Towards a Framework for Business Process Models Reuse

Wassim Derguech

DERI, Digital Enterprise Research Institute,
National University of Ireland, Galway
firstname.lastname@deri.org
http://www.deri.ie

Abstract. Despite there exist different proposals for reusing existing well-designed artifacts for process modelling and capturing variability in business process models, most of them suffer from three major shortcomings: lack of automation support in (i) maintaining the configurable process model, (ii) modelling sub-processes as stand alone entities and (iii) user-centricity and decision support in choosing the suitable alternatives. In this context, we propose a framework that allows for reusing process models. We present in this paper our ongoing research in defining this framework that contains a data structure and its construction principles as well as first thoughts about maintaining it when adding process variants.

Key words: Business process, business process modelling, configurable process modelling.

1 Introduction

Process Aware Information Systems (PAISs) [1] are used to manage and execute operational processes involving people, applications and data sources on the basis of business process models. The discipline that is concerned by this process-centric trend is known as Business Process Management (BPM) [2]. BPM includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes [2]. The basis of BPM is the explicitness of business processes presented by process models.

A business process model is a graphically designed network based on symbols that represent procedures or activities [3] which collectively describe how an organization conduct its business processes .

Business process modelling phase, requires intensive knowledge related to both (i) the business domain and (ii) the modelling language. In fact, the business modeler has to define exactly what tasks must be captured, their execution logic as well as how to specify them through a modelling language. Designing high quality process models from scratch is often time consuming, error-prone and costly [4]. Thus, sharing and reusing business process models by different business actors (e.g., modelers, managers, developers) is highly recommended.

1.1 Motivation

The advent of Reuse-Oriented Development (ROD) in BPM brings a number of reference models and repositories of process templates. They are used to design business process models exploiting proven practices [5, 6]. The benefits of reuse in PAISs are widely accepted in theory and practice [7]. The idea is to generate a process model from already well-designed artifacts. Yet, proposed solutions have not been proved as effective as expected. They are either not supporting enough automation to meet frequent changes in process variants, or they require advanced modelling experience. In other words, they are not applicable without strong modelling language background or support. In the whole paper, we use user experience to refer to modelling language skills.

In addition, configurable process models enable to capture variants of large process models. However, they do not allow for modelling sub-processes within these large models. A customizable process model that manages both processes and sub-processes is needed.

At a first step of this research, we have identified the following requirements for reusing business process models:

1. *Resource efficiency*: First, there is a need to have an approach that enables managing process variants in an efficient way. By efficient way we mean overtaking both resource redundancy and inconsistency problems.
2. *Maintainability*: Second, an approach should support automatic changes. By support automatic changes we refer to adding or removing a variant from the configurable process model with an automation support.
3. *Modularity*: Third, an approach should allow for modelling both processes and their sub-processes.
4. *User-centricity*: Fourth, an approach should not require much modelling skills from the end users.

1.2 Research Problem

The research goal we are trying to reach in this thesis is summarized as follows:

> *Develop an efficient approach that allows for reusing existing process models without (or with little) modelling experience. Our approach should offer a configurable model, provide an automation support for its maintainability and allow for modelling both processes and sub-processes.*

To manage configurable process modelling in an efficient way, we suggest providing a framework that captures the different process variants. A process variant, as it is understood here, is a single process model that achieves a given goal. Such a framework should be easily updated via an automation support. It should also enable business process modelers to define the most suitable (sub)process variant that suits their requirements.

1.3 Outline

The rest of this paper is divided into three sections. Section 2 shows how current solutions handle business process model reuse. Section 3 provides our preliminary work in defining a framework to represent a configurable process model. And finally section 4 concludes the paper and gives brief overview of our future work.

2 Related Work

Several solutions have been investigated to support the reuse of business process models. We classify these solutions into two categories: (i) discovery of existing models and (ii) adaptation of a structure that captures possible model's variants.

2.1 Discovery of existing models

Under this first category, we distinguish two possible approaches. The first one consists of discovering an entire business process model by querying a repository and selecting the most suitable one [8–10]. This method suffers from resource redundancy because it does not consider common parts of process model's variants and consequently they are duplicated in each entry of the repository. In addition, if any changes must be applied on the common parts of process variants, all the entries must be updated and from this arises the inconsistency problem.

The second approach overcomes the problem of resource redundancy and inconsistency by considering business process building blocks instead of entire models. This approach consists of discovering these building blocks and aggregating them in order to construct a business process model [11]. However, modelling experience is needed. In fact, the modeler has to be familiar with these business process building blocks and manage their aggregation for creating a complete model.

2.2 Adaptation of a structure that captures possible model's variants

Here we consider solutions that aim at merging different variants of a business process model into a single model. This is suitable for avoiding both resource redundancy and inconsistency problems. Under this category, we distinguish two possible approaches.

The first one provides a process model which contains placeholders that need to be detailed during the modelling phase [11]. These placeholders represent places where differences between process variants occur. During the modelling phase, users need to rely on their modelling experience to refine the placeholders.

The second approach manages configurable business process models [12]. A configurable model is the result of merging model's variants into a single model. In this case, the modelling phase consists of enabling or disabling different branches of the configurable process model.

This approach has been extended by "Questionnaire-driven Configuration of Reference Process Models" [13]. Here authors introduced a user-friendly method for customizing configurable process models. In fact, the user specifies his business requirements by answering a set of domain-related questions that are mapped to the configurable process model.

However, a process modeler has to define the configurable process model and sit with the domain expert in order to define domain constraints and their mapping to the configurable model. Thus, adding a new variant in this framework, requires another meeting between the modeler and the domain expert in order to add a variant to the process model, define its mapping with the domain constraints and update the questionnaire model. Even if this solution is suitable for a user-centric modelling, it needs to be improved to support automation maintainability of both configurable process model and questionnaire model.

In addition, configurable process models presented in [12, 13] enable to capture variants of large process models. However, they do not allow for modelling sub-processes within these large models.

3 A Framework for Managing Process Variants

To cope with the aforementioned research problem, we introduce a framework to represent configurable process models. This framework consists of a data structure, a set of structuring principles and maintaining operations. At this stage we show how to maintain this data structure only when adding a new variant. In this first version of our study, we consider a business process model as a sequence of business goals.

3.1 Data structure Definition

Our framework defines a data structure as a tuple CPM=$\{\Sigma,\Gamma,\Delta\}$ where:

- Σ represents the set of business goals involved in the whole configurable business process model,
- Γ represents the set of abstract business goals (will be introduced later in this paper) and
- Δ represents the possible variants of each business goal that can be a sequence of business goals. Entries of Δ are presented as:
  $BusinessGoal_1 : BusinessGoal_2(-BusinessGoal_N)^*$ such that
  $\{BusinessGoal_1, BusinessGoal_2, ...BusinessGoal_N\} \subseteq \Sigma \cup \Gamma$
  and "$BusinessGoal_2 - BusinessGoal_3$" is the sequence between the two business goals (i.e., $BusinessGoal_2$ and $BusinessGoal_3$).
  This means that possible variants of BusinessGoal1 are presented as a sequence of other business goals. We call $BusinessGoal_1$ a *variation point* and $BusinessGoal_2(-BusinessGoal_N)^*$ a *variant*.

The framework should respect a set of constraints/principles to assure that the data structure will remain valid and well-formed after an update operation.

1. *Minimality*: Each element of Δ has to be defined only once and should not be derived from other elements of Δ.
2. *Coverage*: By necessity and nature, the framework must cover all defined variants.
3. Consistency: Only defined variants should be deduced from the framework.

3.2 Maintaining the data structure when adding a new variant

When building the data structure, we start from a set of variants and we add them, one at a time, assuring that the principles defined previously are not violated. When adding a new variant of a goal, the idea is to check, using matching detection, whether the variant (or parts of it) already exists in the data structure. There are three situations that may occur according to the matching degree between business goals of the new variant and those in the configurable process model.

1. *Perfect match:* In this case the current variant to be inserted is entirely found in the current data structure. In such situation there is no action to be taken and the data structure remains as it.
2. *No matching:* In this case the current variant to be inserted is not found in the current data structure, not even partially. The variant is inserted as follows: all business goals composing the variant are added to Σ and the variant description is added to Δ.
   Example: We want to insert the variant A:G-H-I in $CPM_1 = \{\Sigma_1,\Gamma_1,\Delta_1\}$ where:
   - $\Sigma_1 = \{A, B, C, D, E, F\}$
   - $\Gamma_1 = \{\ \}$
   - $\Delta_1 = \{A:B\text{-}C, A:F\text{-}E, C:D\text{-}E\}$

   The updated configurable process model is then $CPM_1=\{\Sigma_1,\Gamma_1,\Delta_1\}$ where:
   - $\Sigma_1 = \{A, B, C, D, E, F, G,H,I\}$
   - $\Gamma_1 = \{\ \}$
   - $\Delta_1 = \{A:B\text{-}C, A:F\text{-}E, C:D\text{-}E, A:G\text{-}H\text{-}I\}$

   Using a tree representation, the variant is inserted as an alternative of the variation point as shown in Fig. 1[1].

3. *Partial match:* An intermediary situation is when a partial match occurs between the process variant to be inserted and the current data structure. In this case we distinguish two possible situations:

   - The first situation occurs when the new variant has common parts with another variant of the same business goal. A typical example is depicted in Fig. 2. This example shows adding the variant A:B-H-I in $CPM_1 = \{\Sigma_1,\Gamma_1,\Delta_1\}$ where:
     - $\Sigma_1 = \{A, B, C, D, E, F, G\}$

[1] All figures in the paper are following the BPM notation: www.bpmn.org

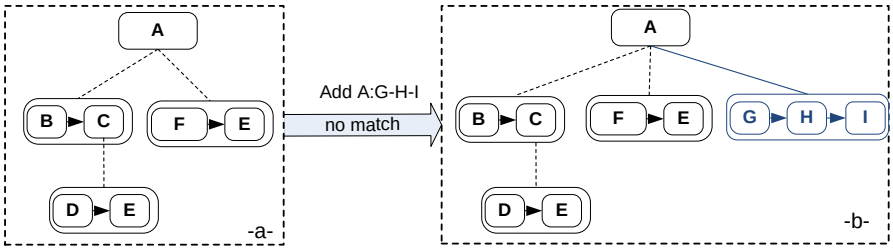

Fig. 1. The data structure (-a-) before and (-b-) after the insertion of a variant

- Γ_1 ={ }
- Δ_1 = {A:B-C, A:F-G, C:D-E}

The new variant A:B-H-I has B in common with A:B-C. To add this variant, an abstract business goal α is introduced to replace the different parts of these variants. The updated configurable process model is then CPM_1={Σ_1,Γ_1,Δ_1} where:

- Σ_1 = {A, B, C, D, E, F, G, H, I}
- Γ_1 ={ α}
- Δ_1 = {A:B-α, α:C, α:H-I,A:F-G, C:D-E}

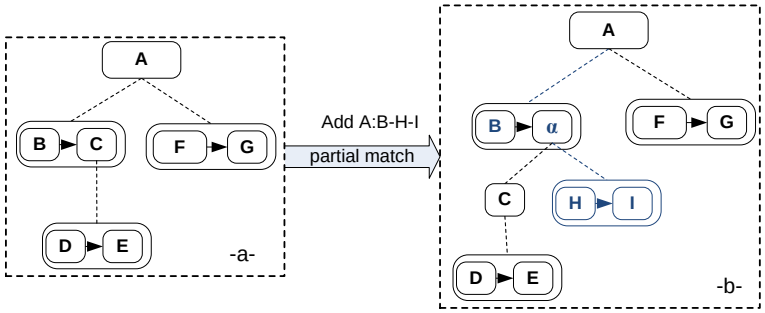

Fig. 2. The insertion of a variant including an abstract business goal

– The second situation occurs when the new variant has common parts with another variant but not of the same business goal. A typical example is depicted in Fig. 3. This example shows adding the variant A:D-F in CPM_1 = {Σ_1,Γ_1,Δ_1} where:

- Σ_1 = {A, B, C, D, E}
- Γ_1 ={ }
- Δ_1 = {A:B-C, B:D-E}

This situation is similar to the second case (no match) and the updated configurable process model is then CPM_1={Σ_1,Γ_1,Δ_1} where:

- Σ_1 = {A, B, C, D, E, F}

- $\Gamma_1 = \{\}$
- $\Delta_1 = \{A{:}B\text{-}C,\ B{:}D\text{-}E, A{:}D\text{-}F\}$

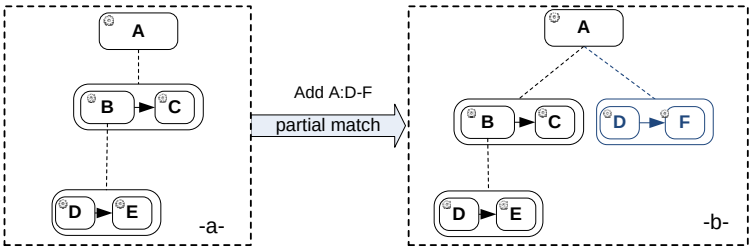

Fig. 3. The insertion of a variant with partial match without introducing an abstract business goal

4 Conclusion and future work

Reusing process models is an important concept for Business Process Management because it can decrease the modelling time and reduce the business user's work and risk to make errors.

Several solutions have been provided for process model reuse. Despite their benefits, they have not yet been adopted to capture configurable process models. This is mainly due to the issues such as the lack of automation support in handling changes (i.e., adding or removing process variants).

In this thesis we propose a framework for managing configurable process models. It defines a data structure that captures process model variability at the business goal level. We presented a set of principles that the proposed data structure has to comply with and we show how it is maintained when adding a new variant.

Our work is still in an early stage and continuous improvements are planned as a future work:

- In the near future, we plan to formally define construction principles. Further construction principles could be considered as well.
- A number of maintaining operations have not yet been explored or are still under definition, for example the deletion of a variant.
- We intend to investigate and extend this framework in order to consider other block patterns (we have presented only sequence pattern in this paper).
- From exploitation point of view, we intend to capture configuration parameters at each variation point. These parameters will help for generating a question flow to assist users in defining their process models.
- Our framework is not exclusively designed for managing process variability. We plan to experiment our approach in the context of Mashups applications development.

Acknowledgments

This thesis is supervised by Dr. Sami Bhiri from the Service Oriented Architecture Unit, National University of Ireland Galway, Digital Enterprise Research Institute. This work is funded by the Lion II project supported by Science Foundation Ireland under grant number 08/CE/I1380.

References

1. Dumas, M., van der Aalst, W.M., ter Hofstede, A.H., eds.: Process-Aware Information Systems : Bridging People and Software through Process Technology. Wiley-Interscience, Hoboken, NJ (2005)
2. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2007)
3. van der Aalst, W.M.P.: Business process management. In Liu, L., Özsu, M.T., eds.: Encyclopedia of Database Systems. Springer US (2009) 289–293
4. Indulska, M., Green, P., Recker, J., Rosemann, M.: Business process modeling: Perceived benefits. In Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M., eds.: ER. Volume 5829 of Lecture Notes in Computer Science., Springer (2009) 458–471
5. Fettke, P., Loos, P.: Multiperspective evaluation of reference models - towards a framework. In Jeusfeld, M.A., Pastor, O., eds.: ER (Workshops). Volume 2814 of Lecture Notes in Computer Science., Springer (2003) 80–91
6. Fettke, P., Loos, P., Zwicker, J.: Business process reference models: Survey and classification. In Bussler, C., Haller, A., eds.: Business Process Management Workshops. Volume 3812. (2005) 469–483
7. Fettke, P., Loos, P.: Classification of reference models: a methodology and its application. Information Systems and E-Business Management **1**(1) (January 2003) 35–53
8. Lu, R., Sadiq, S.W.: On the discovery of preferred work practice through business process variants. In: Proceedings of Conceptual Modeling - ER 2007, 26th International Conference on Conceptual Modeling. (2007) 165–180
9. Markovic, I., Pereira, A.C.: Towards a formal framework for reuse in business process modeling. In Ter, Benatallah, B., Paik, H.Y., eds.: 2nd ed. of proceedings of the Workshop Advances in Semantics for Web services 2007 (semantics4ws 2007); contained in BPM 2007 Workshops, Brisbane, Australia, Springer Verlag (2008) 484–495
10. Markovic, I., Pereira, A.C., Stojanovic, N.: A framework for querying in business process modelling. In Bichler, M., Hess, T., Krcmar, H., Lechner, U., Matthes, F., Picot, A., Speitkamp, B., Wolf, P., eds.: Multikonferenz Wirtschaftsinformatik, GITO-Verlag, Berlin (2008)
11. Becker, J., Delfmann, P., Knackstedt, R.: Adaptive Reference Modeling: Integrating Configurative and Generic Adaptation Techniques for Information Models. In: Reference Modeling. Physica-Verlag HD (2007) 27–58
12. Rosemann, M., van der Aalst, W.M.P.: A configurable reference modelling language. Inf. Syst. **32**(1) (2007) 1–23
13. La Rosa, M., Lux, J., Seidel, S., Dumas, M., ter Hofstede, A.H.M.: Questionnaire-driven configuration of reference process models. In: Proceedings of the 19th International Conference on Advanced Information Systems Engineering, CAiSE. (2007) 424–438