Time-BPMN

Denis Gagné
Trisotech
3100 Côte Vertu, B380, Montreal, Canada
dgagne@trisotech.com

André Trudel
Acadia University
Wolfville, Nova Scotia, Canada
Andre.Trudel@acadiau.ca

Abstract—We present Time-BPMN which is an extension to the current version of the Business Process Modeling Notation (BPMN). Time-BPMN captures the temporal perspective of business processes. This extension deals with the various temporal constraints and dependencies that may occur while characterizing real world business processes. Time-BPMN allows for the specification and depiction of temporal constraints and dependencies within a BPMN diagram. It not only extends, but also simplifies certain existing temporal BPMN constructs. With our extension, BPMN becomes expressive enough to account for a very large set of real world business processes that are time dependent.

Keywords-BPMN; temporal perspective; temporal semantics; temporal constraints and dependencies

I. INTRODUCTION

The Business Process Modeling Notation (BPMN) provides organizations with the capability of specifying and depicting their internal business processes using a graphical notation. The latest version of BPMN, version 1.2 adopted in January 2009, aims to be a visual language to communicate business processes in a standard manner [1]. The suitability and expressiveness of BPMN as a process modeling language has been explored from various perspectives [2, 3, 4].

One key perspective when specifying a process model or workflow is time. Time is a key resource as we cannot change the amount of time there is in a day. Furthermore, time optimization is often an effective cost reduction strategy for an organization.

The temporal perspective is a contributing factor to both the definition and the enactment of a process model or workflow specification. When defining a workflow, the temporal perspective allows the modeler to explicitly specify temporal constraints and dependencies to ensure that all temporal requirements of the process are met. At enactment time, the temporal perspective of the workflow specification leads to the ability to precisely schedule a process and its resources.

Surprisingly, the temporal perspective is currently poorly addressed by the various business process standards. Given the importance of time with respect to process definition and enactment, our current research direction is to establish time as a first class citizen in the main business process standards.

In this paper we describe a specific proposal, called Time-BPMN, for extending BPMN with respect to the temporal perspective. In earlier work [4], the temporal perspective of process modeling and the suitability of BPMN to express its generic temporal constructs were explored.

Time-BPMN provides a series of attribute and property extensions to BPMN along with their depiction. These extensions do not conflict with or alter the current specification of BPMN.

In the remainder of this paper we present the description, depiction, and syntactical rules of the various temporal constructs that makeup the Time-BPMN extension. We then present examples which show how Time-BPMN:

1. Simplifies the specification and depiction of temporal constructs that were already possible in BPMN.
2. Allows the specification and depiction of processes that were not possible in BPMN.

We then reflect on further work required to bring time forward as a first class citizen of business process standards, and offer some conclusions.

II. TIME-BPMN OVERVIEW

The temporal perspective of process modeling was defined in [4] by providing a series of generic temporal constructs:

- time points,
- intervals/durations,
- temporal constraints, and
- temporal dependencies.

This characterization, independent of any specific modeling formalism or approach, precisely defined each temporal construct and provided, when possible, a formal temporal account of these constructs based on Allen's interval algebra [6]. Allen's interval algebra is the most popular language for temporal knowledge representation and reasoning within Artificial

Intelligence. It deals with convex intervals (intervals with no gaps) and is easy to grasp for business users.

As per BPMN, we treat time points and durations as TimeDateExpressions which are subtypes of Expression using all the attributes of the Expression element [1, p. 280]. In Time-BPMN, referenced time points and durations are specified in the Time-BPMN attributes, and do not have a graphical depiction.

Many intervals are of interest to the temporal perspective: *Transfer Time*, *Queue Time*, *Wait Time*, *Set Up Time* (also known as *Changeover Time), Processing Time* (also known as *Working Time*), *Validation Time* (also known as *Inspecting Time*), *Rework Time*, and *Downtime*. We do not address these specific intervals in this paper. See [4] for a discussion and logical formalization. In this paper, we concentrate on temporal constraints and dependencies which have visual depictions.

In general, temporal constraints can be specified to control the start or finish time of a process or activity. These temporal constraints can be inflexible (i.e. tied to a specific time point) or flexible (i.e. not tied to a specific time point). The inflexible temporal constraints are:

- Must Start On (MSO) and
- Must Finish On (MFO).

A flexible temporal constraint does not specify a specific time point for a process or an activity, but rather imposes scheduling upper and/or lower bounds. The flexible temporal constraints are:

- As Soon As Possible (ASAP),
- As Late As Possible (ALAP),
- Start No Earlier Than (SNET),
- Finish No Earlier Than (FNET),
- Start No Later Than (SNLT), and
- Finish No Later Than (FNLT).

Flexible temporal constraints work in conjunction with temporal dependencies, which are described below, to make a process or activity occur as soon or as late as the process or activity dependency will allow.

A temporal dependency is a relationship between two processes or activities in which one process or activity depends on the start or finish of another process or activity in order to begin or end. There are four types of temporal dependencies between a predecessor and successor:

- Finish-to-Start (FS),
- Start-to-Start (SS),
- Finish-to-Finish (FF), and
- Start-to-Finish (SF).

Temporal dependencies can be further constrained with delays called lead and lag time. A lag time is normally specified as a duration (e.g. 2 days). Lead time is useful when the successor activity requires a head start. Usually, lead time is specified using a negative lag time (e.g. -1 day).

Time-BPMN visually extends BPMN with the depiction of temporal constraints, and temporal dependency relations for activities (task and sub-processes). Temporal constraints and dependencies between independent processes are not addressed herein as this would imply the control (via scheduling) of processes that are assumed to be independent and thus not centrally controlled (i.e. Time-BPMN addresses process orchestration and not process choreography).

In related work, Silver proposed in his blog [7] a notation to temporally order activities. He noted that BPMN should be "expressive, simple to draw, and semantically unambiguous" [7]. Unfortunately, his proposal includes new flow objects linked by sequence flows, and thus is not a conformant extension to the current BPMN version. Time-BPMN, on the other hand, does not introduce new flow objects to BPMN. Time-BPMN provides activity decorators and association links between them. Along with a series of attribute/property extensions, these extensions do not conflict with BPMN and constitute a valid extension. Furthermore, Time-BPMN has a semantically unambiguous specification and depiction of temporal constraints and dependencies.

III. TEMPORAL CONSTRAINTS

In Time-BPMN, temporal constraints are specified to control the start and/or finish time of an activity (task or sub-process). These temporal constraints can be inflexible (i.e. tied to a specific value) or flexible (i.e. not tied to a specific value). To specify temporal constraints, Time-BPMN extends the standard BPMN notation with *Temporal Constraint Decorators*.

Temporal constraint decorators can be attached to any location on the boundary of a task or sub-process. Start and end temporal constraints are depicted by shapes coherent with start and end events in standard BPMN. At most one start and one end temporal constraint can be attached to any particular task or sub-process. See Figure 1 for the possibilities.

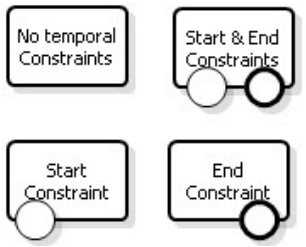

Figure 1. Temporal Constraint Decorators

Start and end temporal constraints cannot be the target or the source of any sequence flow. Markers within the start or end temporal constraints are used to depict particular types of temporal constraints (e.g. ASAP, ALAP, NET, NLT, and ON).

A. Temporal Constraint Markers

1) As Soon as Possible (ASAP): The activity is assigned an As Soon As Possible (ASAP) temporal constraint. The ASAP temporal constraint markers are shown in Figure 2. If the constraint is Start As Soon As Possible (SASAP), the enactment scheduler instantiates the activity as early as it consistently can. No additional time restrictions are put on the activity. This is the default temporal constraint used (or assumed) by most process modelers when specifying workflows. Note that an activity cannot be instantiated prior to the process (Case) being instantiated (i.e., case instantiation time). For a Finish As Soon As Possible (FASAP) constraint, once started the activity must finish as soon as possible. This has the effect of minimizing the duration of the activity. An activity with no temporal constraints specified or depicted is assumed to have an ASAP start and end constraint (see bottom of fig.2).

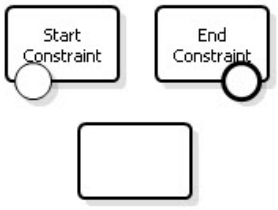

Figure 2. As Soon As Possible (ASAP) Marker

2) As Late as Possible (ALAP): The activity is assigned an As Late As Possible (ALAP) constraint. The ALAP temporal constraint markers are shown in Figure 3. This constraint is analogous to the ASAP temporal constraint, but relative to the desired end of the Case. With a Start As Late As Possible (SALAP) constraint, the activity is scheduled as late as it consistently can, given other scheduling parameters. No additional time restrictions are put on the activity. If activities are scheduled from the desired Case finish-time, the enactment scheduler will determine how late the Case can start and still finish by the specified Case finish-time. If the constraint is Finish As Late As Possible (FALAP), then once started the activity will continue for as long as possible. This has the effect of maximizing the duration of the activity without going beyond the Case finish-time. Since constraints are independent, a FALAP constraint on an activity with no start constraint does not imply SALAP. By default, the start constraint is ASAP.

Note that if a SALAP or FALAP constraint is specified without a Case finish-time, it may be impossible to schedule the activity. We are specifying a notation which may be misused by the user. Inconsistencies may be brought to the user's attention during scheduling. This issue is beyond the scope of this paper.

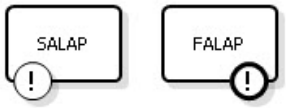

Figure 3. As Late As Possible (ALAP) Marker

3) No Earlier Than (NET): The activity is assigned a No Earlier Than (NET) temporal constraint. The NET constraint markers are shown in Figure 4. This temporal constraint indicates the earliest possible time point that the activity can begin or be completed. In the case of a Start No Earlier Than (SNET) temporal constraint, the activity cannot be scheduled to start any time before the specified time point. In the case of a Finish No Earlier Than (FNET) temporal constraint, the activity cannot be scheduled to finish any time before the specified time point. Recall that the referenced time points are specified in the Time-BPMN attributes and do not have a graphical depiction in Figure 4. Note that a FNET constraint on an activity without a start constraint has a SASAP constraint by default. In this case, the activity starts as soon as possible and does not terminate before the specified time point.

Figure 4. No Earlier Than (NET) Marker

4) No Later Than (NLT): The activity is assigned a No Later Than (NLT) temporal constraint. The NLT temporal constraint markers are shown in Figure 5. This temporal constraint indicates the latest possible time point that the activity can begin or complete. In the case of a Start No Later Than (SNLT) temporal constraint, the activity can be scheduled to start on or before the specified time point. A predecessor activity cannot push a successor activity with an SNLT constraint past the constraint time point. In the case of a Finish No Later Than (FNLT) temporal constraint, the activity must be scheduled to finish on or before

the specified time point. A predecessor activity cannot push a successor activity with an FNLT constraint past the constraint time point.

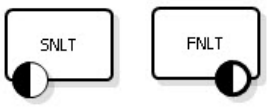

Figure 5. No Later Than (NLT) Marker

5) ON: The activity is assigned an ON temporal constraint. The ON temporal constraint markers are shown in Figure 6. This temporal constraint indicates the exact time at which a process or activity must be scheduled to either begin or complete. In both cases of a Must Start On (MSO) and Must Finish On (MFO) temporal constraint, other scheduling parameters such as temporal dependencies, lead or lag time and delay cannot affect scheduling the process or activity unless this requirement is met.

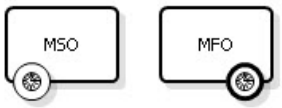

Figure 6. ON Marker

B. Temporal Constraint Attributes

Temporal constraints can be specified for activities (tasks and sub-processes) of a business process diagram. The attributes of a start and end temporal constraint are identical:

- **Id:** Required and unique. For identification.
- **Categories:** Optional. For user-defined semantics. Can be used for purposes such as reporting and analysis.
- **Name:** Optional.
- **Documentation:** Optional.
- **Target:** Required and unique. Target must be an activity (sub-process or task). The temporal constraint is attached to the boundary of the activity and is used to signify a temporal constraint for that activity.
- **ConstraintDetail:** Required. Specifies if the constraint is an ASAP (As Soon As Possible), ALAP (As Late As Possible), NET (Not Earlier Than), NLT (Not Later Than), or ON (Must Be On). The constraint is always relative to some point in time which is specified in the ReferencePoint attribute described below.
- **ReferencePoint:** Required. Can either be CaseStart, CaseEnd, or a specified time point using a TimeDateExpression.

IV. TEMPORAL DEPENDENCIES

In Time-BPMN, temporal dependencies are relationships between two activities (task or sub-process) in which one activity depends on the start or finish of another activity in order to begin or end. Temporal dependency relations are depicted by a directed temporal dependency connection made between a source start or end temporal constraint to a target start or end temporal constraint. The source or target temporal constraint in a temporal dependency relation can be of any temporal constraint type (e.g. ASAP, ALAP, NET, NLT, and ON). A temporal dependency can only connect temporal constraints. It cannot connect any other BPMN elements.

The four types of temporal dependencies in Time-BPMN are:

- Start-to-Finish (SF),
- Start-to-Start (SS),
- Finish-to-Start (FS), and
- Finish-to-Finish (FF) (see figure 7).

Note that as per the current specification of BPMN, a sequence flow has the same temporal semantics as a Finish-to-Start (FS) temporal dependency [4].

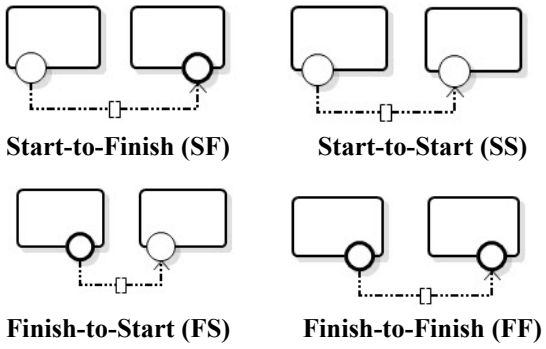

Figure 7. Temporal Dependencies

A. Lead and Lag Constraints

In Time-BPMN, leads or lags are specified by assigning a time interval to a temporal dependency relation. A positive interval value specifies a lag and a negative value specifies a lead. No value or a zero value specifies that there is no lead or lag constraints to the temporal dependency relation. Leads and lags are depicted by putting the interval value in square brackets on the temporal dependency relation. It is not possible to simultaneously specify a lead and lag for a single temporal dependency relation. A lead or a lag also cannot be specified as a time point. The example at the top of Figure 8 is a Start-to-Finish dependency

with a 2 day lag. The same example with a 2 day lead is shown at the bottom of Figure 8.

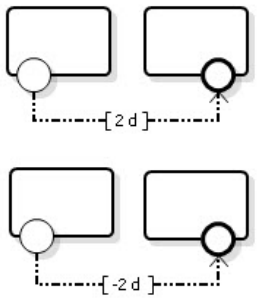

Figure 8. Lead and Lag Constraints

B. Temporal Dependency Attributes

The attributes of a temporal dependency include:

- **Id:** Required and unique. For identification.
- **Name:** Optional.
- **Documentation:** Optional.
- **SourceRef:** Required. Source temporal constraint.
- **TargetRef:** Required. Destination temporal constraint.
- **LeadLag:** Optional. Specifies a lag when a positive value is provided or a lead when a negative value is provided for the dependency.

The SoureRef and TargetRef fields identify the two temporal constraints "connected" by the temporal dependency. Recall that only start temporal constraints and end temporal constraints can be a SourceRef and TargetRef.

V. EXAMPLES

We present examples of processes that can be represented in Time-BPMN but not in BPMN. We also present an example which shows how Time-BPMN simplifies the specification and depiction of existing temporal constructs in BPMN. Note that the diagrams we provide do not represent the only possible specification or depiction of these examples. Process representations in both BPMN and Time-BPMN are not unique.

A. Hunting Example

We present an example where it is not possible to correctly capture the intended temporalities in BPMN. The process description is: the game must be dressed (gutted), cleaned, and bagged within 1 day after returning from a successful hunt. Note how the Finish-to-Finish relationship constrained by a 1 day lag is simply and precisely captured between the two activities Go Hunting and Clean Game in Time-BPMN (see Figure 9).

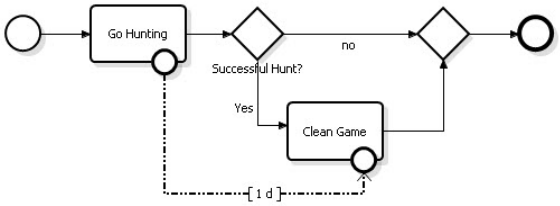

Figure 9. Time-BPMN Diagram of Hunting Example

Alternatively, Figure 10 is an attempt to capture the hunting example in BPMN. Someone familiar with BPMN will notice that the intermediate event on the boundary of the Clean Game task will interrupt and stop the task after 1 day, rather than ensuring that Clean Game is completed. Unfortunately, it is not possible in BPMN to specify a constraint on the end of an activity (task or sub-process) other than the Finish-to-Start default temporal semantic of the sequence flow [4]. Furthermore, in BPMN the concepts of an activity instantiation and its start are considered to be equivalent. Time-BPMN allows one to precisely and concisely specify/depict temporal constraints and temporal dependencies affecting the start and/or end of activities.

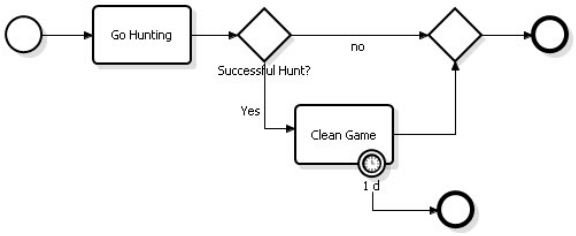

Figure 10. BPMN Diagram of the Hunting Example

One could argue that Figure 11 is the correct BPMN representation of the hunting example. Note that an annotation is used to convey the end constraint of the Cleaning Game activity. Although easily understandable by a business user, we argue that leaving any specification details, in this case temporal constraints, to textual annotation can lead to misinterpretation of the actual meaning. Furthermore, a structured specification is always desirable if automation is the targeted implementation of the model.

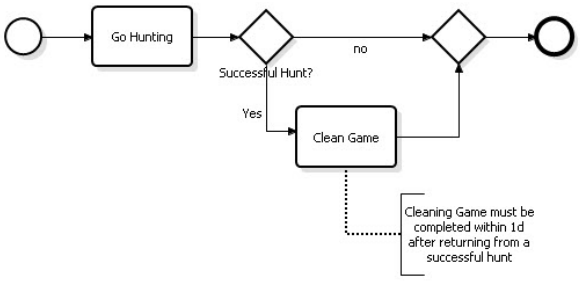

Figure 11. Alternate BPMN Diagram of the Hunting Example

B. Final Exam Example

The exam is to take place on December 1st from 2 to 5 pm. The exam can finish earlier if all students have completed the exam before 5 pm. Tardy students may show up during the first hour of the exam and still be allowed to complete the exam. Signing the exam attendance sheet starts 1 hr after the start of the final exam. Note that this example is intended to model the high level exam process. We are not modeling the process from the student perspective.

1) Simplifying the BPMN representation: Figure 12 is a possible BPMN specification/depiction of this simple process.

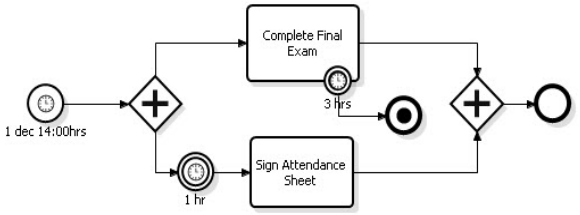

Figure 12. BPMN Diagram of the the Final Exam Example

Using Time-BPMN (see Fig. 13) we can precisely capture the temporal constraints and dependencies intended in our description using a more concise notation.

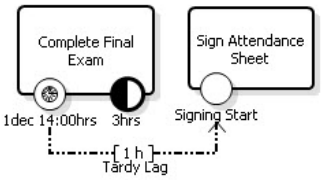

Figure 13. Time-BPMN Diagram of the Final Exam Example

Note that the BPMN semantics does not require a business process diagram to have a start and end event, and further specifies that all activities without an incoming sequence flow receives a token at instantiation. Therefore, the two activities in Figure 13 will receive instantiation tokens but their individual start times will be governed by the temporal constraints.

2) Extending the BPMN representation: We may also want to express the constraint that the signature of the attendance sheet has to be completed prior to the end of the exam. Figure 14 depicts this additional temporal Finish-to-Finish dependency between Sign Attendance Sheet and Complete Final Exam. Note how the direction of temporal dependency connector captures the fact that the end of Sign Attendance Sheet is dependent on the end of the Complete Final Exam and not the other way around. This temporal construct is not possible in BPMN, as there is no direct way to depict the dual temporal dependencies that exist between the two activities.

3) Future Extensions: One could also add the constraint that no student can leave the exam room during the first hour of the exam given that they have to remain to sign the attendance sheet. This means that the Complete Final Exam also has FNET of 1 hr constraint in addition to its already existing FNLT 3hrs constraint. Our current implementation does not allow for either multiple start or end constraints. We are currently working on relaxing this limitation.

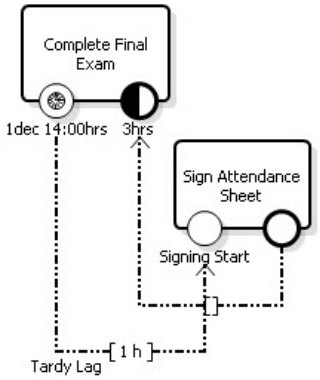

Figure 14. Time BPMN extending BPMN

VI. CONCLUSION

In this paper we presented Time-BPMN, an extension to BPMN that captures the temporal perspective of business processes. This extension deals with the various temporal constraints and dependencies that may occur while characterizing real world business processes. Time-BPMN allows for the depiction of temporal constraints and dependencies within a BPMN diagram. With this extension, BPMN becomes expressive enough to account for a very large set of real world business processes that are time dependent.

Time-BPMN is simple to draw, adequately expressive with respect to the temporal perspective, and is semantically unambiguous. The Time-BPMN extension does not conflict with or alter the current

specification of BPMN, and is thus a conformant extension to BPMN.

Note that the temporal constraints and dependencies within Time-BPMN are not prioritized. It is the scheduler's responsibility to find a consistent schedule (if one exists) which satisfies all the temporal constraints and dependencies. Scheduling is performed relative to a Case instantiation or finish time. The specification of a scheduler is beyond the scope of this paper.

Time-BPMN is part of an on-going research project at Trisotech to introduce time as a first class citizen to some of the main business process standards. We are currently working at extending Time-BPMN to capture the intervals of interest that were mentioned earlier (e.g., Transfer Time). We are also working at relaxing the limit of one start and one end temporal constraint per activity. For example, the combined constraints FASAP and FNET 3pm forces the activity to be scheduled so as to terminate at the earliest possible time after 3pm. Another avenue of research we are also working on is an XPDL extension to allow the serialization of Time-BPMN. The first version of this XPDL extension is presented in [5].

ACKNOWLEDGEMENTS

The contribution of the Trisotech (www.Trisotech.com) Development Team is gratefully acknowledged, Abel Ferreira, Mélanie Gauthier and Simon Ringuette in particular. The second author is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] Object Management Group (OMG), Business Process Modeling Notation (BPMN), Version 1.2, OMG Document Number: formal/2009-01-03, OMG, January 2009.

[2] P. Wohed, et al. "On the Suitability of BPMN for Business Process Modelling", In Dustdar et al. (Eds), Business Process management proc. of 4th International Conference (BPM2006), volume 4102 of LNCS, pp.161-176, Springer Verlag, 2006.

[3] M. zur Muehlen and J. Recker, "How Much Language is Enough? Theoretical and Practical Use of the Business Process Modeling Notation", In 20th International Conference on Advanced Information Systems Engineering (CAiSE 2008), LNCS, Springer Verlag, 2008.

[4] D. Gagné and A. Trudel, "The Temporal Perspective: Expressing Temporal Constraints and Dependencies in Process Models", Fisher, L. (Ed.), 2008 BPM and Workflow Handbook, pp. 247-260, Future Strategies inc., 2008.

[5] D. Gagné and A. Trudel, "Extending XPDL with the Temporal Perspective", Fisher, L. (Ed.), 2009 BPM and Workflow Handbook, Future Strategies inc., 2009.

[6] J.F. Allen, "Maintaining Knowledge about Temporal Intervals," Communications of the ACM 26, pp. 832-843, 1983.

[7] B. Silver, "My New BPMN Wish List", BPMS Watch Blog entry dated July 17, 2008. Retrieved on March 13, 2009: http://www.brsilver.com/wordpress/2008/07/17/my-new-bpmn-wish-list/