

# Visualizing a Business Process Control-Flow

Ivaylo Kamenarov and Katalina Grigorova

**Abstract**—This paper presents the developed algorithm for visualization the business process graph structure. In business processes repository for each process it is stored data about its sub-processes. Using the sub-processes characteristics including basic data and triggering and terminating events, the described algorithm automatically creates a graphic image representing a control-flow. The sub-processes are connected via events – a terminating event for one sub-process is a triggering event for another.

**Index Terms**—Business process repository, visualizing business process, business process models.

## I. INTRODUCTION

Nowadays, a fast growing industry requires corporations to continuously adapt their activities to changing market situations, ensuring their relevance and competitiveness. Many companies collect and store business processes models in repositories. Such repositories may contain hundreds or even thousands of business process models [1]. This situation needs a flexible tool for presentation and visualization.

There are two ways of storing models in business process repository: explicit and implicit. If the models are stored explicitly, that predisposes some disadvantages for future work. For example, if two or more processes use common parts (sub-processes), then these parts should be stored in more than one place in the repository. So when some changes of any sub-processes are made, the corrections have to be made in more than one place. It is possible to come to conflict situations, when a process is stored in different places with different content. Another unpleasant situation is when a small part of a process has been changed, then it is necessary to change all the processes associated with it and to store new models for them.

To avoid such undesirable situations it is better to store implicitly models of business processes and to present them when it is necessary. This affords an opportunity different functions of a business process to be saved independently of one another. Thereby the integrity of the process is not getting destroyed while updating any particular function of it. It is needed a mechanism that recovers the whole business process by the available functions.

Very often in practice there is a set of processes, which are arranged in time in a definite order, i.e. it is not possible for one process to start before another has finished. This order is presented by some kind of relations between the processes. This suggests that in order to be completed a business

processes, together with the characteristics of each of the sub-processes it is necessary to be kept also information about the connections between them.

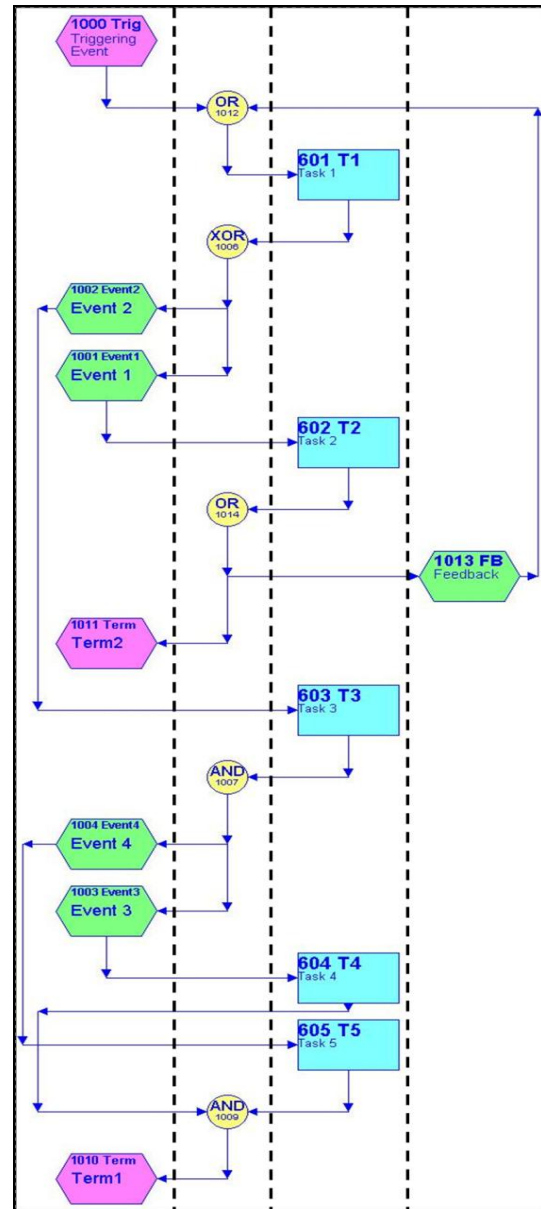


Fig. 1. Generate model splitting to four columns: Events, connectors, processes and feedback events.

## II. ALGORITHM FOR VISUALIZATION THE BUSINESS PROCESS

The authors have developed internal data model for implementation of business process repository [2]. The main idea is to store not the whole business process model, but its components only, i.e. its sub-processes at all levels. A uniform description of all processes, both simple and complex, and regardless the levels in the hierarchy, is supported.

Data about processes corresponds to EPC (Event-driven

Manuscript received July 25, 2015; revised November 26, 2016.

The authors are with the Department of Informatics and Information Technologies, University of Ruse, Bulgaria (e-mail: ikk@ami.uni-ruse.bg, kgrigorova@ami.uni-ruse.bg).

Process Chain) standard for business process modeling. The EPC standard represents the control-flows of business processes through graphical notations. Over the years, the standard has been developed and established as a powerful tool for modeling, analyzing and transforming business processes.

The basic elements of the standard are functions, events and connectors, and connections between them. But not every diagram composed of these elements is a correct EPC diagram. Functions describe activities of processes and events – preconditions and results of functions. There are three kinds of connector types including AND ( $\wedge$ ), OR ( $\vee$ ) and XOR ( $\times$ ) for the definition of complex routing rules. Connectors have either multiple incoming and one outgoing arc (join connectors) or one incoming and multiple outgoing arcs (split connectors). As a syntax rule, functions and events have to alternate either directly or indirectly when they are linked via one or more connectors. OR- and XOR-splits after events are not allowed since events cannot make decisions [3]. Control-flow arcs are used to link these elements [4].

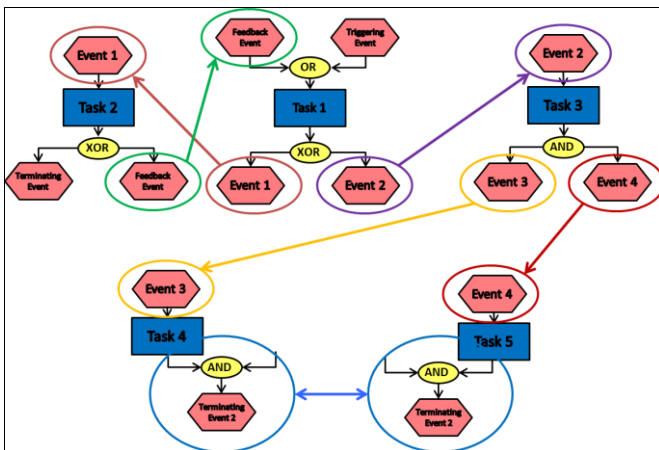


Fig. 2. Connecting processes by EPC events.

The informal semantics of an EPC can be described as follows: The AND-split activates all subsequent branches in concurrency; the XOR-split represents a choice between one of several alternative branches; the OR-split triggers one, two or up to all of multiple branches based on conditions [4]. In addition to the informal semantics are used some formal definitions for EPC standard [5].

The authors' goal is to propose an algorithm for visualization of stored in business process repository models (Fig. 1). In the repository for each process a set of its sub-processes is stored. They present the graph structure of the process model. Sub-processes are characterized by the triggering and terminating events. By means of these data a graphic image, which is a control-flow, is automatically created. As it has been mentioned, the business process repository contains description of business processes from each level. Thus, the components of the process, its sub-processes stay somewhat independent of each other. This achieves independence between the elements of the process and appears as source for further developments and improvements of these elements without affecting the already established business process models. Changes automatically, if necessary, reflect in the business process model.

In the EPC standard events are passive elements and serve

as a link between the different functions of the business process (Fig. 2). Events are divided in two groups – simple and complex. Simple events are single items and complex events combine together several events (possibly they are also complex events) through connector. Each process (function) is preceded by a triggering event and ends with a terminating event where the triggering and terminating events can also be complex. For each complex event is formed tree of events. This kind of tree contains only simple events on its leaves. If a process has a complex triggering event, by its tree of events it is possible to find all simple events, which are triggering for that process. Similarly, if a process has complex terminating event, by its tree of events all simple events, which appear as a result of process execution could be identified [5] (Fig. 4).

The algorithm's main task is to sort in such a manner the elements of graph structure so that the image is understandable. The elements of the structure include processes, their triggering and terminating events and connectors. These elements are input data for the algorithm. There are three types of components: processes, events and connectors, situated in such a manner that the developed image is expressive and clear.

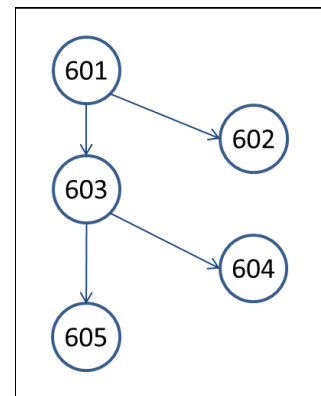


Fig. 3. New graph of processes needed for topological sorting.

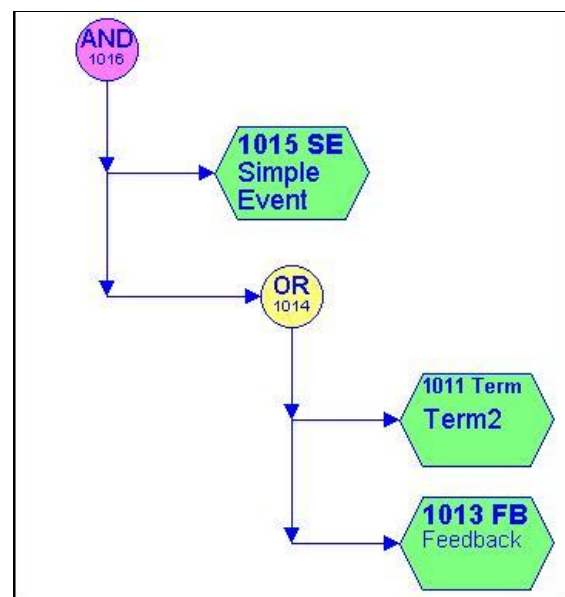


Fig. 4. Tree of events for connector (end complex event).

The arrangement of the graph structure elements forms a grid of rows and columns. Each column is intended for a different type of elements. There are three main columns:

events, connectors and processes (Fig. 1 and Fig. 5). It is possible to have a fourth column, which is intended for events of type "feedback". A "feedback" is such an event that goes back at previous stage in the control-flow of a process. For readability each element of the graph structure takes a separate row, i.e. as many are the elements in the structure as many rows appear in the image. With such a distribution of the elements a clear idea about the type of element is achieved.

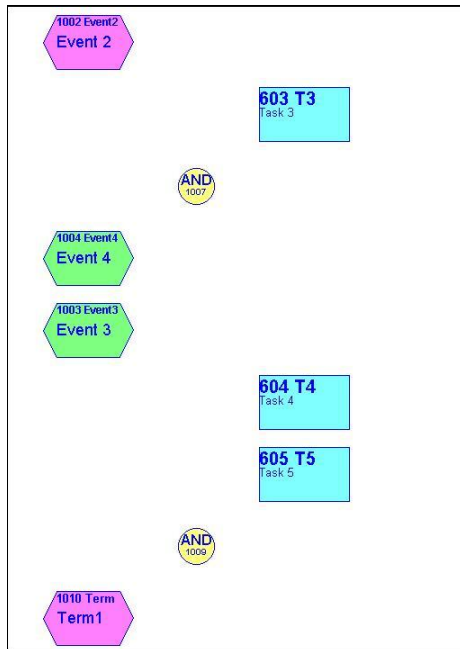


Fig. 5. Business process model after sorting elements without being connected.

In the column designed for processes appear only the processes included in the graph structure and they can be easily distinguished from the other elements, not only by the shape and color of the image, but also by their place in the whole picture. This applies also to the other kinds of elements – connectors and events.

The graphic image is a business process model and its elements within the different rows should be connected by arrows. To maximize the expressiveness and clearness of the model is necessary to arrange the elements in an appropriate manner. The processes should be ordered first, because the other elements (triggering and terminating events of the processes) depend on them.

For this purpose a new directed graph is formed in which nodes represent the processes, included in the model, and the edges are defined by the events and the connectors (Fig. 3). The connectors help to find the triggering and terminating simple events for each process [4]. If an event is terminating for one process and triggering for another process, then both processes are connected by an edge. For each simple event an edge is added, except the events which are not triggering or not terminating for any process.

Processes are topologically sorted using the built graph. As the topological sorting algorithm [6] can be applied only to acyclic graph, it is necessary the edges, corresponding to the events of type "feedback" to be removed from the graph. If it is not possible to sort the graph, then there exists at least one event, not defined as "feedback" event, but that closes a loop

in the graph. Thus it is not possible to create a graphical representation of the model, before removing this error. If the graph is not connected, it is impossible to create a model which includes all processes, because there are parts without links between them.

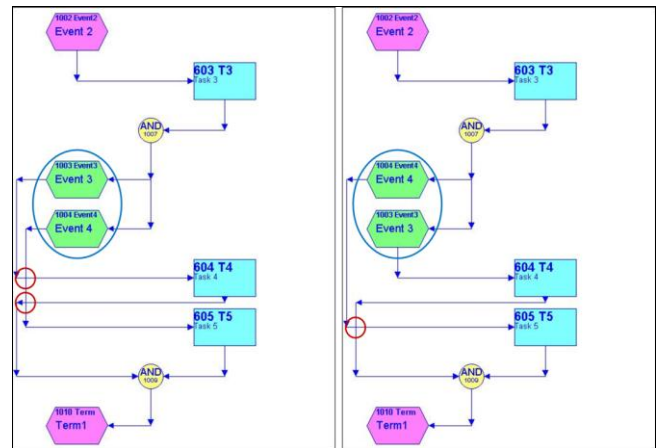


Fig. 6. Connecting processes by EPC events.

After sorting the processes, it is necessary to arrange the events and the connectors (Fig. 5 and Fig. 6). In the chain of the model between two processes must exist only one event, and between an event and a process must appear zero or more connectors [7]. In the grid rows, downward of each process are arranged its terminating events and connectors, located between the process and the terminating events. In the rows upward of each process are arranged the connectors, located between the process and its triggering events. Along with them are situated those of its triggering events that are not terminating for any other process.

An event which is terminating for one process and triggering for another process must be connected by an arrow with the process that it triggers or with the complex triggering event containing it. If terminating events for a process are more than one, they must be arranged appropriately in order to cross as fewer arrows as possible and to produce more readable image (Fig. 6). If a connector is related to more than one event (including complex event with connector) then these events are arranged in the following manner. "Feedback" events and events that are not triggering for any process are arranged at the end. All other events are arranged in descending order according to the number of the processes from the topological sorting, for which they are triggering, i.e. the first event corresponds to the last process from the topological sorting.

After the final arrangement of all the elements of the model, it is necessary to connect all of them by arrows (Fig. 6). With this the visualization of the business process model is completed.

### III. FUTURE DIRECTION

The next step of this research is to develop a method for storing and managing versions of business processes. Storing different versions of a business process supports the possibility of analyzing and selecting the most appropriate version. When reusing complete business processes for new systems, the different versions (variants) of these processes

provide a wide range of options and a possibility to choose that version, which most accurately reflects the requirements.

#### IV. CONCLUSION

The described method for visualizing business process models improves the functionality of the business processes repository. Thanks to this new algorithm business process models can be automatically rebuilt from their independently stored sub-processes. It also provides verification that a set of business processes forms a new one of a higher level in the hierarchy. It is possible quickly and easily to compare several new generated business process models. All these features facilitate the work with the models stored in the business process repository.

#### ACKNOWLEDGMENT

This work is supported by the National Scientific Research Fund under the contract ДФНИ - И02/13.

In the course of this study has been used equipment purchased under the project BG161PO003-1.2.04.-0011-S0001 "Development of applied research at the University of Ruse" implemented with the financial support of OP "Development of the Competitiveness of the Bulgarian economy 2007-2013 ", co-financed by the European Union through the European Regional Development Fund and the state budget of the Republic of Bulgaria."

#### REFERENCES

- [1] R. Dijkman, M. Dumas, B. Dongen, R. Käärrik, and J. Mendling, "Similarity of business process models: Metrics and evaluation," *Information Systems*, 2011, pp. 498–516.

- [2] K. Grigorova and I. Kamenarov, "Object relational business process repository," *Com. Sys. Tech.* '12, pp. 72-78, 2012.
- [3] Overview Event-Driven Process Chain Notation. Rules for EPC Modeling. [Online]. Available: <http://www.ariscommunity.com/event-driven-process-chain>
- [4] I. Kamenarov, "Classification of events in the EPC standard," *ISGT'13*, pp. 320-327, 2013.
- [5] W. M. P. Aalst, "Formalization and verification of event-driven process chains," Department of Mathematics and Computing Science, Eindhoven University of Technology.
- [6] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, "Section 22.4: topological sort," *Introduction to Algorithms*, 3rd ed., MIT Press, pp. 612–615.
- [7] J. Mendling, *Metrics for Process Models*, Springer, 2008.



**Ivaylo Kamenarov** was born in 1985. He is Ph.D. student at the Department of Informatics and Information Technology at University of Ruse, Bulgaria. His research interests include business process modeling, software engineering.



**Katalina Grigorova** is the head of the Department of Informatics and Information Technologies at University of Ruse, Bulgaria. She received her MSc degree in applied mathematics from Moscow Power Engineering Institute and the PhD degree in computer aided manufacturing from University of Ruse. Her research interests include business and software architectures modeling, business process modeling, automated software engineering, databases, data structures and algorithms design, Programming. Prof. Grigorova is a member of Association of Information Systems (AIS) and its Bulgarian chapter BulAIS. She is a winner of IBM Faculty Award.