

# Business Process Evolution: a Rule-based Approach

Mario Lezoche<sup>1</sup>, Michele Missikoff<sup>1</sup>, Leonardo Tininini<sup>1</sup>

<sup>1</sup> LEKS, IASI-CNR, Viale Manzoni 30,  
00185 Rome, Italy  
{lezoche, missikoff, tininini}@iasi.cnr.it

**Abstract.** Business Process (BP) evolution is a key aspect of BP lifecycle. There are several reasons that may cause BP modifications. Among these, particularly important are the changes of the enterprise organization and operation strategies, which can be captured by business rules (BRs). We focus here on a BP-based enterprise that, in addition to the enterprise BPs, is also regulated by a set of BRs: such BPs and BRs need to be globally consistent (and have to be maintained consistent after any changes). In this paper we present an ontological approach capable of representing BRs and BPs in a coherent way. Then, our objective is twofold: (i) clarifying the intended meaning of a BR that (when firstly formulated) may have more than one interpretation; (ii) identifying all processes in the BP repository that are (or have become) inconsistent with the BRs and thus need to be changed to reestablish the overall consistency.

**Keywords:** Business Process Modelling, Business Rules, Ontologies

## 1 Introduction

Business process (BP) modelling is opening a new phase in the development of enterprise software applications (ESA), thanks to the recent proposal of the Model Driven Architecture (MDA) approach, promoted by OMG [1]. In essence, MDA proposes 3 levels of BP models:

- I. computational independent models (CIM), where business experts provide a first (informal) description of a BP (e.g. by using EPC [2]);
- II. platform independent models (PIM), where business experts work together with IT experts to build a procedural specification of the process in a rigorous way (e.g., by using BPMN [3]), but leaving out low level technical details;
- III. platform specific models (PSM), developed by IT experts, where all the technical details are introduced to achieve a complete specification, (e.g., by using BPEL4WS [4]), ready to be executed by a suitable engine (e.g., ActiveBPEL<sup>1</sup>) [5], invoking the required services.

The OMG-MDA proposal has been mainly conceived to support a layered development of enterprise software applications, however for what concerns BP lifecycle, and in particular BP evolution it does not propose any specific approach. In

---

<sup>1</sup> [www.activebpel.com](http://www.activebpel.com)

a dynamic enterprise, BPs need to be periodically revised and updated. Such BP evolution may be necessary for different reasons, for instance: poor functional performances (e.g., a process takes too much time or fails to fully achieve what expected), poor non-functional performances (e.g., security and privacy are not sufficiently guaranteed), new company policies (requiring the update of non conformant BPs). In this paper we address the latter issue, in a context where company policies are mainly expressed by business rules (BR).

Company policies are often represented in the form of business rules (BRs.) Business experts usually tend to formulate a BR in natural language. For instance, a BR can say: "all expenses greater than 5.000€ require the approval of the Head of Unit". Another BR may involve the way business operations are performed, for instance: "the receiving of a quotation must precede the issuing of a purchase order". The latter BR can be synthesized by the expression: *A precedes B*. In the context of BP evolution, a challenging problem is to automatically identify the BPs that violate a BR and, possibly, to make the former evolving according to the latter.

In this paper we present an ontological approach to BP modelling, proposing a method to verify if, given a BR, a process is consistent with it. To this end, we briefly introduce the BPAL (Business Process Abstract Language) ontological framework and we show how the semantics of a BP can be modelled. The modelling primitives of BPAL have been derived from BPMN. We focus then on the problem of a BR formulation in the context of a BPAL ontology and on how to decide if a given BP is still consistent with the BR content or it should be modified. Our proposal is positioned across the CIM and PIM levels of the MDA. The idea is to enhance intuitive modelling tools, such as BPMN, by associating to them a BP ontology system. The latter will provide a formal setting and the related semantic services.

Often, BRs are specified in natural language (NL) by business people and the intended meaning may not be sufficiently precise, therefore it is necessary to preliminarily understand what the BR content is, i.e., its semantics. The focus of the first objective of the paper is, therefore, on a method aimed at supporting business expert to disambiguate BRs that may be interpreted in different ways.

Once a BR has been disambiguated and reformulated to explicitly represent its intended meaning, we can address the second objective of the paper: BP consistency. Here it is necessary to first identify the BPs that violate the BR, and then to propose the updates necessary to make the former consistent with the latter.

BRs can represent a large variety of enterprise directives and constraints. Here we focus on a specific case, the procedural aspect of a BP, namely the sequence of activities, and BRs that indicate the correct order in which activities should be performed. Such BRs, having the form: *A precedes B*, have been extensively analysed by Van der Aalst [6]. Here, for sake of conciseness, we address the same problem, but our solution is inherently different since we address the intensional level, i.e., BP schemas, while in [6] the focus is on the extensional level, i.e., BP instances.

The literature reports a large number of relevant results in the areas of process modeling, rules representation and management and, in particular, results addressing the analysis of activities sequencing. However, we wish to point out that we avoided sophisticated approaches, such as those based on temporal algebra (see the proposals based on the Allen's work [7]) or temporal logic [8], and its variations (such as Event Calculus [9] and Situation Calculus [10].) Instead, here we approached the analysis of

BP and the sequencing of their activities by using the simple notion of precedence. Furthermore, we are not addressing here the problem of consistency of conditional expressions, such as those that may appear in a decision that determines the behavior of a branching. Here we simply record the nature of the branch, namely: and, or, xor. The simplified approach assumed in this work is already sufficiently expressive to produce a few interesting results and, at the same time, leaves space for future work where we plan to introduce a richer modeling paradigm.

The rest of the paper is organised as follows. The next section introduces a simplified version of BPAL to represent a BP and its specification. Section 3 illustrates how a BR can be expressed and formally represented, then Section 4 elaborates on the precedence rule and its different meanings. Here two are the main contributions: (i) how to support the business expert in clarifying his/her mind, when the BR looks ambiguous; (ii) how to verify if a given BP is contradicting the rule or not. Finally, Section 5 presents the conclusions and some directions for future research.

## 2 BP schema and instances in BPAL

BPAL [11] is an ontology modelling framework that allows a predicative specification of a BP Schema to be formulated. The simplified version of BPAL<sup>2</sup> adopted in the rest of the paper uses symbols for constants ( $a, b, c, \dots$ ), variables ( $?a, ?b, ?c, \dots$ ), and conditionals ( $?h, ?k, \dots$ ), as well as the following atoms to represent:

- Activities:  $act(?a)$
- Precedence relations:  $prec(?a, ?b)$
- Decisions:  $dec(?k, ?a)$

Note that decision atoms can be specialized to AND, OR or XOR type, e.g. a XOR decision will be represented by  $xdec(?k, ?a)$ . Where  $?a$  represents the then-activity in a typical IF-THEN-ELSE decision. When more than 2 mutually exclusive branches are available, it is necessary to use a sequence of  $xdec$ . Conversely, the AND, OR decisions essentially act as a semaphore “go/no-go” and therefore they only require a conditional.

We distinguish a BP Schema (BPS), which is a set of predicative atoms, from a BP Instance (BPI) represented by a chain of ground terms. A BPI originates from the actual execution of a BP Schema and each element of the chain (activity instance) corresponds to an execution of an activity (variable) in BPS.

Activity variables are assumed to be typed, according to a set of activity kinds, denoted by capital letters (A, B, C). To keep the notation lightweight, in this paper we avoid the explicit typing of activities, and the implicit typing is achieved by matching the first letter of the term. For example,  $type(?a, A)$  is always true. A similar method is applied to instantiation and activity instances are denoted by letters that correspond to the activity kinds. A relation  $inst(?a, a)$  says that  $a$  is an instance activity of  $?a$ . We will also refer to a BPI as a BP Log.

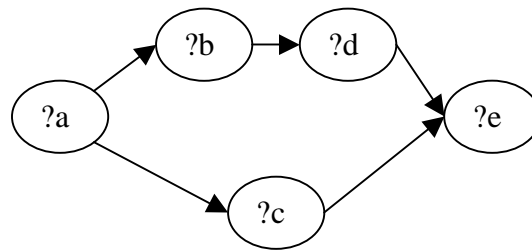
---

<sup>2</sup> A complete specification of the BPAL ontology framework falls outside the scope of this paper, can be found in [11].

Activity kinds are essential to express rules and constraints, since they are general and do not apply to a single BPS or BPI, but to all of them (unless otherwise specified.) Therefore a rule of the form:  $prec(A,B)$  imposes a control flow constraint on all activities  $act(?a)$  and  $act(?b)$  of type A and B, respectively. Then,  $prec(?a,?b)$  is consistent with a BR stating  $prec(A,B)$  and consequently all its executions will produce valid ground terms of the form:  $prec(a,b)$ .

In the following we show a simple example of a BPS modelled as a set of atoms, and its BPIs, modelled as chains of ground terms. In Figure 1, the following BPS is displayed:

$$bps1 = \{ act(?a), act(?b), act(?c), act(?d), act(?e), prec(?a,?b), prec(?b,?d), prec(?d,?e), prec(?a,?c), prec(?c,?e) \}$$



**Fig. 1.** A simple BPS diagram.

The execution of the BPS in Figure 1 will produce one of the following instances, depending on the kind of branching:

- 1) In case of an AND branching:  
 $bpi_{AND} = \{ \langle a,b,c,d,e \rangle, \langle a,c,b,d,e \rangle, \langle a,b,d,c,e \rangle \}$
- 2) In case of an OR branching:  
 $bpi_{OR} = \{ \langle a,b,c,d,e \rangle, \langle a,c,b,d,e \rangle, \langle a,b,d,c,e \rangle, \langle a,b,d,e \rangle, \langle a,c,e \rangle \}$
- 3) In case of a XOR branching:  
 $bpi_{XOR} = \{ \langle a,b,d,e \rangle, \langle a,c,e \rangle \}$ .

Please note that an activity instance is registered in the log at the moment of its completion. Therefore  $prec(b,c)$  may hold, even if the activity instance  $c$  has started before  $b$ , because  $c$  execution may take longer, causing  $c$  completion to follow that of  $b$ .

### 3. Representing Business Rules

Business rules are used by managers to indicate invariants that impact on the organization and its way of operating. Rules represent a very powerful modelling paradigm. Here we focus on the impact of rules guiding the design and the evolution of BPSs.

BRs are usually specified in natural language, but it is well-known that NL is often ambiguous and error-prone. For this reason, there are interesting proposals of using structured (controlled) natural language. Recently, OMG promoted the use of structured English in the business rules framework SBRV [12]. Another interesting proposal is ACE (Attempto Controlled English, [13]): a rich subset of standard English, designed for specification and knowledge representation. By using ACE, BRs can be expressed in rigorous way, having at the same time a simple (almost) natural language interface that can be easily used by business experts. ACE relies on a platform capable of translating a sentence expressed in a controlled natural language into a first order logic (FOL) formula. The latter can be then verified by using a theorem prover.

In this paper we show how the problem of BR formulation can be effectively solved in the context of a BPAL ontology. In particular, we will show how it is possible to support a business expert in formulating an example BR of the kind: “*A precedes B*”. The BR will be initially expressed in generic terms, then incrementally refined, by prompting the business expert with simple, specific questions, aiming at progressively removing the ambiguities.

### 3.1 On the semantics of a BR

As anticipated, in order to illustrate the BPAL approach to BR formulation, we use an example BR, informally expressed by the sentence: “*A precedes B*”. Although the intuitive meaning seems to be clear, a more careful analysis shows different possible interpretations. Here we will consider, as an illustrative example, a subset of the cases reported in [6]. The semantics of each case will be represented at an intuitive level, reporting a BPI.

- 1) *Response*: every time activity A executes, activity B has to be executed after it. B does not have to execute immediately after A, and another A can be executed between the first A and the subsequent B. Furthermore, an execution of B does not require to be preceded by A. i.e.  $\langle b, a, a, c, b \rangle$
- 2) *Precedence*: if activity B is executed, its (possibly multiple) executions must follow the execution of A. i.e.  $\langle a, c, b, b, a, b \rangle$
- 3) *Alternate response*: after activity A there must be an activity B, and before that activity B there can not be another activity A (but a B not following A is ok). i.e.  $\langle b, a, c, b, a, b \rangle$
- 4) *Alternate precedence*: every instance of activity B has to be preceded by an instance of activity A and the next instance of activity B can not be executed before the next instance of activity A is executed. i.e.  $\langle a, c, b, a, b, a \rangle$
- 5) *Chain response*: the next activity after each activity A has to be activity B, then the execution  $\langle b, a, b, c, a, b \rangle$  is a correct one.
- 6) *Chain precedence*: requires that the activity A is the first preceding activity before B (but is not necessary that a B follows an A) and, hence, the sequence  $\langle a, b, c, a, b, a \rangle$  is correct

### 3.2 Clarifying the semantics of a BR

Our goal is to disambiguate the BR “*A precedes B*”, with respect to the six reported cases, by posing few and simple questions to the user. Instead of using a question for each of the six (fairly complex) cases reported above, we propose a more intuitive disambiguation method, requiring a lower number of simple yes/no questions.

- α. Can I execute any activities between A and B?
- β. Can an activity B be executed before an activity A?
- χ. Does an activity B has to be executed after an A?

The proposed disambiguation method is based on these three questions and a Decision Table, as reported below. By answering these simple yes/no questions the user will be driven to clarify the intended meaning of the BR *prec(A,B)*, by selecting it among possible alternative meanings. Table 1 represents the mapping from answer combinations to possible meanings.

In the next section we will show the impact of BR onto BPS in relation to the intended meaning.

**Table 1.** Decision Table to disambiguate the BR “*A precedes B*”.

		Rules							
Questions	Can I execute any activities between A and B?	Y	Y	Y	Y	N	N	N	N
	Can an activity B exist before an activity A?	Y	Y	N	N	Y	Y	N	N
	Must, an activity B, be executed between two activities A?	Y	N	Y	N	Y	N	Y	N
Semantics	Response (1)		X						
	Precedence (2)				X				
	Alternate Response (3)	X							
	Alternate Precedence (4)			X					
	Chain Response (5)					X			
	Chain Precedence (6)								X

### 4. Impact of BR onto BPS

As anticipated the goal of this paper is twofold: to support the business expert in clarifying the intended meaning of a rule (when its interpretation is ambiguous) and verifying which BPs in a repository are not consistent with the rule. In the previous section we addressed the first objective, here we address the latter. To this end, we consider an example of a BPS fragment and analyse the impact that the constraint *prec(A,B)* has on it. Note that our approach is inherently different from BP instance based approaches (e.g. that in [6]), since we only address the intensional level, i.e., BP schemas. Note also that our approach can be applied even if some decision nodes are

not fully specified (i.e., as AND, OR, XOR), although more detailed and precise information about BR violation can be inferred when decision nodes are specified.

#### 4.1. Testing BP consistency to BRs

Table 2 shows the validity of a process schema for each of the six different semantics of the constraint “*A precedes B*” considered above. We assume that in order to be valid a BP must satisfy the rule for all its possible execution (instances). In the first column an example of business process schema is shown with two (generic) decision nodes. The possible specification combinations for the decision nodes are listed in the second column, in particular:

- *AND*: The presence of an AND operator creates a non determinism in the process execution. In fact, it is not possible to determine a-priori the execution order of two activities located on two parallel paths.
- *XOR*: In this case the two branches are mutually exclusive and the resulting logs will be constituted by the activity sequences of either one or the other branch.
- *OR*: The semantics of the OR decision node is the most articulated since the resulting logs is the union of the two previous cases.

The third column is organised in six sub-columns: one for each meaning that the natural language sentence “*A precedes B*” can assume, more precisely:

1. *Response*
2. *Precedence*
3. *Alternate Response*
4. *Alternate Precedence*
5. *Chain Response*
6. *Chain Precedence*

The row-column intersections show the validation results corresponding to the combinations of decision node specifications and sentence meanings, namely:

- *V*: Every instance of the schema validates the constraint.
- *N*: No instance of the schema validates the constraint.
- *S*: Some instances of the schema doesn't validate the constraint.

**Table 2.** Consistency Table for a partially specified BPS.

Schema	Decision		Constraint					
			1	2	3	4	5	6
	AND	AND	V	N	N	N	N	N
	AND	OR	V	N	N	N	N	N
	AND	XOR	V	N	N	N	N	N
	OR	AND	V	N	N	N	N	N
	OR	OR	S	N	N	N	N	N
	OR	XOR	S	N	N	N	N	N
	XOR	AND	V	N	N	N	N	N
	XOR	OR	V	N	N	N	N	N
	XOR	XOR	S	N	N	N	N	N

The values obtained (*V*, *N*, *S*) can be effectively used to support the BP designer in updating and refining the BP schema, by highlighting the inconsistencies and providing warnings for potentially inconsistent situations. Note that in 5 out of the 6 possible BR interpretations an inconsistency can be detected, even if both decision nodes remain unspecified. Conversely, when considering the first interpretation, the consistency can be verified in some cases, even if only one of the two decision nodes is specified.

#### 4.2. A practical example

In Figure 2, we present an example of a BPMN process (realised by using the Intalio Editor Tool [14]). This process deals with a procurement scenario; the two pools are two different organizations performing the roles of Buyer and Supplier. The Buyer sends the Request for Quotation (RFQ) to the Supplier which replies sending back a Quotation. Afterwards, the Buyer analyzes the Quotation and, if satisfied, sends the Purchase Order (PO) to the Supplier, otherwise, the Quotation is rejected. Invoicing from the Supplier and Payment from the Buyer concludes the process.

The illustrated BPMN process can be represented in BPAL as follows:

```

act (sendingRFQ) , act (processingRFQ) ,
act (sendingQuotation) , act (receivingQuotation) ,
act (issuingPo) , act (processingPo) , act (invoicing) ,
act (payingInvoicing) , act (rejectingQuotation) ,
prec (sendingRFQ, processingRFQ) , prec (processingRFQ,
sendingQuotation) , prec (sendingQuotation,
receivingQuotation) , prec (receivingQuotation,
quotationAccepted) , prec (xdec, rejectingQuotation) ,
prec (xdec, issuingPO) xdec (quotationAccepted,
issuingPO) , prec (issuingPO, processingPO) ,
prec (processingPO, invoicing) , prec (invoicing,
payingInvoice) .
    
```



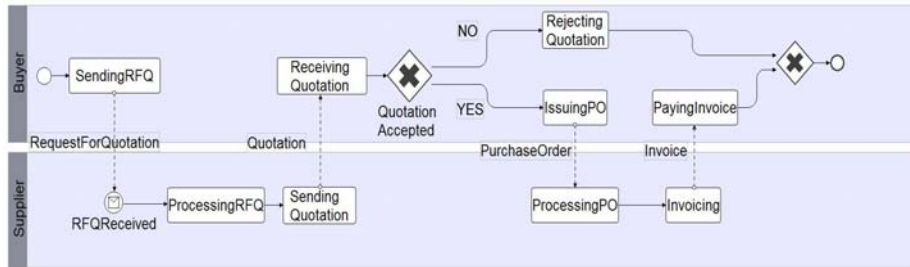


Fig. 2. A Purchase Business Process.

An example of BR to be tested on this BPS is the following: the activity of receiving a quotation must precede that of issuing a Purchase Order.

In order to disambiguate the BR we apply the decision table method presented above and we assume, for example, that the three yes/no answers of the table identify the Alternate Precedence (4) as the intended BR meaning. By analysing the consistency table for this BPS, we can infer that a path is inconsistent and some BP instances may be invalid. In particular the BPI can violate the specified BR if the quotation is not accepted. In general the management should therefore be promptly alerted that some modifications to the BPS, to make it consistent with the BR are required, sometimes could also happen that the management comprehend that the BR describes no more the reality of the company and it should be updated. In Figure 3 the branch arising the problem is highlighted.

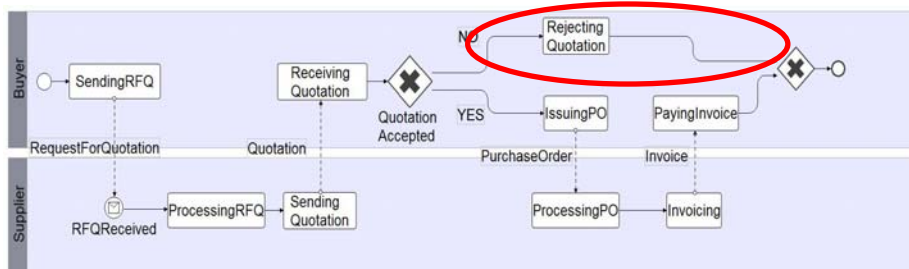


Fig. 3. The highlighted problem in the precedent Purchase Business Process.

### Conclusions

In this paper we have focused on some important issues related to BP evolution, in particular to changes of the enterprise organization and operations strategy, which can be captured by business rules (BRs). We have presented an ontological approach, where BPs and BRs can be formally represented in a unified context and their consistency checked and maintained by means of well-established tools (e.g. theorem-provers on decision tables). Our contribution is twofold. Firstly, we have illustrated a method for the disambiguation of BRs having multiple possible interpretations, based on decision tables and simple natural language questions. Secondly, we have

proposed an approach for identifying the processes in a BP repository that have become inconsistent after some BR change (update or insertion of some new ones, that require appropriate modifications in the inconsistent BPs), and supporting BPS, evolution to reestablish the overall consistency. This truth maintenance process can be applied also to partially specified BPS, say BPS Sketches. Then starting from a BPS sketch, with undefined decision nodes, and a BR, it is possible to consistently completing the BPS with the decision nodes appropriate for the selected BR interpretation. Our future studies will be focused on providing a method for the creation of a minimal set of questions to disambiguate a BR, and a general technique for updating the whole BPs in relation to the expressed BRs.

## References

1. Soley, R., et al.: Model Driven Architecture, OMG, 2000
2. Van der Aalst, W.M.P., Formalization and Verification of Event-driven Process Chains, *Information & Software Technology* 41, 639--650 (1999)
3. White, S.A.: Introduction to BPMN, IBM, 2004
4. IBM Inc., Microsoft Corp., BEA Inc., SAP AG, and Siebel Systems Inc: The Business Process Execution Language for Web Services. Version 1.1, May 2003, <http://www-128.ibm.com/developerworks/library/specification/ws-bpel>
5. Qian, Y., Xu, Y.: Tool Support for BPEL Verification in ActiveBPEL Engine. In: Proceedings of the 18th Australian Software Engineering Conference (ASWEC), pp. 90--100 (2007)
6. Van der Aalst, W.M.P., Pesic, M.: DecSerFlow: Towards a Truly Declarative Service Flow Language. In: Bravetti, M., Nunez, M., Zavattaro, G. (Eds.) WS-FM 2006. LNCS, Vol. 4184, pp. 1--23, Springer (2006)
7. Allen, J.F.: Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26, 832-843, (1983).
8. Gabbay. D.M.: Temporal Logic: Mathematical Foundations and Computational Aspects. Oxford University Press (1991).
9. Kowalski R. and Sergot M.: A logic-based calculus of events. *New Generation Computing*. 4(1):67-96 (1986).
10. Pirri F., Reiter R.: Some contributions to the metatheory of the Situation Calculus. *Journal of the ACM*, 46(3):325-361 (1999)
11. De Nicola, A., Lezoche, M., Missikoff, M.: An Ontological Approach to Business Process Modeling. In: Proceedings of the 3rd Indian International Conference on Artificial Intelligence (IICAI), pp. 1794--1813. (2007)
12. OMG Group: Semantics of Business Vocabulary and Business Rules Specification. OMG (2006)
13. Fuchs, N.E., Kaljurand, K., Kuhn, T.: Discourse Representation Structures for ACE 6.0. Technical Report ifi-2008.02, Department of Informatics, University of Zurich (2008)
14. Intalio Business Process Management Suite, <http://bpms.intalio.com/>