# Towards a Process Patterns based Approach for Promoting Adaptability in Configurable Process Models

Hanae Sbai, Mounia Fredj and Laila Kjiri
*ALQualsadi research team, ENSIAS, Mohammed V – Souissi University, Rabat, Morocco*

Keywords: Business Process Management, Configurable Process Model, Variability, Evolution, Process Pattern.

Abstract: Reference process modeling approach has been proposed in order to provide a generic solution for reuse in business process management. These models are intended to be adapted by different organizations to respond to new requirements. As a result of this adaptation, a set of variants of the reference process model must be managed. In this context, configurable solutions have been developed to manage variability by integrating all possible configurations. However, enterprises are involved in competitive and complex environments which impose them to often adapt their configurable process models to deal with the increasing globalisation. Therefore, configurable process models should evolve over time in terms of activities, resources and data to meet new needs. In this paper, we propose a process patterns based approach to guide designers in evolving configurable process models at design time as a first step of our contribution to provide an automated support for evolving configurable process models. The presentation of our process patterns system is limited in this paper to the Activity Insertion Process Pattern.

## 1 INTRODUCTION

Today, enterprises should adapt their business process models to meet new needs, imposed by rapidly changing technologies and business models. For this, research on "Business Process Management" (BPM) stresses the importance of business process reuse. BPM is an approach which suggests the alignment of information systems through a process-oriented approach, driven by a Process Aware Information System (PAIS), PAIS provides a separation between the layer application and layer process (Weber, 2008). Moreover, BPM proposes a generic system through an explicit design process using the concept of process model (Aalst, 2003). Thus, the main goal of BPM is reducing the cost and time of process design and application development. However, the PAIS system introduced by this approach does not provide sufficient mechanisms to reuse business process models. Therefore, the *reference process model* appeared to allow the reuse of business process models. This model identifies common practices and activities of organizations and offers a generic solution for business process execution. As a result, this solution can be adapted by multiple users to meet different needs. In this context, the *configurable process model* has been proposed to manage all variants of the process (Rosemann, 2008); (Gottschalk, 2009); (La Rosa, 2009); (Hallerback, 2009); (Ayora, 2011). This kind of model allows reusing different process variants, if all the variation options have been integrated within the model beforehand (Gottschalk, 2009).

When there is a need to integrate information systems of enterprises, configurable process models must be adapted to meet new requirements of the new information system. In this case, process model configuration has to be enhanced with adaptation mechanisms, to add new behaviour of the configurable process model. This adaptation can be made with a basic technique (addition/substitution and deletion of the process elements) or with other techniques namely aggregation, specialization or instantiation of configurable process models. Generic adaptation of configurable process models can be found in (Becker, 2007). Hence, it is essential to dispose of a process which guides designers to manage evolution of configurable process models in terms of simple or advanced adaptation mechanisms. In this paper, we propose a solution for reuse process, in using process patterns system to guide designers in evolving configurable process models at

Sbai H., Fredj M. and Kjiri L..
Towards a Process Patterns based Approach for Promoting Adaptability in Configurable Process Models.
DOI: 10.5220/0004565503820387
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 382-387
ISBN: 978-989-8565-61-7

design time. As the first step of our contribution, we focus on the basic adaptation mechanisms. This work is an extension of our previous work (Sbai, 2012).

The paper is organized as follows. Section 2 is devoted to presenting related works. Our approach for managing evolution of configurable process models is introduced in Section 3, with an example of application. Finally, some research perspectives are developed in the conclusion.

## 2 RELATED WORK

Some research has  dealt with reuse in business process modeling and this under two aspects : i) managing business process variability which focuses on modeling variability, and controlling process configuration, ii) managing business process evolution which determines mechanisms for managing evolution at design time and runtime.

Among the work reviewed in the perspective of managing reference process model variability, we find configuration mechanisms such as the behavioural model solution (Gottschalk, 2009), (La Rosa, 2009); (Lu, 2009); (Aalst, 2009) and the structural model solution (Schnieders, 2006); (Hallerback, 2009); (Groner, 2010); (Rolland, 2010). We find work on variability modeling namely Rich-variant (Puhlman, 2005), Hierarchical method (Razavian, 2008) which proposes representing variability of BPMN models at the activity and control flow level, and Configurable nodes (Rosemann, 2008) which propose C-EPC and C-iEPC configurable process model languages. Moreover, BVL/CVL approach proposes a separate model for modeling variability at the control flow and task related elements level (Ayora, 2012). In the perspective of managing evolution in BPM, we mention change patterns and change support features work (Weber, 2008), frameworks for an automatic adaptation during the execution of business process such as AristaFlow (Muller, 2008) and Adept2 (Dadam, 2009). We find also several studies on maintaining configurable process model changes, namely the work of (Becker, 2007) which defines a set of generic adaptation concepts for adapting EPC reference process models. The work of (Li, 2010) proposes a heuristic approach to design a new reference process model by measuring the distance between the new reference process model and existing variants. Lately, extending adaptability of C-EPC reference process models by specialization has been supported by the ADOM-EPC formalism (Berger, 2011).

Solving process model evolution can be summarized in the following questions (Jaccheri, 93): which process model fragments should be changed, how and when? How to analyze and **guide change**? According to this survey, we find that there is a lack of a **guide** which assists designers to evolve configurable process models at design time. For this, a process patterns system which contains a set of process patterns for guiding evolution of activity, data and resource (as variation point or variant), is introduced in this paper.

## 3 A PROCESS PATTERNS BASED APPROACH FOR MANAGING EVOLUTION

The aim of the proposed process patterns system is to guide designers when evolving configurable process models in terms of activity, resource and data. Each configurable process model elements (activity, resource, and data) may be a variation point with a set of variants. So, if a need to build a new configurable process model occurs, we can have the following basic evolution types for each process elements (activity, data and resource):

- Insertion/ Substitution/ Deletion of a variation point
- Insertion/ Substitution/ Deletion of a variant in a variation point

In order to represent the proposed process patterns, we use the P-Sigma formalism (Conte, 2002) which is composed of three parts: the Interface part, the Realization part, and the Relation part.

The *Interface part* contains all elements allowing the patterns selection:

- **Identification**: defines the couple (problem, solution) that references the pattern.
- **Classification:** defines the pattern function through a collection of domain keywords
- **Context:** describes the pre-condition of pattern application.
- **Problem:** defines the problem solved by the pattern.
- **Force:** defines the pattern contributions through a collection of quality criteria.

The *Realization part* gives the solution in terms of Model Solution and Process Solution.

The *Relation part* allows organizing relationships between process patterns with three specific relations:

- **Uses**: if a pattern P1 *uses* a pattern P2, P1's Process Solution must be expressed using P2.
- **Refines**: if a pattern P1 *refines* a pattern P2, P1's Problem must be a specialization of P2's one.
- **Requires**: if a pattern P1 *requires* a pattern P2, P2 must have been executed before executing P1.

We present in the following figure the proposed process pattern system.

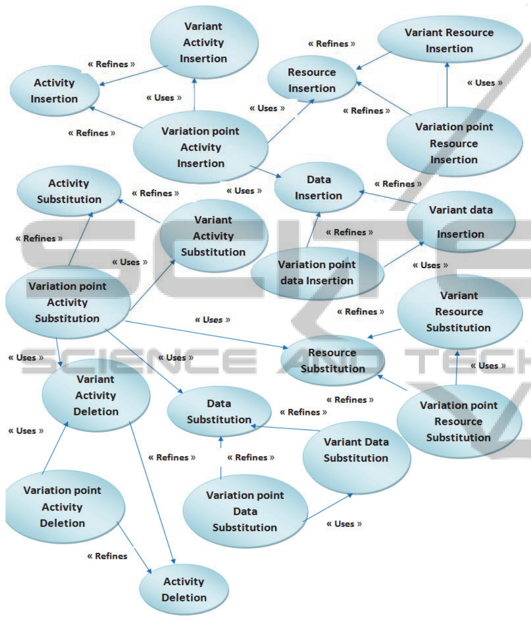
Variant Activity Insertion
Activity Insertion
Variant Resource Insertion
Resource Insertion
Variation point Activity Insertion
Variation point Resource Insertion
Data Insertion
Activity Substitution
Variant Activity Substitution
Variant data Insertion
Variation point data Insertion
Variation point Activity Substitution
Variant Resource Substitution
Resource Substitution
Variant Activity Deletion
Data Substitution
Variation point Resource Substitution
Variant Data Substitution
Variation point Activity Deletion
Variation point Data Substitution
Activity Deletion
« Refines »
« Uses »

Figure1: The process patterns system.

For each evolution type (insertion, substitution, deletion), a set of process patterns collaborate using defined relations (uses and refine) in order to lead evolution of configurable process model. In the next section, we present in detail the Activity insertion process pattern.

## 3.1 The Activity Insertion Process Pattern

In order to manage evolution of the configurable process models when adding new activity element (the activity may be a variant or a variation point), a set of process patterns must collaborate to lead this evolution. The Activity process pattern is refined by the variant activity process pattern and Variation point Activity process pattern. It represents the generic process that decides which insertion process pattern has to be applied. The application of the Variation point Activity insertion can invoke insertion of a new data (variation point/variant), or a new resource (variation point/variant).

### 3.1.1 The Interface Part

We present in this section, the Interface part of the following process patterns: the Activity insertion (table 1), Variant activity insertion (table 2), and Variation point Activity insertion (table 3).

Table 1: the Interface part of Activity Insertion process pattern.

| **Identification** | Activity Insertion |
|---|---|
| **Classification** | Provides a generic process for inserting an activity as a variation point or a variant in the configurable process model |
| **Context** | --- |
| **Problem** | Allows evolving the configurable process model with inserting an activity |
| **Force** | helps in the decision of which Activity Insertion process pattern we have to apply depending on the type of activity (variant/variation point) |

Table 2: The Interface part of Variant activity Insertion process pattern.

| **Identification** | Variant Activity Insertion |
|---|---|
| **Classification** | Provides a process for inserting an activity as a variant in the configurable process model |
| **Context** | **Refines** {Insertion Activity Process Pattern} |
| **Problem** | Allows evolving configurable process model with inserting a variant activity. |
| **Force** | Allows verifying a set of constraints before inserting a variant activity. |

Table 3: The Interface part of Variation point Activity Insertion process pattern.

| **Identification** | Variation point Activity Insertion |
|---|---|
| **Classification** | Provides a process for inserting an activity as a variation point in the configurable process model |
| **Context** | **Refines** {Insertion Activity Process Pattern}<br>**Uses**: {Data Insertion, Resource Insertion, Variant activity Insertion} |
| **Problem** | Allows evolving configurable process model with inserting a variation point activity. This insertion can invokes insertion of a data, insertion of resource or assignment of resource. |
| **Force** | Allows verifying a set of constraints before inserting a variation point activity. |

In the Realization part, we present the process solution of the Activity Insertion, Variant Activity insertion, and Variation point Activity Insertion process patterns.

### 3.1.2 The Realization Part

The realization part of the activity insertion process pattern provides a process solution as a set of algorithms. The proposed process solution suggests the definition of two main concepts, to optimize the assignment of a resource to a given activity:

- **Functionalities** of a resource: represents all functions ensured by the resource to perform an activity.
- **The required functionalities** of an activity: represents all needed functions of an activity to be performed by a resource.

For presented algorithms, we need to define the following variables**:**

*Req_f_AVP: Required functionalities of a variation point activity*

*Req_f_AV: Required functionalities of a variant activity*

*F_R: Functionalities of resource*

*C_nbr_A: Current number of activities assigned to a given resource*

*Max_nbr_A: Maximum number of activities assigned to resource*

***a) Activity Insertion Process Pattern:***

```
Detect change type
  If change type=”insertion” then
    Check the type of activity change // variant change or a variation point change
    If the activity change is a variant change    then
      Apply the Variant Activity Insertion process pattern
    Else if the activity change is a variation point change then
      Apply the Variation point Activity Insertion process pattern
      End if
  End if
```

***b) Variant activity Insertion Process Pattern:***

```
If the activity is an add variant then
  Check the type of the add variant
    If the type of the add variant is not an activity then
    refuse the insertion
    Else check the nature of the add variant activity // an activity can be a task or a sub-process
      If the add variant activity is a sub-process then
      list all the tasks
      End if
    End if
    Identify the corresponding Variation point Activity
    Check the required functionalities of the add variant activity
      If the Req_f_AV belongs to Req_f_AVP then modify resource
        Apply variant resource insertion process pattern
        Insert (variant activity)
        Update “Variation point activity“
      Else choose an another variation point to insert the add variant activity
      End if
End if
```

***c) Variation point Activity Insertion Process Pattern:***

```
If the activity is an add variation point then
  Determine the position of the insertion in the sequence flow
  Insert the required functionalities of the add variation point activity
  Apply Variant Activity Insertion process pattern// Insert a default variant activity
  Insert the flow sequence condition//The condition needed to perform the add variation point activity
  Apply the Data Insertion process pattern// Insert data
     Check resource
      If Rq_f_AV belongs to f_R then check resource availability
        If    C_nbr_A>Max_nbr_A    then
        Refuse
        Else assign resource
          If resource is a variation point then
            Apply Variant Activity Insertion process pattern// the required resource variant must be added
          Else resource becomes a variation point
          End if
        End if
    Else Apply Insertion Resource process pattern
      Calculate the number of the variation point activity and variants
        If the current number<initial number then
        Refuse // the configurable reference process model is defined by an initial number of activities (Variation point/variant)
        Else Insert (variation point activity)
        End if
End if
```

## 3.2 Example for Applying the Activity Insertion Process Pattern

In this section, we apply the Activity Insertion process pattern to ensure the transition from one version to another of a simplified version of an E-healthcare configurable process model (cf. figure2) taken from (PESOA, 2005). The process is modeled using BPMN language (OMG, 2007) and a rich-variant technique for modeling variability (PESOA, 2005). The first version of the configurable process model contains the following possible configurations:

➢ An **optional variation point activity** "Perform examination": it allows selecting different tests in parallel depending on the state of the patient.

➢ An **alternative variation point activity** "Treatment processing": it allows selecting only one variant depending on the medical report.

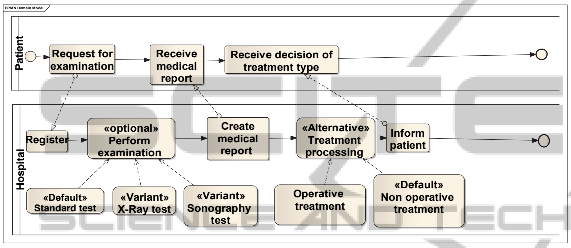

Figure 2: The e-Healthcare configurable process model before changes.

To support new requirements, the procedure of payment should be integrated to the hospitals system, to allow selection of the kind of the payment, depending on the type of treatment and patient. For this, the configurable process model should evolve by inserting new variation point "payment" which contains the following variants: the default variant "payment by cash ","social office payment support", "payment by check", and payment by card.

The following figure shows the new version of the configurable process model:

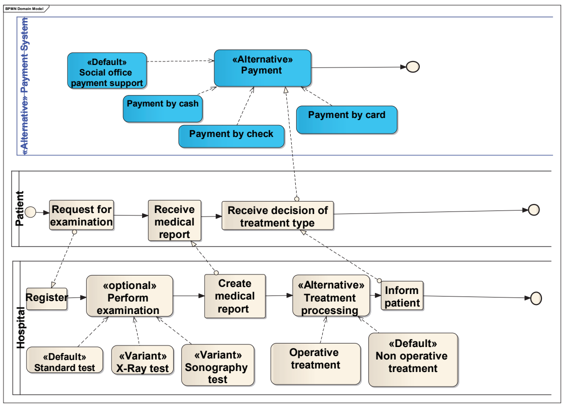

Figure 3: The e-Healthcare configurable process model after changes.

Table 4: Applying the Activity Insertion process pattern.

| **Scenario of add a variation point activity** | **Elements to check of process patterns** | **Application** |
|---|---|---|
| | Detect **change type** | Insertion |
| | Check the type of **activity change** | **-Variation point change**<br>-In this case, we apply the Variation point Activity insertion process pattern |
| | Check the nature of activity | It is a sub process |
| | List related tasks | Enter payment information, check information, validate or cancel payment. |
| | Determine the position of the insertion in the sequence flow | The variation point activity must be added after the activity "receive decision treatment". The payment depend on the type of the treatment (operative or non operative). |
| | Insert the required functionalities of the **add variation point activity** | Identification of patient, encryption of payment data, validate/ cancel a transaction. |
| | Insert the flow sequence condition | If decision of treatment type is made. |
| | Apply Variant Activity Insertion process pattern | Insert variant activities |
| | Insert data | Apply the Data Insertion process pattern |
| | Check resource | we have to apply resource insertion process pattern to insert new resource "payment system" with the following variants depending on selected configuration: "bank system", "social office system", "payment center system" |
| | Insert (**add variation point activity**) | **Variation point** "payment" |

In this section we have presented an overview of the approach which allows design decision made before changing configurable process models. At the end, we have detailed and illustrated by an example, the activity insertion which invokes resource and data evolution.

## 4 CONCLUSIONS

In this paper, we have presented a process patterns based approach, which allows guiding evolution of configurable process models for a basic evolution. The proposed process patterns system aims to lead evolution of activity, data, and resource elements (variation point/variant). We have presented only one pattern of the proposed patterns, the Activity Insertion Process Pattern. Our current work focuses on the validation of the proposed process patterns system by developing a prototype which allows automation of the proposed approach and an extension of the proposed approach to support other adaptation techniques (specialization and aggregation). Moreover, an evolution metamodel will be integrated to the prototype, in order to describe and preserve traceability of changes applied to configurable process models.

## REFERENCES

Ayora, C., "Modelling and Managing Variability in Business Process Models", Polytechnic university, Valencia, Spain, 2011.

Becker, J., Delfmann, P., Knackstedt, R. "Adaptive Reference Modelling: Integrating Configurative and Generic Adaptation Techniques for Information Models", *Efficient Information Systems Design Through Reuse of Information Models*, Springer, 2007.

BPMI.org., Business Process Modeling Notation 2.0 Beta2. Object Management Group, 2011. http://www.bpmn.org.

Conte, A., Fredj, M., Hassine, I., Giraudin, J-P., Rieu, D., "A tool and a formalism to design and apply patterns", *IEEE OOIS'02*, published in Lecture Notes in Computer Science, 2002.

Dadam, P., Reichert, M., "The ADEPT project: A decade of research and development for robust and flexible process support - challenges and achievements", *Computer Science - Research and Development*, 2009.

Gottschalk, F., "Configurable Process Models", PhD thesis, Eindhoven University of Technology, Netherlands, 2009.

Groner, G., Wende, C., Boskovic, M., Parreiras, F. S., T.Walter, F. Heidenreich, Dragan Gasevic, Steen Staab, "Validation of Families of Business Processes", *Springer*, 2011.

Hallerbach, A., Bauer, T., Reichert, M., "Correct Configuration of Process Variants in Provop", Technical Report, university of Ulm, Germany, 2009.

Jaccheri, M. L., Conradi, R. , "Techniques for process model evolution in EPOS", *IEEE Trans. Software Eng.*, 1993.

La Rosa. M., "Managing Variability in Process-Aware Information Systems", PhD Thesis, Queensland University of Technology, Brisbane, Austria, 2009.

Li, C., "Mining process model variants", Thesis, university of Twenty, Nertherlands, 2010.

Lu, R., Sadiq, S., Governatori, G., "On managing business processes variants", *Data & Knowledge Engineering*, 2009.

Muller, M., Reichert, J. Herbst, "A new paradigm for the enactment and dynamic adaptation of data-driven process structures", *CAiSE'08*, Springer, 2008.

Puhlmann, F., Schnieders, A., Weiland, J., Weske, M., "Variability Mechanisms for Process Models", *PESOA-Report*, Germany, 2005.

Razavian, M., Khosravi, R.,"Modeling Variability in Business Process Models using UML", In S. Lati, editor, *Proceedings of the 5th International Conference on Information Technology*, 2008.

Reinhartz-Berger, I., Soffer, P., Sturm, A., "Extending the Adaptability of Reference Models", *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, 2011.

Rolland, C., Nurcan, S., "Business Process Lines to deal with the Variability", *International Conference on System Sciences (HICSS)*, Hawaii, USA, 2010.

Rosemann, M., Van der Aalst, W.M.P., "A Configurable Reference Modeling Language", Information Systems, 32(1):1–23, 2008.

Sbai, H., Fredj, M., .Kjiri, L., "A process pattern for managing evolution of configurable process models", *International IEEE CIST'12*, Fez, Morocco, 22-24 October, 2012.

Schnieders, A., Puhlmann, F., "Variability Mechanisms in E-Business Process Families", *Proc. International Conference on Business Information Systems*, Klagenfurt, Austria, 2006.

Van der Aalst, W.M.P., Ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A. P., "Workflow patterns", *Distributed and Parallel Databases*, 14(1):51, 2003.

Van der Aalst, W. M. P., Dumas, M., Gottschalk, F., ter Hofstede, A. H. M., La Rosa, M., and J. Mendling, "Correctness-Preserving Configuration of Business Process Models", *FASE'08*, Held as Part of ETAPS'08, pages 46{61, 2008., 2009.

Weber, B., Reichert, M., Rinderle-Ma, S., "Change patterns and change support features - enhancing flexibility in process-aware information system", *Data and Knoweldge Engineering* 66(3), 2008.

Weber, B., Sadiq, S., Reichert, M.: Beyond rigidity - dynamic process lifecycle support: A survey on dynamic changes in process-aware information systems. *Int. Journal of Cooperative Information Systems (IJCIS)* 23, 2009.