BP-FAMA: BUSINESS PROCESS FRAMEWORK FOR AGILITY OF MODELLING AND ANALYSIS

Mohamed Boukhebouze, Youssef Amghar and Aïcha-Nabila Benharkat
LIRIS, National Institute of Applied Sciences of Lyon, Lyon, France

Keywords: Business process modelling, modelling methodology, business rules, business agility, business process verification.

Abstract: In this paper we present the BP-FAMA framework (Business Process Framework for Agility of Modelling and Analysis), motivated by the need to have an abstraction level in which algorithmic preoccupation and the management rules are separated for a better evolution of processes. Nowadays the approaches proposed are based on a mixing of business rules and algorithmic structures making the process difficult to change. The other motivation is the improvement of the quality of the translation of process specification to high-level executable process. These objectives don't go without the need for new agile modelling languages: BPAMN (Business Process Agile Modelling Notation) and a new agile execution language: BPAEL (Business Process Agile Execution Language).

1 INTRODUCTION

Efficient organizations working impose to refer to robust business processes adapted to their activities. The definition and execution of these processes require respectively a model and tools for collaboration, definition, deployment, implementation, and process control. The business process management (BPM) consists in managing from beginning to end of company business processes to get a better overview with their interactions. The objective is to enable policy deciders, business analysts, functional teams and technical teams to collaborate in the definition and evolution of business processes via a single tool aggregating different visions. It is also designed to optimize these processes, and try to automate them as much as possible with the help of business applications.

The process modelling is then an important step in this management, because it allows specifying the business knowledge of a company. For this reason, it must be based on powerful languages in order to give a full business process description. In this context two standards are proposed: a common graphical notation for modelling tools called BPMN, and process execution language called BPEL, to make processes portable on different platforms. These two specifications are now stable and adapted to business needs. Several editors have adopted and included them in their tools. However, these specifications have to be checked before their implementation, unfortunately, these standards focus more on the business description level, including functional aspects of a process, without providing mechanisms to support the specifications verification. Indeed, both the reliability of the process and maintenance costs drives us to give a great attention to verification issues.

Business rules are a collective knowledge which involve shared representations of behaviours or business models (organizational rules), the common representations on the environment for example (the facts), and the language used to communicate and interpret the rules and facts. By the dynamic nature of the environment, the facts change, the rules change because the organization can change. The language can change as well because it corresponds to successive interpretations made before.

Accordingly two major problems are identified: (1) Mixing algorithmic preoccupation and management rules at the abstraction level. This mixing makes the process difficult to change: agility. (2) Even if there are tools that allow the transition from a process specification to a high-level BPEL process, nothing guarantees the quality of the process generated at the analysis level.

In this paper we describe the architecture of the BP-FAMA framework (Business Process Framework for Agility of Modelling and Analysis) which is in progress within our laboratory and which responds to those two problems. We introduce in section 2 our motivations for proposing this framework. In section 3 we describe the BP-FAMA

Boukhebouze M., Amghar Y. and Benharkat A. (2008).
BP-FAMA: BUSINESS PROCESS FRAMEWORK FOR AGILITY OF MODELLING AND ANALYSIS.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - ISAS*, pages 368-373
DOI: 10.5220/0001712703680373

architecture. Section 4 specifies in more detail the two new languages BPAMN and BPAEL. In the remaining sections we describe an illustration through a use case and discussion of our work.

2 MOTIVATIONS

To reach our goals, we have attempted to answer two questionings: how to model a business process in an agile way and how to analyse a business process. First, the process modelling is then an important step in this management, because it allows specifying the business knowledge of a company. For this reason, it must be based on powerful languages in order to give a full business process description. In this context, several process description languages have been proposed, such as UML (OMG, 2007) and BPMN (OMG, 2006) YAWL (Van der Aalst, 2005). These languages are generally intended to represent the definition of a process using graphical notations that can intuitively describe a business process. This convivial manner increases mainly the comprehension and visibility of these processes, makes possible the design evaluation and allows then the generation of the associated executable language. The latter specify the fulfilment of the process activities. These languages which are interpreted by execution engine use XML syntax to describe the implementation of the process. In the literature several languages execution process have been proposed. BPEL4WS (OASIS, 2007) are the most commonly known language because it can represent a larger number of business process basic elements compared to its predecessors. For this reason, it becomes the most used language in industry.

However, using these languages and reference models, designers implement business rules in the code of business process, this practice makes business process rigid and difficult to maintain. Indeed, the dynamic nature of organizations environments makes the rules subject to modification. Therefore, the integration of algorithmic logic and business rules into the same code processes compromises agility of modelling. To solve this problem, business rules tracking and identification in a business process modelling seems necessary and important to allow the evolution and maintenance of rules independently of the process and make sharing these rules possible by other processes.

Secondly, companies must be based on robust business processes to achieve their objective. The process reliability is a crucial issue because they automate all or part of the company value chain and at the same time capitalize on their information system. An erroneous business process can have economically grievous effect. Several studies have been conducted to investigate the nature of the errors and exceptions occurred in a business process like (Russell, 2006). In general way, these exceptions can be divided into two classes:

2 - The random exceptions relate to events which are not be modelled by the designer and which are liable to cause exceptions. In fact, these events are infrequent and unexpected for example a hardware failure in the computer system can destabilize the business process operation or unavailability of one or more resources when an activity instance in process wants access. To assist the designer to find errors in the process specifications, a number of techniques are used:

1- The verification by formal models is viewed as the most used technique to detect errors in specifications during a business process modelling. These models need to formalize the specification in a formal model (e.g., the Petri net, process algebra… etc.). The objective, of using these models in specification process, is double: on the one hand, they provide the complete process description reliance on its requirement by eliminating the contradictions and ambiguities (formal specification). On the other hand, they minimize the likelihood that a malfunction can occur while on basing on the verification properties of models for example deadlock or net vivacity...etc. In this way we can detect errors by taking into account the model properties. We have identified a number of works that use this technique to verify the proper functioning of business processes. In general way, the Petri Net (PN) is the model mostly used (Martens, 2003) and (Yang, 2005) because it combines the advantages of graphical representation with the semantics aspect attributed to the modelled processes behaviour. However, the process algebra also has imposed on the process verification (Koshkina, 2004). It consists of a language based on a mathematical formalism dedicated to the description of concurrent systems. This model is based principally on the competition theories and the algebras techniques.

2- The design verification aims to find the elements which can lead to errors or elements which represent a potential source of errors in a business process modelling. This technique offers a good modelling style by requiring to the designers to respect modelling rules in order to minimize the errors risk. An example of these rules is proposed by (Gruhn, 2007) for the verification of EPC and Dongen's work (Van Dongen, 2005) which checks MySAP reference models across ARIS AGL.

The design technique verification can't replace the verification by formal models, because it tries only to limit the well-known error. The combination of verification design technique and verification formal models technique proves to be profitable to analyze a business process modeling, because it increases the certitude that failure will not occur. However, the algorithms time consumption of these technologies depends on the complexity of the algorithms translation of the business processes definition in formal models and the complexity of algorithms for detecting properties to be checked. This time is added to the effective time of business process execution by an execution engine. What is more, the change in a part of a process leads to its re-verification in totality. On the other hand, these techniques focus only on the coherence of the processes and don't guarantee that the implemented process meets the requirements of designers. For this reason, the need for a framework that helps us to solve these problems is growing. For this way, we propose BP-FAMA framework which allows:

- Favour flexibility of business processes modelling and execution to take into account the dynamic character of the different business processes elements (business rules).
- React in an unstable situation and reach a more stable situation i.e. respond to exceptions runtime in order to avoid to call algorithms verification at each change of a process element, as well as avoid interrupting the business processes execution at each random exception.
- Analyze the fulfilment of process executable to verify that it agree with what has been modelled.

3 THE BP-FAMA ARCHITECTURE

The objective of the FAMA is to improve the functioning of BPMS (see Figure 1). In fact, this platform can:

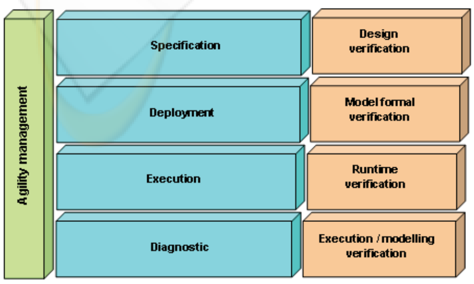


Figure 1: the BP-FAMA architecture.

- Favour the agility of the modelling business processes. For this reason, we propose two new modelling languages, the first language is called BPAMN (Business Process Agile Modelling Notation) is a graphical language very similar to its ancestor BPMN. This new language allows the identification of the business rules in modelling level. In this manner, the monitoring of these rules is possible from beginning to end of a business process. The second language is called BPEDL (Business Process Elements Description Language) is a language describing the various elements of business processes, this language allows to add information to complete the definition graphics process. This information will be used to ensure consistency when we integrate updates of the business rules automatically.
- Favour the agility of the process in running. For this reason, we propose a new execution language called BPAEL (Business Process Agile Execution Language) which is based entirely on his ancestor BPEL and extends with location and identification of business rules in a process BPAEL. This is done by adding a new activity structured called "RULE".
- Integration of a verification step in each phase of a business process life cycle. In the specification phase, the designer is assisted to model the process by detecting elements that can be a potential source of errors. In the deployment phase, functional coherence process modelling is verified by using a formal model. In the execution phase, the verification is launched in demon to intercept possible exceptions and try to react in order to drive the execution of the process towards a stable situation. In the diagnostic phase, the process is rebuilt from the information accumulated during the execution of a large number of the process instance, to ensure that what has been modelled corresponds to what actually is running.
- Analyze the fulfilment of process executable to verify that it agree with what has been modelled.

3.1 The Specification Phase

In the specification phase, the designers define the elements constituting the business process or they redefine the elements of a process in order to improve it. This definition is a dialogue medium between the processes responsible and operational teams in charge of executing them. To this end, graphical languages are used in this phase to describe the process models because they provide a convivial representation and an easier way to understand the process. The new graphical language, called BPAMN, is used in order to describe a FAMA business process (see Figure 2).

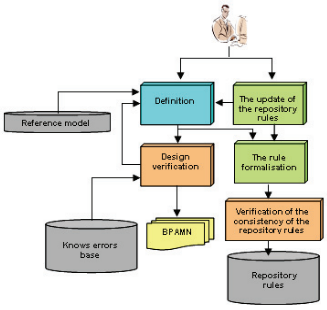


Figure 2: Architecture of the specification phase.

This language allows using a complete graphical notation (graphics and charts) for the representation of a business process. This graphical notation is inherited from the standard BPMN. The advantage of this new language is the separation of the business rules definition from the algorithmic logic of a business processes. This separation makes possible to independently manage business rules using a rules repository, and allows the sharing of rules by multiple processes. The check of rules updating is necessary to maintain the coherence of the rules repository. This is why the formulation of business rules must be rigorous, concise and precise to ensure that those rules are unambiguous, coherent, complete and enunciate with a common business vocabulary. At the same time, an early verification is performed to identify known errors in the modelling process.

3.2 The Deployment Phase

In the deployment phase, the process model is implemented by developing business applications needed in the allocation of resources which carry out different process activities. In this way, the process becomes operational. The FAMA business process is implemented by translating its modelling expressed using the BPAMN in a new execution language called BPAEL (see figure 3). Indeed, this new language is entirely based on the BPEL standard, at which we add a new structured activity called "Rule". Thelatter allows identifying the business rules in a process BPAEL. In that way, updates rules can be incorporated into the process automatically. However, the language BPAEL, like its predecessor BPEL, doesn't provide mechanisms to support the verification of process specifications. For this raison, we have used formal models to identify possible functional errors. Among the different models used in the literature for this type of verification, we opted for Petri Net (PN) due to their great ability to model a wide variety of business processes.

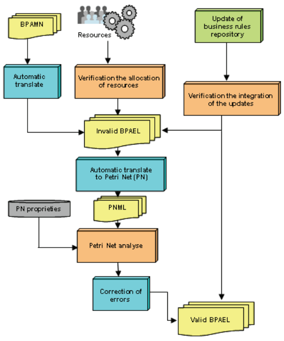


Figure 3: Architecture of the deployment phase.

Moreover, the PN offer a wide range of mathematical properties for analyzing the proper functioning of the process. To this end, a tool for rewriting the specification of the process in terms of PN (translators) is proposed. The Petri Net obtained are expressed in the language PNML (Petri Net Markup Language) which is used by large PN analysis tools (analyzers). The latter allow certain properties to verify for instance, deadlock or net vivacity...etc. They can detect errors by taking into account the properties detected. Finally, the allocation of resources can be checked using process elements descriptions expressed in BPEDL.

3.3 The Execution Phase

In the execution phase, an execution engine interprets the fulfilment specification of process activities which are expressed by using the FAMA execution language (BPAEL). This interpretation is performed by automating interactions between process participants (the documents, the information and tasks) and allocating the different resources. A verification demon is launched in parallel with the execution of FAMA process in order to intercept non modelled exceptions like a hardware failure or an unavailability of a resource (see figure 4).

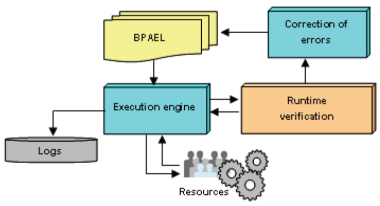


Figure 4: Architecture of the execution phase.

In this way, we try to reach from an unstable process situation to more stable situation. We also avoid calling verification by PN model, which is costly in terms of execution time (time of performing the translation algorithm plus analysis algorithm). Finally, the different log files and traces accumulated during the execution of a large number of process instances are stored in a specific base to be used in the next step.

3.4 The Diagnostic Phase

In the diagnostic phase, the executable process is analyzed in order to measure the operational performance basing on the log files. For this raison, the FAMA executable process is analyzed in order rebuilt the business processes based on information accumulated in the previous phase (see Figure 5).

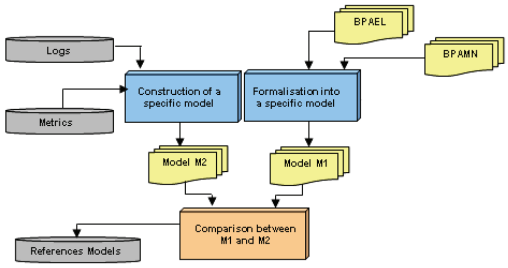


Figure 5: The architecture of the diagnostic phase.

The result will be compared with the modelled process to check if this last correspond to the same set of activities which are actually running.

In the following, we will describe the two new FAMA languages: BPAMN and BPAEL.

4 THE FAMA LANGUAGES

4.1 The BPAMN Language

The specification of a business process consists in formal description of the basic elements that constitute the process by using a meta-model. The latter allows the definition of the syntax and semantics of a process respecting meta-model. For this reason, graphical notations are used to help an intuitive description of a business process. In this way, the representation of the process becomes understandable and enables evaluation of the design phase as well as the generation of the associated executable code. Whereas, a complete representation of the business enterprise knowledge is necessary to automate and analyze the process, this obviously depends on the power of the meta-model used. BPAMN comes to add an agility dimension into a BPMN representation. Indeed, this new language allows the identification of the business rules in modelling level by using new symbols dedicated to model the business rules elements separately from the other business process elements. In this manner, the location of these rules is possible from beginning to end of a business process.

4.2 The BPAEL Language

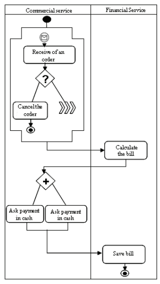
In this section we describe a new process execution language called BPAEL. It is proposed in order to trace business rules in the process. Indeed, the BPAEL is based entirely on the BPEL standard, at which we add a new structured activity called *Rule* for the location of the business rules in a process BPAEL. Indeed, this new activity combines the operation of the *Pick* activity which allows intercepting events and *If* activity which allows introducing conditions. A rule identifier is added into this new activity in order to keep track of business rule within rule repertory (see figure 6).

```
<Rule RuleName = "ncname" >
  <onMessage partnerLink="ncname" portType="qname"
   operation="ncname" variable="ncname"?>*
     <Rule condition="bool-expr">+
       Activity+
     </ Rule condition >
  </onMessage>
<onAlarm (for="duration-expr" | until="deadline-expr")>*
     <Rule condition="bool-expr">+
       Activity+
     </ Rule condition >
</onAlarm>
</Rule>
```

Figure 6: The declaration of the rule activity

5 USE CASE

In this section we introduce the famous example of purchase order process to illustrate BPAMN language (see Figure 7.A) and illustrate BPAEL language (see Figure 7.B).




```
<process name="PurchaseOrder">
<sequence>
  <Rule IDName="R1">
   <onMessage partnerLink="ComercialService"
    <RuleCondition>
      Bpel: IsClientIsRegistered("ClientName") = false
    <RuleCondition>
    <Reply "Purchase order Cancelled"/>
   </onMessage>
  </Rule>
   <invoke operation="requestBillPayment"/>
   <if>
    <Condition>
     Bpel: PaymentInCash()= true
    <Condition>
    <invoke operation="Save Payment Bill in casch"/>
   </if>
   <else>
   <invoke operation="Save Payment Bill in credit card"/>
   </else>
 </sequence>
</process>
```

(A) (B)

Figure 7: Purchase order process.

On receipt of an order from a client, this process calculates the final price and sends a bill to the client, the latter has the option to pay in cash or by credit card. The bill was finally registered. However, this process must respect a constraint: requires that the customer must be saved in the database in order to satisfy his command. When we went to model this business process, we have to identify and represent its different elements using a modelling lanuage.

The Figure 7.A shows a graphic representation of the purchase order process by using BPAMN. The rectangles with rounded corners represent the activities of the process as "*Calculate the initial price*". In other hand, the designer of this business process can predict the business rules which are more susceptible to change. We assume that the constraint which the process must respect has a good chance to change. For this reason the designer will need to separate its definition from the process. By using BPAMN, this separation is possible. Indeed this constraint (business rules) will be represented by using a cross. As a result, the rule represented by the first cross figure will trigger if the receipt of an order (receipt event message is represented by the icon), if the customer is registered (this condition is expressed by the icon), then we continue to execute the process. To express this mechanism we use the symbol rule link. Otherwise (if the client is not registered) stops the process by cancelling the order. At the same time, the identification of business rules within a BPAEL process is done by using the RULE activity. The rule is implemented in the RULE activity (see Figure 7.B), this last will trigger if the receipt of an order (*<onMessage>*), if the customer is registered (*IsClientIsRegistered("ClientName") = false*), then we stops the process (*<Reply "Purchase order annulled"/>*) .

6 DISCUSSION

The BPM come today to include the entire life cycle process. Indeed, this cycle is beginning from definition of processes, through deployment and execution until the analysis of these processes. The modelling phase is crucial for a company. Because it helps to describe its value chain. Especially since it is a means of dialogue between processes responsible and operational teams in charge of executing them. To be successful, it must be based on methods and standards languages. In this context two specifications have been proposed: the BPMN and BPEL. Unfortunately by using these specifications, the designers face up to two problems: 1) the implementation of business rules in the business process code makes the latter rigid and difficult to maintain. 2) The lack of mechanisms to support the verification process. For this raison, we have proposed in this paper a framework called BP-FAMA which tries to respond to these two problems. Indeed we believe that business rules must be identified by the designer during the specification phase and the deployment phase of the process. The lack of a rules identification mechanism in both standard BPMN and BPEL pushed us to propose extensions to these standards: the BPAMN language graphical modelling agile processes and BPAEL language implementing agile processes. This identification rule in a business process allows keeping track of rules in order to manage them separated from the algorithmic logic of the process and also to integrate its updated automatically. We also believe that providing a complete analysis to a business process, the integration of a verification step in each phase of the business process life cycle is necessary. In the specification phase by detecting elements that can be a potential source of errors. In the deployment phase by using a formal model. In the execution phase by trying to react in order to drive the execution of the process towards a stable situation. In the diagnostic phase rebuilding the process to ensure that what has been modelled corresponds to what actually is running.

REFERENCES

Cardoso, J., Sheth, A., 2006. The book: Semantic Web Services, Processes and Applications, Springer 2006

Gruhn, Volker., Laue, Ralf., 2007 What business process modelers can learn from programmers, In Science of Computer Programming Journal 65 (2007) 4–13.

Koshkina, M., van Breugel, F., 2004. Modelling and verifying web service orchestration by means of the concurrency workbench. In ACM SIGSOFT Software Engineering Notes.

Martens, A., 2003. On usability of web services. In 4th International Conference on Web Information Systems Engineering Workshops, Italy.

OASIS, 2007. Business Process Execution Language for Web Services (BPEL4WS): Version 2.0. In BPEL4WS specification report.

Object Management Group, 2007. Business. Unified Modelling Language Superstructure version 2.1.1 In final report of adopted specification 07-02-03.

Object Management Group, 2006. Business Process Modelling Notation (BPMN) Specification. In final report of adopted specification dtc/06-02-01

Russell, N., van der Aalst, W.M.P. ter Hofstede, A.H.M.., 2006. Exception Handling Patterns in Process-Aware Information Systems. BPM Center Report BPM-06-04

Van der Aalst, W.M.P., ter Hofstede, A.H.M., 2005, YAWL: Yet Another Workflow Language. In Information Systems, 30(4):245-275.

Van Dongen, B.F., Jansen-Vullers, M.H., 2005. Verification of SAP Reference Models. In BPM 2005, 3rd International Conference, France.

Yang, Y., Tan, Q. Yu, J., Liu, F. 2005. Transformation BPEL to CP-nets for verifying web services composition. In the International Conference on Next Generation Web Services Practices, Korea.