BPL-Framework 2.0: Support tool for Creation and Instantiation of Business Process Lines

Delacyr Almeida Monteiro Ferreira [2]   Débora Maria Barroso Paiva [1]*   Maria Istela Cagnin [1]

[1] College of Computing (Facom), Federal University of Mato Grosso do Sul (UFMS), Campo Grande, MS, Brazil

[2] Faculty of Exact Sciences and Technology (FACET), Federal University of Grande Dourados (UFGD), Dourados, MS, Brazil

Abstract

The use of software reuse techniques in the context of business processes, such as Business Process Lines (BPL), makes viable the efficient reuse of business process models. In this context, it is highlighted the importance of computational support to aid the creation, instantiation and evolution of BPLs, once that the models created are complex due to the domain inherent characteristics and the dynamism of the business. The objective of this paper is to present an evolution to the BPL-Framework 1.0, tool responsible for creating BPLs, to also include the instantiation of this kind of line. The tool facilitates the resolution of variability throughout the instantiation and generates the instance automatically without human intervention.

1 Introduction

The modeling of business processes aids in the comprehension and optimization of existing business processes, and also in the conception of new business processes to make organizations more competitive and efficient [1]. The resultant models from the business process modeling also provide help in the Engineering Requirements [2] once that they help in the comprehension of the business, and they aid the selection of functional and non-functional requirements for the software system adequate to attend the organization.

Even though the modeling of business processes is important, many organizations opt not to adopt this practice due to time and costs to execute it. The reutilization of business process models makes it possible to reduce the time and effort put in the elaboration of this kind of artifact, in addition to improve its quality, once they were already tested [3]. Under this perspective, the usage of software reuse techniques in the context of business processes is a mechanism to make viable the efficient reuse of business process models, such as Business Process Lines (BPL).

The BPL represents a set of closely related business processes with variabilities in their characteristics and operational contexts, instead of a single business process [4].

Basically, a BPL is composed by the following artifacts [5][6]: DOPs (business process models of the domain); a variability model where all the variabilities of all business processes that compose the BPL are represented; a Business Process Model *Template* (BPMT), which represents the flow of the processes, as well as the location of the variation points, which represent the subject of the variability [7] (for example, the color of the cars from a car maker) and its variants which represent the object of the variability [7] (for example, silver, black and white are the color options for the cars of the car maker); and finally a mapping between the variability model and the BPMT, representing the traceability between them.

The computational support for the creation, instantiation and evolution of BPLs is primordial once that the created models are complex due to the characteristics inherent to the domain and dynamism of the business. In this context, Terenciani et al. [6] developed a tool, named BPL-Framework 1.0, which is an Eclipse plugin that supports the creation and documentation of BPLs, in accordance to the approach of Management (creation, instantiation and evolution) of Business Process Lines (MBPL) [8]. This tool uses the notations of feature model [9], BPMN [10] and BPMN* [11], and manages the variability of the BPMN basic elements (activity, task and data object). Except that the BPL-Framework 1.0 does not support the instantiation of BPLs.

Two other tools were found to support the creation and instantiation of BPLs, whose instances are business process models. In one of them [12], the resolution of variabilities, during the instantiation of the line, is based on a questionnaire and the other [13] is done by the selection of graphic elements that represent the variabilities. However, both tools do not generate in a completely automatic way an instance of the line and also do not represent variability in elements of the data object type and do not offer vast validation in relation to the correctness of the artifacts of the line.

In face of the above mentioned lacks, it was noted the opportunity to evolve the BPL-Framework 1.0 to also include the instantiation of BPL, since it has compliance with an approach of management for this kind of line and already supports the creation of BPLs based on consolidated notations both for representation of variabilities and representation of business process models. This way, the objective of this work is to present the BPL-Framework 2.0, which is an evolution to the BPL-Framework 1.0, that also include the instantiation of BPLs. The results of a performed evaluation

*Financial support by Fundect (T.O. nº 219/2014 and 102/2016)

DOI reference number: 10.18293/SEKE2017-080

showed that the tool satisfactorily meets the quality requirements of a software product from the ISO/IEC 25010 [14]. The tool also offers support to the resolution of variability of the BPL by selection, generates automatically an instance without the need of human intervention and validates the artifacts of the line.

2 BPL-Framework 1.0

BPL-Framework 1.0 is a computational tool to support the creation and documentation of BPLs. It supports the documentation of the BPL, since it allows the elaboration of its artifacts, by means of a computational tool, such as DOPs in the BPMN notation [10], variability model in the feature model notation [9], and BPMTs in the BPMN* notation [6]. The BPMN* is an extension to the BPMN notation to represent variabilities on activities, tasks and data objects of business process models; and it also gives control over traceability between the feature model and the BPMT.

The BPL-Framework 1.0 was developed as a open-source plug-in for the Eclipse IDE. It was composed by the BPMN2 Modeler plugin [15], for the elaboration of DOPs, and by the FeatureIDE plugin [16], for the elaboration of the feature model.

The BPMN2 Modeler plug-in is composed by the Graphiti plug-in [17], which is a graphical structure for the development of diagram editors, and by the metamodel generated by the Eclipse Modeling Framework (EMF) [18]. This metamodel is created and defined in the *ecore* format, which is basically a subset of class diagrams. From the *ecore*, it is possible to generate Java code compatible with the specifications from the BPMN 2.0.2 notation [10].

Throughout the development of the BPL-Framework 1.0, the BPMN2 Modeler was extended to support the BPMN* notation such that the tool could aid the elaboration of the BPMT. For that, five new attributes were added to the classes of the *ecore* from the EMF to make configurations of variability: i) *IsVarpoint*: represents variation points in the BPMT; ii) *IsVariant*: represents the variants in the BPMT; iii) *VarPointType*: represents types of variation points in the BPMT (*AND, XOR* and *OR*); iv) *FeatureId*: enables traceability between the feature model and the BPMT; v) *FeatureType*: represents optional and mandatory elements in the BPMT (optional, mandatory and none), according to the features configurations used by the FeatureIDE.

3 BPL-Framework 2.0

The BPL-Framework 2.0 was developed to support both creation (already included in the version 1.0) and instantiation of BPLs. During the instantiation, the business analyst must choose an existing BPL that he/she wants to instantiate and the relevant BPMTs. The tool then generates automatically a configuration model for each selected BPMT. After that, the business analyst can start the variability resolution for each configuration model.

The business analyst must analyze each variation point in the configuration model and select the necessary variants for the instantiation of the BPL according to the business process of the organization. Subsequently, the DOPs are generated based on the selected variants from configuration models considered during the instantiation of the BPL.

There can be commonalities in the configuration model that will be part of all DOPs. The green color was used to represent commonalities and variants selected of activity and sub-process types. The orange color was used to represent commonalities and variants selected of data object type.

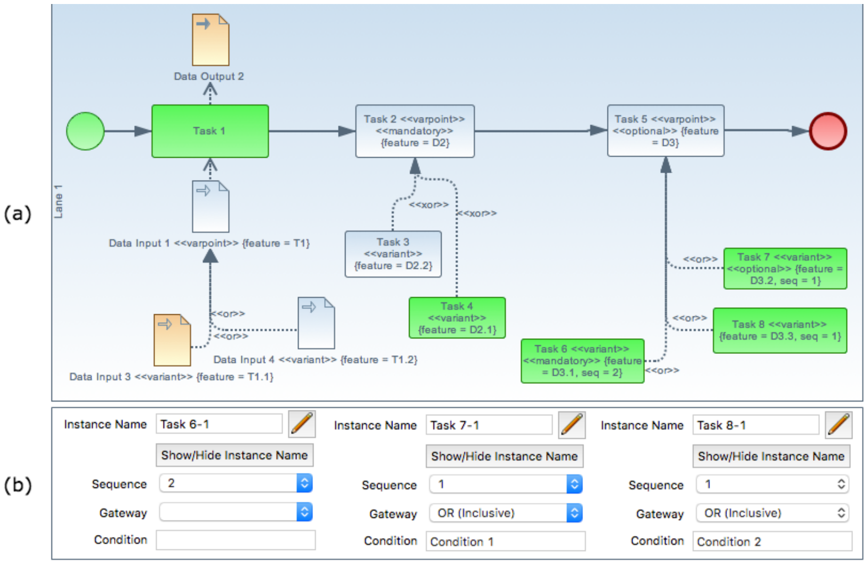
When resolving the variabilities by selecting the adequate variants to attend the objectives of the organization's business, it is possible to encounter some situations. When the variation point (Task 2 from Figure 1(a)) has one or more variants (Tasks 3 and 4 from Figure 1(a)), but its variability association has the <<*xor*>> stereotype, it indicates that only one of the variants can be selected.

Another situation is observed when the stereotype of the variability association between a variation point (Task 5 from Figure 1(a)) and its variants (Tasks 6, 7 and 8 from Figure 1(a)) is from the <<*or*>> type. When this happens, the business analyst has the possibility of selecting one or more variants from this variation point. The analyst must configure the properties for each selected variant, as shown in Figure 1(b), for the proper validation of the configuration model.

If every variant from a variation point with the <<*or*>> has the attribute *Sequence* with distinct values, the flow of these variants in the DOP generated based on the configuration model with the solved variabilities will obey the order defined by the values of this attribute. If there is a set of selected variants that has two or more equal values from the *Sequence* attribute, it is up to the business analyst to inform the type of *Gateway* that will represent this set of variants. This data can be informed by the business analyst at any time during the instantiation.

If the chosen gateway is of the XOR type (*Exclusive*), it represents a flow where only one of the paths will be taken, according to conditions to be verified. If the chosen gateway is of the *Parallel* type, it represents a flow where two or more paths can be executed in parallel. If the chosen gateway is of the OR (*Inclusive*) type, it represents a flow where there can be a combination of the paths, based on the conditions to be verified, according to the values of the *Condition* attributes. For example, the values of the *Condition* attributes from Task 7-1 and Task 8-1 illustrated on Figure 1(b).

When the variability happens on an element of the data object type with variability associations of the <<*or*>>


(a)
Lane 1
Data Output 2
Task 1
Task 2 <<varpoint>> <<mandatory>> {feature = D2}
Task 5 <<varpoint>> <<optional>> {feature = D3}
<<xor>>
<<xor>>
Task 3 <<variant>> {feature = D2.2}
Data Input 1 <<varpoint>> {feature = T1}
<<or>>
Task 7 <<variant>> <<optional>> {feature = D3.2, seq = 1}
<<or>>
<<or>>
Task 4 <<variant>> {feature = D2.1}
<<or>>
Task 8 <<variant>> {feature = D3.3, seq = 1}
Task 6 <<variant>> <<mandatory>> {feature = D3.1, seq = 2}
Data Input 4 <<variant>> {feature = T1.2}
<<or>>
Data Input 3 <<variant>> {feature = T1.1}
(b)

Instance Name	Task 6-1	Instance Name	Task 7-1	Instance Name	Task 8-1
	Show/Hide Instance Name		Show/Hide Instance Name		Show/Hide Instance Name
Sequence	2	Sequence	1	Sequence	1
Gateway		Gateway	OR (Inclusive)	Gateway	OR (Inclusive)
Condition		Condition	Condition 1	Condition	Condition 2

Figure 1. Configuration model with commonalities, variation points and variants.

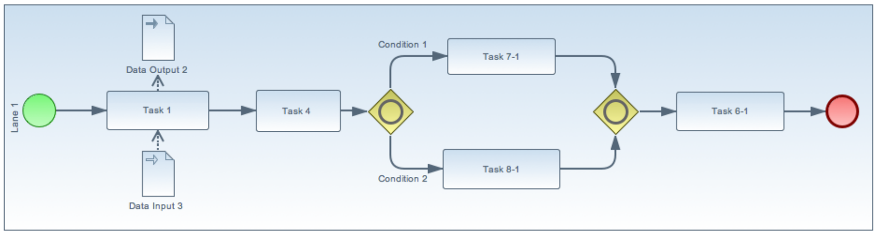


Figure 2. DOP obtained from the configuration model from Figure 1(a).

type (Data Input 3 from Figure 1(a)), there is no need to inform values for the *Sequence* attribute, once that all selected elements will be directly connected to the activity or sub-process by means of the data association during the automatic generation of the DOP.

After finishing the resolution of variabilities in the configuration models of the BPL, the business analyst may ask the tool to generate the DOP for each configuration model. Figure 2 presents the DOP generated automatically by the tool, without human intervention, based on the configuration model from Figure 1.

It is highlighted that once all variabilities were solved by the business analyst, the configuration models used for the generation of the DOPs are kept for reuse purposes on future instantiations, enabling the business analyst to take existing configuration models as a base to generate the DOP of an organization, reducing the time and costs of the instantiation of the BPL.

4 BPL-Framework 2.0 Evaluation

The objective of the evaluation conducted was to evaluate the functionalities of the BPL-Framework 2.0 through the instantiation of a BPL in the rental domain, based on: i) functional suitability sub-characteristics (accuracy), legibility and usability (ease of learning, ease of use and attractiveness) from the ISO/IEC 25010; and ii) reusability of business processes by means of BPLs instantiation supported by the BPL-Framework 2.0, by the point of view of the business analyst.

Fourteen evaluators were selected to join the evaluation of the BPL-Framework 2.0. Evaluators received a two hours training about variability, features model and essential concepts about BPL, in addition to an overview of the functionalities of the BPLs creation module of the BPL-Framework. The evaluators main task was to instantiate a BPL from the rental domain [6] based on a given description about a business process of rental of dresses. At the end of the instantiation, the evaluators answered an evaluation form, which contained a set of questions related with each defined evaluation requirement.

The data from the evaluation forms answered by the evaluators were charted and analysed. For the learnability requirement, the attribute that corresponds to the ability to understand how the tool works, 57% of the evaluators were satisfied and the other 43% were completely satisfied.

As for the attractiveness, the attribute that points to the capacity of the tool to be attractive for the user, 78,5% of the evaluators were satisfied, 14,5% were completely satisfied and only 7% of the evaluators showed to be partially satisfied. The justification from these last evaluators pointed to the low number of informative messages during the usage of the tool. Concerning the accuracy, which evaluates the capacity of the tool to obtain, with the necessary precision degree, the DOP according to the description of the business process, 50% of the evaluators were satisfied and the other 50% were completely satisfied. The reason for this is that the tool performs automatic validation of the configuration models to guarantee that all variabilities are solved before starting the generation of the DOP.

The legibility requirement allowed to evaluate the evaluators' satisfaction towards the ability of the BPL-Framework 2.0 to allow the making of legible DOPs. Some practices were added to provide more legibility while obtaining the DOPs, such as emphasizing, through the use of colors, the selected variants in the configuration model. This way, 21,5% of the evaluators were completely satisfied and 57% of the evaluators were satisfied. On this requirement, the other 21,5% of the evaluators were partially satisfied, once they reported having had to change the relative position of some elements in the obtained DOP to improve visibility, once that some elements from model automatically generated by the tool were not completely aligned with the flow.

Another requirement in the evaluation of the BPL-Framework 2.0 was related to the reusability of the business process models by means of BPL, where 78,5% of the evaluators were satisfied, 14,5% were partially satisfied and the other 7% of the evaluators were completely satisfied. The evaluators affirmed that the tool made simpler the instantiation of a BPL, since it enables the selection of the desired variability in a graphic and visual way, easing the resolution of variability. Besides that, the tool is equipped with an algorithm that automatically adjusts, in the best possible way, the position of the elements in the generated DOP to make it the most readable as possible.

5 Conclusions and Future Work

This paper presented a computational tool to support the creation and instantiation of BPLs. The evaluation performed evaluated the BPL-Framework 2.0's instantiation module. The results from this evaluation allowed to note that the tool enables the instantiation of BPLs with adequate accuracy, legibility and usability. In particular, it enables the resolution of variabilities to be done visually by the selection of variants suitable to the organization, it validates the configuration models, generates automatically an instance of the BPL, without the need of human intervention, and provides the configuration models with solved variabilities to also be reused. In the future, we plan to implement a mechanism of version control for the artifacts of the BPLs managed by the BPL-Framework 2.0, and to extend the BPL-Framework 2.0 to allow the evolution of BPLs, in addition to the integration with software assets repositories, aiming to ease the management of this kind of line.

References

[1] K. Laudon and J. Laudon, *Management Information Systems*, 12th ed. São Paulo: Prentice Hall, 2011.

[2] R. S. Wazlawick, *Object-Oriented Analysis and Design for Information Systems*, 3rd ed. Campus, 2015, in portuguese.

[3] S. A. Z. Ladeira, R. Dellosso, R. T. V. Braga, and M. I. Cagnin, "Reuse of Bussiness Modeling based on Views: A Case Study," in *Brazilian Symposium on Software Engineering*, 2008, pp. 140–155, in portuguese.

[4] N. Boffoli, D. Caivano, D. Castelluccia, and G. Visaggio, "Business process lines and decision tables driving flexibility by selection," in *International Conference on Software Composition*. Springer, 2012, pp. 178–193.

[5] G. Gröner, M. Bošković, F. S. Parreiras, and D. Gašević, "Modeling and validation of business process families," *Information Systems*, vol. 38, no. 5, pp. 709–726, 2013.

[6] M. F. Terenciani, G. B. Landre, D. M. B. Paiva, and M. I. Cagnin, "A plug-in for eclipse towards supporting business process lines documentation," in *International Conference of Computer Systems and Applications*, Marrakech, Morocco, 2015, pp. 1–8.

[7] K. Pohl, G. Bockle, and F. van der Linden, *Software Product Line Engineering: Foundations, Principles, and Techniques*. Springer-Verlag Berlin Heidelberg, 2005.

[8] G. B. Landre, "GLPN: An Approach for Business Processes Lines Management," Master's thesis, College of Computing, Federal University of Mato Grosso do Sul, Campo Grande-MS, Brazil, 2012.

[9] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-Oriented Domain Analysis (FODA): Feasibility Study," Carnegie-Mellon University, Software Engineering Institute, Pittsburgh, USA, Tech. Rep., 1990.

[10] O. M. G. OMG, "Business Process Model and Notation (BPMN)," 2013. [Online]. Available: http://www.omg.org/spec/BPMN/2.0.2/

[11] M. F. Terenciani, D. Paiva, G. Landre, and M. I. Cagnin, "BPMN* - A Notation for Variability Representation in Business Process Towards Supporting Business Process Line Modeling," in *International Conference on Software Engineering and Knowledge Engineering*, Pittsburgh, USA, 2015, pp. 1–4.

[12] F. Gottschalk and M. L. Rosa, *Process configuration*. pp 459-487, 2010, vol. Modern Business Process Automation.

[13] R. Cognini, F. Corradini, A. Polini, and B. Re, "Process variability modeling for complex organizations," in *International Conference on Enterprise Systems*, Basel, Switzerland, 2015, pp. 9–20.

[14] ISO, "Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models," International Organization for Standardization, ISO 25010:2011, 2011.

[15] M. Gille, "BPMN2 Modeler," 2013. [Online]. Available: http://eclipse.org/bpmn2-modeler/

[16] T. Thüm, C. Kästner, F. Benduhn, J. Meinicke, G. Saake, and T. Leich, "FeatureIDE: An Extensible Framework for Feature-Oriented Software Development," *Science of Computer Programming*, vol. 79, pp. 70–85, 2014.

[17] M. Wenz, "Graphiti - a Graphical Tooling Infrastructure," 2015. [Online]. Available: https://eclipse.org/graphiti/

[18] D. Steinberg, F. Budinsky, M. Paternostro, and E. Merks, *EMF: Eclipse Modeling Framework*, 2nd ed. Addison-Wesley Professional, 2015. [Online]. Available: http://www.eclipse.org/modeling/emf/