Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 471–474

DOI: 10.15439/2021F138
ISSN 2300-5963 ACSIS, Vol. 25

Business Process Recomposition as a Way to Redesign Workflows Effectively

Piotr Wiśniewski, Krzysztof Kluza, Paweł Jemioło, Antoni Ligęza
AGH University of Science and Technology
al. A. Mickiewicza 30, 30-059 Krakow, Poland
E-mail: {wpiotr,kluza,pawljmlo,ligeza}@agh.edu.pl

Anna Suchenia
Cracow University of Technology
ul. Warszawska 24, 31-155 Kraków, Poland
Email: asuchenia@pk.edu.pl

***Abstract*—Business process models are subject to changing requirements. The purpose of this paper is to present methods that enable computer-aided recomposition of process models, understood as using existing processes to design new ones. This procedure involves dividing existing BPMN diagrams into smaller components, from which new models can be created. This kind of model generation can be performed manually by the user or run automatically, based on the Constraint Programming technique. The presented algorithms can improve the process of model redesign and allow users to avoid typical anomalies that may occur in the modeling phase.**

***Index Terms*—Business Process Management, process models, BPMN, process decomposition, process composition**

I. Introduction

BUSINESS process management is a set of methods aimed at designing, analyzing, implementing, and improving task sequences performed in a specific organization. Process models are a method of knowledge representation that helps interested entities visualize and optimize implemented processes, allowing them to achieve their business goals more effectively. There exist many tools supporting knowledge management processes [1]. However, observing process industrial applications [2], many manually created models have quality defects. For this reason, computer-assisted modeling is a valuable technique to eliminate basic errors, the removal of which at the implementation stage is costlier and time-consuming. With this in mind, the work aims to present selected methods allowing to automatically generate a process model based on a diagram repository created due to the division of other models. The concept of process recomposition presented in this work is defined as a combination of decomposition of models into subprojects [3] and synthesis of process models (based on declarative specifications) [4]. The scheme of the described approach is presented in Figure 1.

The concept of decomposition of business process models is mainly used to redesign existing models or create new ones using ready-made components. One of the existing approaches is based on the identification of the largest repeating fragments [5]. The process of recomposition, however, uses formal representations of models that allow their faster and easier reconstruction. Models based on the declarative specification are equally common [6], which instead of describing the sequence explicitly, is based on the constraints and dependencies between the activities in the process.

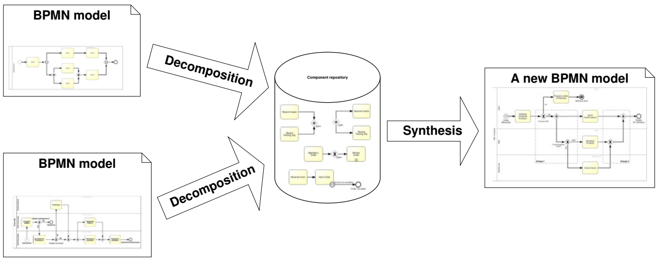

Figure 1. Schema of the idea of recomposing business process models

The useful tools in redesigning the business process are also model repositories that can be presented in the form of a database storing relevant process fragments related to metadata, such as types of objects and possible connections to other fragments [7]. In the case of the synthesis of a model composed of such fragments, it is necessary to ensure the correctness of the generated solution. For example, during automated process modeling, verification of typical anomalies such as deadlock, or loops should be considered [8].

II. Background

A. Business Process Modeling

One of the essential areas of using process models is highly developed IT systems employed in industry and business. One of the most frequently applied notations used to model business processes is BPMN (Business Process Model and Notation). Creating models of functional and technical integration of a part of the system and visualization of the processes supported by the system allows making the specification understandable for all people participating in its implementation.

B. Simple Formal Model of a Business Process

The process model decomposition method used in the presented approach is based on the representation of the BPMN diagram in the form of a directed graph. This type of data structure is considered to best suit the process model [9]. The applied method allows the existence of a loop as well as many start and end events. It can therefore serve as accurate mapping of the model created in the BPMN notation. The graph of the business process is defined by a coherent graph directed $G_P = (V_O, E_F)$, where: V_O is a non-empty set of vertices representing all flow objects, and E_F is a non-empty set of edges representing all connecting objects.

Transforming the BPMN diagram into a graph representation consists of formulating a list of all flow objects in the process along with additional information such as ID and facility name, pool/lane, object type (task, logical gate, etc.), condition (optional) and existing data units (optional). Then, all vertices of the graph should be connected with the addressed edges, to which, in the case of conditional flows (outputs from the XOR or OR gateway), a condition corresponding to a given branch in the process should be attached. A neat form of recording the business process graph is the neighborhood matrix D. Assuming that the considered process consists of n flow objects, D is a $n \times n$ square matrix in which the row and column indexes correspond to the numbers of individual objects. The individual elements of the neighborhood matrix assume a value of 1 when the two vertices are combined and zero in the other cases. Each process graph of matrix D should be accompanied by a detailed specification of flow objects described above.

III. Business Process Recomposition

As it was shown in Figure 1, the recomposition of business processes occurs in three steps: decomposing models, storing model components in a repository, and constructing new diagrams. This section describes in detail each of the phases.

A. Decomposition of process models into sub-diagrams

The business process model decomposition algorithm consists of finding all subgraphs induced by k vertices of the process graph, where $2 \leq k < n$. In the first stage, appropriate k-numerous sets of vertices defining the generated subgraphs should be determined. Then, for each subgraph, a neighborhood matrix is created by removing from the matrix D all rows and columns that do not correspond to the vertices of the selected subset. The simplest example of decomposition is the division of the process graph into all subgraphs with two vertices. From the input graph, each edge with its endpoints is extracted. A more complex problem is the generation of subgraphs consisting of three objects. This type of trigram concerning the vertex v can be defined in one of three ways:

- v with two predecessors,
- v and one predecessor and one successor,
- v with two successors.

To illustrate our method, two BPMN diagrams describing banking processes (Figures 2 and 3) were decomposed.

The first one presents in simplified the procedure for opening a savings account or a savings and settlement account, and the second describes the process of granting a loan or opening a deposit. As part of one process, it is possible to execute its main activities several times, including selecting the right service and launching it. For both models, a decomposition was performed on three-element subgraphs.

The problem of decomposition of process models can also be generalized to the generation of k-element subgraphs induced by each k vertices of the process graph. Its solution is possible by using the Constraint Programming technique with the following input data:

- decision variable: k-element vector v_{Ok} corresponding to a subset of vertices,
- constraints: all elements of the vector are different pairs and connected by edges,

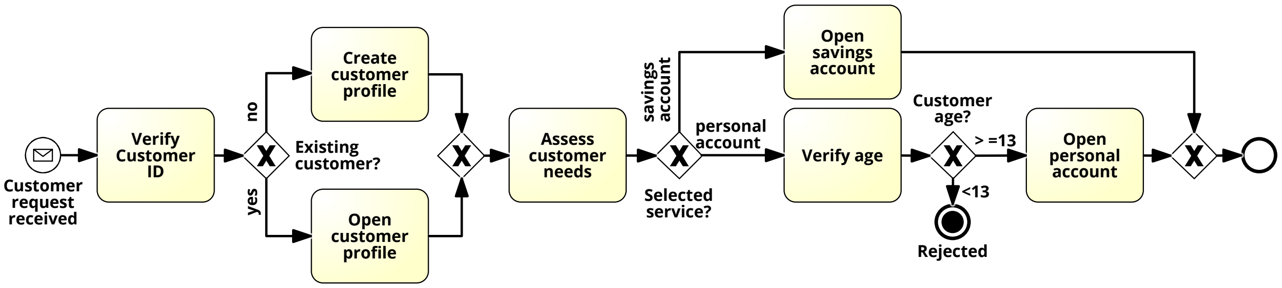

Figure 2. Business process model of opening a bank account

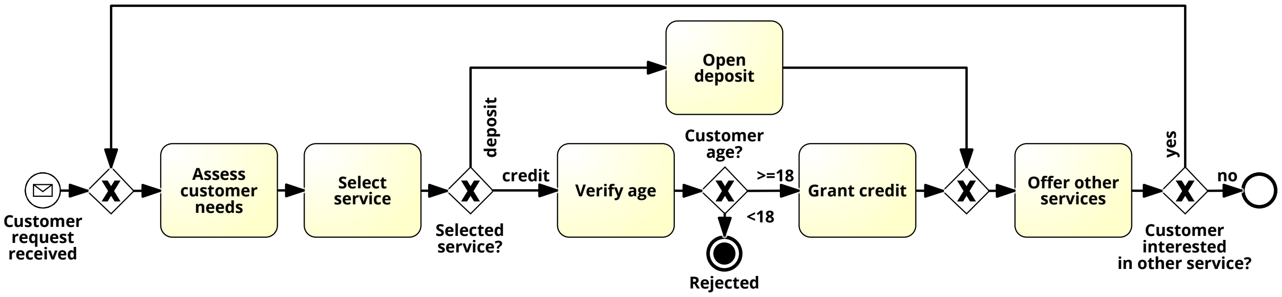

Figure 3. Model of the business process of opening a deposit and granting credit

- solutions: the adjacency matrix of a specific subgraph.

The subgraphs generated that way can be re-converted into BPMN diagrams by performing a reverse algorithm for converting the process model into a directed graph. In other words, each vertex of the subgraph may be changed to the appropriate flow object, while non-zero values of the neighborhood matrix correspond to sequential flows in the process.

Based on the defined groups, it is possible to estimate the number of potential inputs and outputs in each of the generated subgraphs by calculating the existing input and output flows of a given flow object and comparing them with the group to which the given object belongs. This number is used to determine the function $\sigma_{(I)}$, which takes 0 if there are no more entries, 1, if there is one entry, 2, in other cases.

An example of a sub-diagram generated using decomposition of the business process of opening a deposit and granting credit (Figure 3) is shown in Figure 5. It is a component of the MESE type, having many potential entries and exits.

B. The concept of a component repository

The previously described decomposition algorithm generates all possible k-element subgraphs based on the input process graph. This solution means that every possible configuration of k connected flow objects is ready for reuse in new business process models. The disadvantage of this approach is the risk of a situation in which some of the generated solutions may result in the creation of inconsistent models as a result of recomposition due to the imprecision of individual sub-diagrams. In order to ensure the correctness of the models contained in the repository and, at the same time, minimize the risk of omitting significant sub-diagrams, a method of validating the generated components based on the user's assessment was proposed. After the decomposition, a list of potentially inconsistent components is generated. For additional validation, diagrams that meet at least one of the following conditions are selected:

- split gateway without any output branch,
- merge gateway without any input branch,
- number of swimlanes greater than two.

The next step is to classify the subgraphs due to the potential number of entries and exits, as well as the similarity between them. The first of the classifiers consists of grouping models based on the values assumed by the functions $\sigma_{(I)}$ and $\sigma_{(O)}$. The second one is based on the vector technique [10], in which similarity between the two sub-systems depends on the number of common flow objects. Bearing in mind the two presented criteria, selecting the appropriate sub-diagram consists of selecting the appropriate group or specific flow objects, which should be included in the selected diagram. The format of recording sub-patterns in the repository should be conditioned by the method used to synthesize the models later. In the case of manual connection of components, it is recommended to store them in the BPMN 2.0 XML standard, which consists of saving the process model in a file containing definitions of process elements and connections between them. This representation allows you to directly load the file into the graphic editor and facilitate its subsequent modification. The automatic synthesis of processes described in this work uses a graphical representation to generate a complete model. The more practical solution is to leave the sub-diagrams in the form of graphs represented by the neighborhood matrix and the specification of the contained flow objects.

C. Composition of a process model based on the existing components

The last stage of model recompilation occurs by synthesizing BPMN diagrams based on the created subprogram subdiagrams. The first of the proposed methods is the manual creation of a process model in a graphical editor, using a ready-made component database associated with the basic modeling guidelines. In addition to storing generated subgraphs, the purpose of having such a knowledge base is to avoid typical anomalies occurring in process models [8] and reduce the need for formal verification of the final model [11]. Among modeling errors, two basic groups are distinguished:

- syntactic anomalies, such as wrong use of flow objects or swimlanes,
- structural anomalies that concern incorrect behavior of the process during execution, such as deadlocks or infinite loops.

The user interface of the graphic editor should suggest the recommended connections and protect against creating connections that may lead to incorrect execution of the process. An example of an incorrect connection is shown in Figure 6.

An illustration of the process recomposition result using the manual synthesis method is presented in Figure 4. The main assumption was to include in the process of opening a bank account (Figure 2) the possibility of opening a deposit and allowing the situation where the client orders several services within one visit. The colors are marked components extracted as a result of the decomposition of different models.

The synthesis of sub-model process models can also be carried out in an automated way, using a Constraint Programming technique. The modeling person then determines the list of tasks that should be performed as part of the process, and the task of the synthesis algorithm is to create from the available fragments a correct BPMN diagram containing these tasks and meeting the limitations resulting from the adopted notation. The necessary constraints include:

- exactly one start event,
- at least one end event,
- correct number of inputs and outputs of each object,
- closed gateway structures – no possibility of two final events at the same time.

The use of the Constraint Programming method for the modeling problem finds many solutions that meet the given conditions [12]. This approach allows the user to compare several generated models and choose the best result based on the adopted criteria.

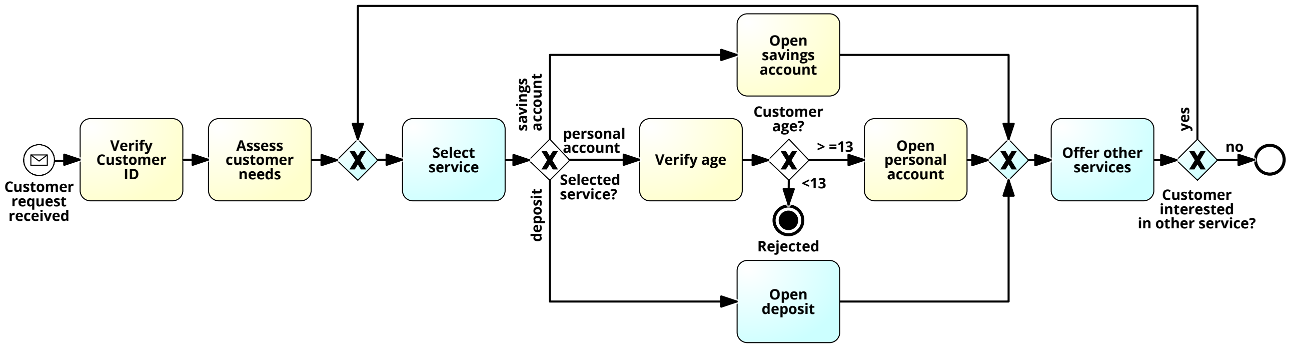

Figure 4. The result of the recomposition of two business processes in the BPMN notation

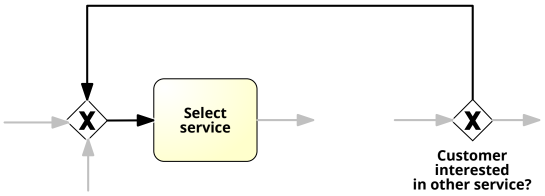

Figure 5. A fragment of a diagram created as a result of decomposition. The gray color indicates potential connections with other process components

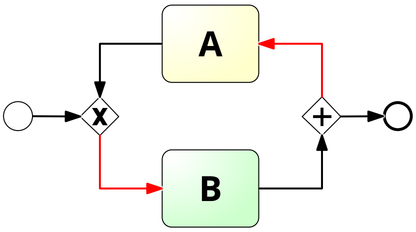

Figure 6. A potentially infinite loop

IV. Conclusions

We presented a method of automatic generation of business process models based on a repository of diagrams created as a result of fragmenting the existing models. Such an approach can be a support for modeling processes by automating the creation of models. Our method consists of synthesizing BPMN diagrams based on the defined components resulting from the decomposition of the processes. It allows a user to create a new, correct model using ready-made fragments of diagrams. An essential element of the recomposition process is the model repository, which allows a user to search for components necessary to synthesize a given diagram efficiently. Implementing a solution supporting automated recomposition of business processes may reduce the amount of time and resources used so far by the organization during process modeling.

In the future, we plan to extend the existing method of component classification in the repository and develop the possibility of synthesizing the process based on a description in natural language. In the case of automatic synthesis of models from sub-diagrams, a significant extension can support the user's decisions in selecting the diagram most suited to the actual process.

References

[1] M. Pondel and J. Pondel, "Selected it tools in enterprise knowledge management processes–overview and efficiency study," in *IFIP International Workshop on Artificial Intelligence for Knowledge Management*. Springer, 2017, pp. 12–28.

[2] H. Leopold, J. Mendling, and O. Günther, "Learning from quality issues of BPMN models from industry," *IEEE software*, vol. 33, no. 4, pp. 26–33, 2015.

[3] P. Wiśniewski, "Decomposition of business process models into reusable sub-diagrams," in *ITM Web of Conferences*, vol. 15. EDP Sciences, 2017, p. 01002.

[4] P. Wiśniewski, K. Kluza, M. Ślażyński, and A. Ligęza, "Constraint-based composition of business process models," in *International Conference on Business Process Management*. Springer, 2017, pp. 133–141.

[5] M. Dumas, L. García-Bañuelos, M. La Rosa, and R. Uba, "Fast detection of exact clones in business process model repositories," *Information Systems*, vol. 38, no. 4, pp. 619–633, 2013.

[6] F. M. Maggi, A. J. Mooij, and W. M. van der Aalst, "User-guided discovery of declarative process models," in *2011 IEEE symposium on computational intelligence and data mining (CIDM)*. IEEE, 2011, pp. 192–199.

[7] M. Skouradaki, V. Andrikopoulos, and F. Leymann, "Representative BPMN 2.0 process model generation from recurring structures," in *2016 IEEE International Conference on Web Services (ICWS)*. IEEE, 2016, pp. 468–475.

[8] A. Suchenia, T. Potempa, A. Ligęza, K. Jobczyk, and K. Kluza, "Selected approaches towards taxonomy of business process anomalies," in *Advances in Business ICT: New Ideas from Ongoing Research*. Springer, 2017, pp. 65–85.

[9] I. M. Weber, *Semantic Methods for Execution-level Business Process Modeling: Modeling Support Through Process Verification and Service Composition*. Springer, 2009, vol. 40.

[10] B. Van Dongen, R. Dijkman, and J. Mendling, "Measuring similarity between business process models," in *Seminal Contributions to Information Systems Engineering*. Springer, 2013, pp. 405–419.

[11] R. Klimek and P. Szwed, "Verification of ArchiMate process specifications based on deductive temporal reasoning," in *2013 Federated Conference on Computer Science and Information Systems*. IEEE, 2013, pp. 1109–1116.

[12] R. Chenouard, L. Granvilliers, and R. Soto, "Model-driven constraint programming," in *Proceedings of the 10th international ACM SIGPLAN conference on Principles and practice of declarative programming*, 2008, pp. 236–246.