throughout a temporal stretch. *Catch* events like the reception of a message, are in general exogenous, i.e., their happening is outside of the control of the pool they belong to or, at least, of the branch of the pool process at stake. In this perspective *None* start events could be understood as 'the system is turned-on'. In addition, being culminating perdurants the catch events are anti-cumulative. Anti-cumulativity and atomicity characterize the subcategory of achievements in DOLCE.

The process of *Service Provider A* in Figure 1 cannot proceed unless a trigger is received, i.e., a message is received. Accordingly, if the system of this service provider is 'turned-off', the message will never be received. Thus, behind a catch event there is the assumption that the process is waiting to be triggered, i.e., the system is on a receiving mode.

Differently from activities, these kinds of perdurants (e.g., *waiting*) are *homeomeric*—i.e., the temporal slices of waiting-instances (if any) are themselves waiting instances—and *cumulative*—i.e., the mereological sum of two (immediately consecutive) waiting-instances is still a waiting-instance. Homeomeric and cumulative perdurants are called *states* in DOLCE. For example, Figure 1 indicates that *Service providers A, B* and *C* are (by default) in a waiting status for receiving messages. Thus, a catch event identifies (perhaps implicitly) a state and it further indicates that the pool is committed towards a specific trigger to occur.

However, on the same line of [10], one could consider catch events as state-transitions. For example, the reception of messages can be understood by referring to two states: the state of 'waiting for the message' and the state of 'message received', where the latter is the pre-condition for executing the successive task. The trigger thus enacts a state transition and, in turn, the starting of the new state enables the process to perform its subsequent tasks[5].In the case of *None* catch (start, intermediate) the trigger that is holding the process is not specified. From our viewpoint, there are at least two possible views regarding the semantics of this modelling construct. It might be a placeholder for the initial temporal boundary of the process that in our example corresponds, as a logical constraint, to 'there are no parts of the process that precede the *Request Quote* task'. In this case, the *None* catch bears no further ontological commitment. On the other side, one can return to the idea of a (hidden) waiting state. The latter case seems to be incompatible with the interpretation of the start event as 'the system is turned-on'.

Throw events: Similarly to catch events, throw events are instantaneous, then temporally atomic, and anti-cumulative, i.e., in DOLCE they are classified as achievements. Differently from *catch* events, throw events tend to be endogenous: actions under the control of the pool they belong to. Note that differently from tasks, which can be conceived as structured perdurants, although atomic under a certain granularity, throw events are punctual, thus intrinsically unstructured.

Throw *None end events* can be understood as the achievement of the whole process, and, in a fashion similar to start events, they can be interpreted either as an ontologically neutral placeholder in the model, as a logical constraint, or as an (ontologically committed) achievement. Note that the throw end events marked with a specific trigger icon, like *Message*, *Terminate*, *Signal* and *Error*, indicate an achievement as well but now the culmination point is qualified: the triggers that are specified in these cases (message, signal, termination and error) are amongst the participating entities of the achievement.

[5] The analysis of the causal dependencies among triggers, events and tasks could be very informative.

5 Conclusions

We focused on the ontological analysis of the BPMN notions of activity (task) and event, and classified them within the DOLCE account of perdurant entities. The results are still preliminary and the hope is that it can help to reach a deeper understanding of the system; and to develop sound BPMN-driven ontologies. In the future, we shall expand this first analysis and develop a formalization capturing our results on BPMN.

Acknowledgements: This research is in part supported by the Gecko and Pro2Evo projects of the "Fabbrica del Futuro" (funding: MIUR).

References

1. W. Abramowicz, A. Filipowska, M. Kaczmarek, T. Kaczmarek Semantically enhanced Business Process Modelling Notation. In M. Hepp et al. (Eds.), *Semantic Business Process and Product Lifecycle Management. Proceedings of the Workshop SBPM 2007*, Innsbruck, April 7, 2007
2. J. de Bruijn, D. Fensel, U. Keller, M. Kifer, H. Lausen, R. Krummenacher, A. Polleres, L. Predoiu Web Service Modeling Language (WSML). W3C Member Submission 3 June 2005 Available at: http://www.w3.org/Submission/WSML. Last access March 2014
3. R.M. Dijkman, M. Dumas, and C.Ouyang. Semantics and analysis of business process models in BPMN. *Information and Software Technology*, 50(12):1281–1294, 2008.
4. C. Ghidini, M. Rospocher, L. Serafini A Formalisation of BPMN in Description Logics. Technical Report TR 2008-06-004, FBK-irst, 2008.
5. G. Guizzardi, G. Wagner Can BPMN Be Used for Making Simulation Models? 7th International Workshop on Enterprise & Organizational Modeling and Simulation (EOMAS 2011), together with the 23rd International Conference on Advanced Information System Engineering (CAiSE'11), London, UK.
6. C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari. WonderWeb Deliverable D18. Ontology Library. Available at: http://wonderweb.man.ac.uk/deliverables.shtml, 2002. Accessed March 2014
7. D.L. McGuinnes, F. van Harmelen OWL Web Ontology Language. Overview. W3C Recommendation 10 February 2004. Available at: http://www.w3.org/TR/owl-features. Last access March 2014
8. M. Missikoff, M. Proietti, and F. Smith. Linking ontologies to business process schemas. Technical Report 10-20, Istituto di Analisi dei Sistemi ed Informatica del CNR, 2010. ISSN: 1128 - 3378.
9. Object Management Group (OMG) Business Process Model and Notation (BPMN). Version 2.0, 2011 Available at: http://www.bpmn.org Last access March 2014
10. J. Recker, M. Indulska, M. Rosemann, P. Green. Do Process Modelling Techniques Get Better? A Comparative Ontological Analysis of BPMN. Australasian Conf. on Information Systems. Sydney, 2005
11. E.M.Sanflippo, S. Borgo and C. Masolo. Events and Activities: Is there an Ontology behind BPMN? In: *Formal Ontology in Information Systems*, IOS Press, 2014
12. T. Sider, *Four-Dimensionalism. An Ontology of Persistence and Time.* Oxford: Clarendon Press, 2001.