

BPMN-EPC-BPMN Converter

Vladimir Kotsev, Ivan Stanev, Katalina Grigороva

BPMN-EPC-BPMN Converter: The problem for generation of new business processes partially described by BPMN, and EPC is described in this paper. The software architecture of the Business processes Generator (BPGen) is presented. Three conversion rules for BPMN – EPC transformation are defined. The Software architecture and business processes conversion is described. Two business processes managing the BPMN – EPC conversion are defined describing the forward and backward conversion operations.

Key words: Business Modeling, Business Process Generation, Automated Programming, Software Engineering, Software Architecture, BPMN, EPC.

1. INTRODUCTION

Business process (BP) modeling has become a significant part of enterprise management during the last decade. Nowadays' business modeling aims at the integration of the partial models that represent particular views on an enterprise. This means not only those models of distinctive and important parts of the enterprise should be created, but also that semantic relationships between partial models can be expressed.

The main idea of business modeling is to help people gain better understanding of complex systems by providing different appropriate abstractions. Therefore, a corresponding modeling language is used based on specific terminology and notation that is common within particular view. The best way to show the common situation is diagrams. This makes business processes and relationships between them easier to understand by wider and different circles of people within an organization.

One of today's most powerful mean of business process modeling is the Business Process Modeling Notation (BPMN). It is a graphical notation for expressing business processes in a Business Process Diagram (BPD). The objective of BPMN is to support business process management by providing a notation that is intuitive to business users yet able to represent complex process semantics.

Naturally, BPMN is not the one and only notation and it also has its disadvantages. According to most researches found on the Internet the most significant drawback of BPMN is the lack of good linking to other elements of the enterprise architecture. And here people turn back to the historically older Event-Driven Process Chains (EPC) that can strengthen this aspect of the business process modeling. Fig. 1 describes the abilities of BPMN and EPC to link different dimensions (see [4]).

In fact, EPC and BPMN can do a great job, if used together. The reason behind this

is that each of them can compensate the disadvantage of the other. In this paper we show how we try and hopefully manage to combine both EPC and BPMN to achieve our goals.

The paper is organized as follow: section 2 – Definition of the need for generation of new business processes, collaboratively presented with BPMN, and EPC; Section 3 – Description of the Business Process Generator Software Architecture; Section4 – Introduction of Conversion Rules Between BPMN and EPC; Section 5 – Presentation in more details of the BPGen conversion

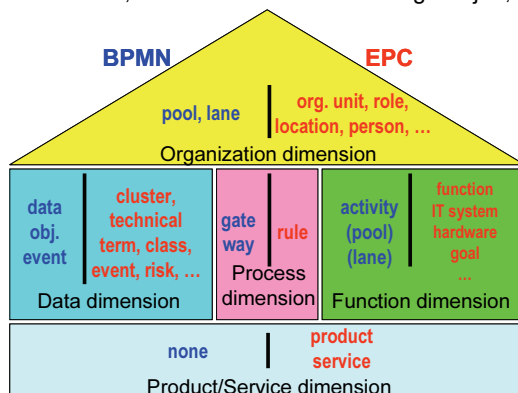


Fig. 1 BPMN / EPC Product/Service Dimensions

module Software Architecture, and the two main business processes expressing its work.

2 DEFINITION OF THE PROBLEM

Since EPC was used in business process modeling for a longer period of time than BPMN, there is a vast amount of tools that are based on the EPC standards. One such tool we worked on is a Business Process Generator (BPGen) which uses EPC rules to link different tasks and processes from a database and help us generate business processes. On the other hand Unified Modeling Language (UML) and BPMN are getting stronger and are considered as the world leading standard in the sphere of modeling as a whole.

In order to keep using our business process generator BPGen but show our business process diagrams pursuant to the newest standards and requirements we need to link our tool to some UML based modeling environment. Aiming for a continuous period of usage we have to choose such that is being updated frequently enough to follow the world standards. These requirements made us have a closer look on and eventually select Sparx Systems Enterprise Architect. To make this linking possible it is necessary to create a converter and use it as a bridge between the tools.

Creating such a tool that converts the business process and its elements from a diagram and into a database can be used not only in our particular case.

Database models of business structures are usually hidden and the user does not access them directly, but rather through an interface on the software tools for capturing and analyzing the models. In many cases the database design plays an important role for making possible the kinds of analysis that the business designer wishes to perform on the model. If the designer is authorized to have knowledge of the structure of the business model in the database, more efficient understanding, implementation and support may be possible. For example, this can allow the designer to structure the representation in such a way to facilitate faster analysis or higher degree of reuse of business process templates [1].

When the business designer needs to deal with only a small part of business model working with its database model is very convenient. There are a number of aspects to the enterprise, each giving a different view on it, presented by corresponding business structure.

3 BPGen SOFTWARE ARCHITECTURE

Most developers of business process modeling environments try to provide the users with a variety of tools that can be helpful for making modeling easier and more descriptive. For administrative processes and others, that tend to be repeatable without being changed frequently, it is enough to create a model that can be used multiple times.

However, if we think about processes in the industrial branch things are a bit different. Those processes are very dynamic because they are being modified frequently. The users need not only a toolset that allows them to change the business model, but one that can also automatically generate new processes or edit the existing ones using the already available information. For this purpose the possibility of the following modifications should be provided: (1) Inserting and editing data about processes (process properties and additional useful information); (2) Storing data about different versions of a process; (3) Inserting and editing data about the connections between processes and their sub-processes; (4) New process generating with information about its sub-processes information available; (5) Editing of existing process, which includes sub-process adding and removing; (6) Creating a potentially executable business model; (7) Defining the sub-processes of a process; (8) Printing the data about a process as an aggregation of sub-processes on all levels; (9) Printing the data about a process as a chain of sub-processes.

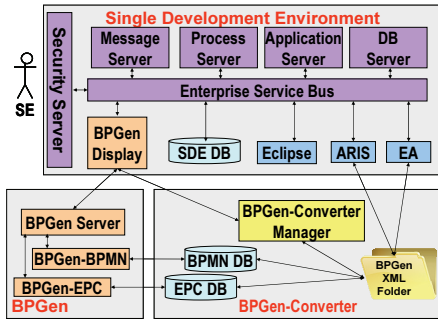


Fig. 2 BPGen Software Architecture

BPGen-BPMN, and BPGen-EPC, for generation of new processes based on respectively on BPMN, or EPC; (3) The component BPGen-Converter, including manager (BPGen-Converter Manager) for realization the conversion from BPMN to EPC, and vice versa, and support the BPMN, and EPC databases (DB) and its catalogues.

4 EPC - BPMN CONVERSION RULES

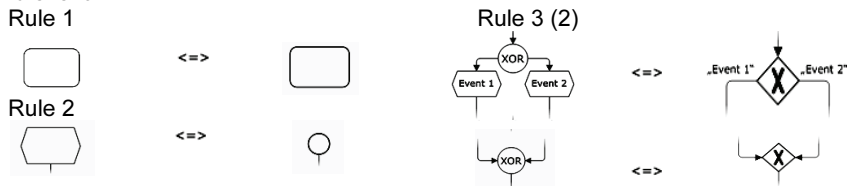
In order to convert the information about the business processes from BPMN and translate it to information suitable to fill it in an EPC database we meet the necessity to map EPC to BPMN notations first. Once mapped, the elements of both notations can be translated in either direction.

Direct mapping is the best solution for translating one of the notations to another but there are still some minor problems. According to most researches on the Internet it is almost impossible to have a universal mapping between EPC and BPMN because of some differences of their nature. It is recommended that the direct mapping is done only for the core of the notations. The rules of the mapping are described by Willi Tscheschner [5]. Fig. 3 provides the translation between the notations.

In EPC a function is described as an activity which supports the completion of a business objective. It is semantically a process rule for transforming an input system state to a following output system state. In BPMN an activity is a generic term for work that a company performs [3]. A task is specified as an atomic activity which cannot be broken down to several activities. With both definitions a function in EPC is mapped to a task in BPMN.

A process model, described in EPC, is determined by one or more events (an explicit entrance state of this process). Similar to this, a process model in BPMN is also implicitly or explicitly started through a start event. Out of this, an event in EPC which has no incoming control flows can be mapped to a start event in BPMN.

Process models defined in EPC are terminated by events (an explicit end state of this process). In BPMN a process model is also implicit or explicit terminated through an end event. With this, an event in EPC which has no outgoing control flows can be mapped to an end event in BPMN.



These are the main principles of BPGen tool. Its software architecture and interfaces are shown on Fig. 2. This tool have three main parts: (1) Single Development Environment (SDE), which include all important design components of the end user environment (in the particular example – instruments for realization of Service Oriented Architecture), and the BPGen-Display – a Graphical User Interface of BPGen; (2) The core tool called BPGen, including the components BPGen Server – responsible for the tool operations management, as well as two generators –

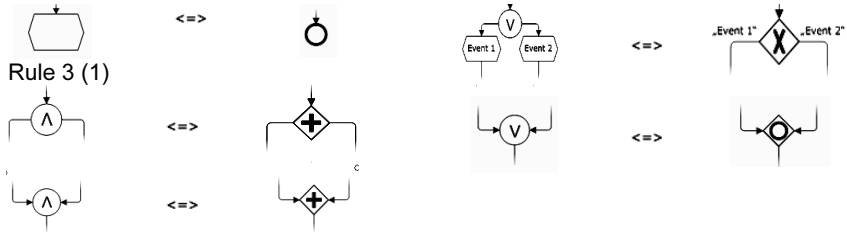


Fig. 3 EPC – BPMN Conversion Rules

Connectors (xor-connector, and-connector, and or-connector) in EPC are defined as concatenation points in the process for events and functions [2]. Gateways in BPMN are defined for controlling the divergence and convergence of the sequence flow [3]. With this, the several connectors can be mapped to the same gateway (xor-connector to data-based exclusive gateway, and-connector to parallel gateway, and or-connector to inclusive gateway) independent from the splitting or joining behavior.

In addition, if a xor-connector or an or-connector occurs which has splitting behavior (by concatenation of events) the process flow is affected by the states of the following events. Depending on the previous function and the output situation, this event gets chosen which represents the appropriated state. Similar to this, in BPMN the process flow after a gateway is determined by the condition of the following sequence flow. Therewith, those events in EPC can be mapped to the condition expression of the following sequence flows.

In future versions we desire to go even further and make the converter capable to translate the full sets of each notation to the other. However, it will be a hard task because every complex diagram can use some of the elements for its own purposes that don't obey every common regulation. It is still believed that making the conversion on a deeper level is risky and should be decided for each business process model diagram.

5 BPGen-CONVERTER SOFTWARE ARCHITECTURE

As we have the BPGen and Enterprise Architect ready to use, we also need a connection between them. We have already decided the conversion rules between BPMN and EPC elements. Since Enterprise Architect provides a diagram exporting to an XML file, we can use it as a bridge between the environments (Fig.2 - BPGen XML folder).

Our converter can receive the exported XML file of a diagram and extract the useful information. This data can directly be inserted into the BPGen database, while a backup copy of the XML file shall be saved in order to keep the information about the tasks and events that may be useful for someone else.

After the business process model is ready the converter receives an appropriate message through BPGen-Display and is allowed to access the database and extract the needed information to build the BMPN diagram up. The software architecture of BPGen-Converter is shown on Fig. 4. As you can see we've planned to create another business process generator for BPMN and

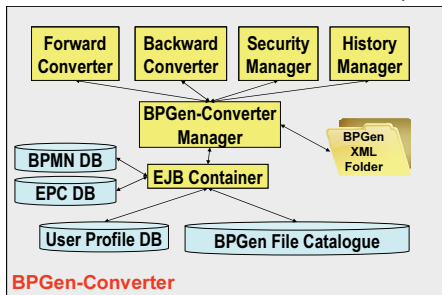


Fig. 4 BPGen-Converter Software Architecture

possibly include ARIS Toolset in our business process environments, thus making it possible to solve different tasks without worrying about what kind of notation or element linking the businesses need.

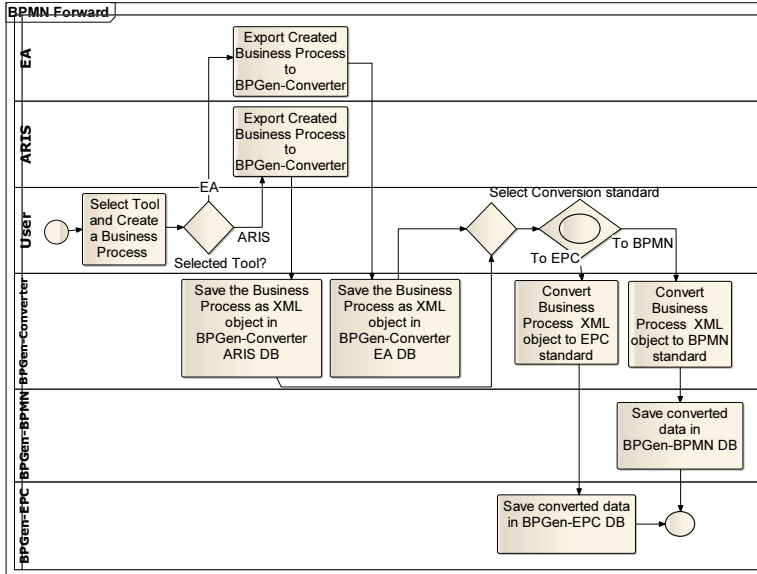


Fig. 5 Main Forward Converter Business Process

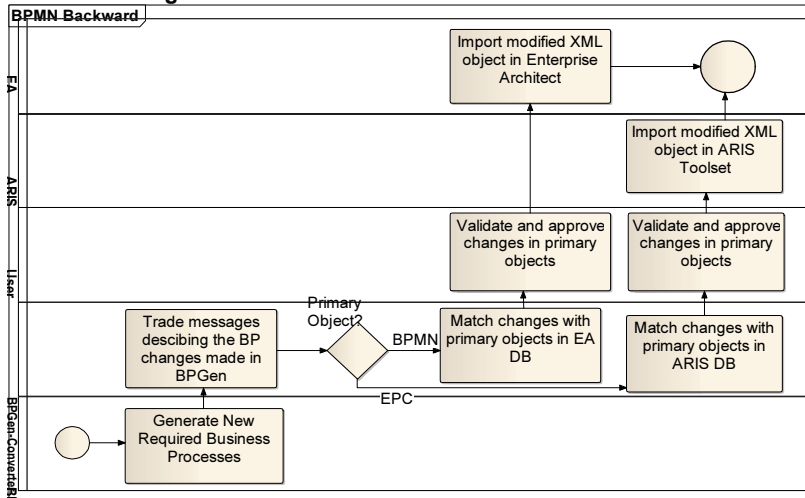


Fig. 6 Main Backward Converter Business Process

The actual work process (Fig.5) can be initiated by the er or one of the BP generators. If the user wants to interact with the system he/she should first create a business process and select an environment to represent it in one of the business process modeling standards. When this task is done the business process should be exported to an XML file and sent to BPGen Converter. The system saves the original business process as an XML object in the internal respective data base, depending on the tool selected by the user. Afterwards a conversion standard is to be chosen. The user can either select one between EPC and BPMN or choose both of them. Shortly after the

chosen conversion is completed by the converter module of the system and then sent to the respective external data base that makes it reachable for the business process generators.

At some point one of the BP generators can initiate a conversion back to a diagram view. In order for this to happen a new business process (Fig.6) should be requested by a user of the respective tool and then generated. Then the BPGen Converter trades messages describing the new business process and what changes were made on the original object. These tasks are marked as a Send and a Receive task. After the needed information is gathered by the system the type of the primary object is checked. Depending on that the converter matches the changes with the source object (EA or ARIS object) that was earlier saved in the respective data base. Finally, the user should validate and approve the changes and then import the modified XML object to its business process modeling environment and see the result diagram.

6 CONCLUSIONS

After a short research of business process modeling principles we managed to create a tool to serve our and hopefully others' needs. We designed and created a converter that will let us use our existing business process generator that works with Event-Driven Process Chains in combination with a powerful tool such as Sparx Systems Enterprise Architect. The converter successfully translates data from a business process model XML file and provides it for modification to the business process generator. After the linking task is done and the business process is ready the converter translates the information back to the XML file.

Future plan is to enhance the converter and make it possible to work with EPC environment (such as ARIS Toolset) and a business process generator that works under the rules and conditions of BPMN.

REFERENCES

- [1] Grigorova K., P. Hristova, G. Atanasova and J. Q. Trelewicz, Levels of Business Structures Representation, IBM Research Report, RJ10349, 2005.
- [2] Keller G., M. Nüttgens and A. W. Scheer. Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)". Technical Report 89, Institut für Wirtschaftsinformatik Saarbrücken, Saarbrücken, Germany, 1992.
- [3] Object Management Group (OMG), Business Process Modeling Notation. V1.1. Technical report, January 2008.
- [4] Stein S. BPMN EPC semantic comparison. Aris Community. Apr 2010.
- [5] Tscheschner W., Transformation from EPC to BPMN. Oryx Research, Potsdam, Germany, 2010.

CONTACT DATA:

Vladimir Kotsev, BSc, Informatics and Information Technologies Department, University of Ruse, e-mail: vladimir.kotsev@googlemail.com

Ivan Stanev, MSc Eng, Informatics and Information Technologies Department, University of Ruse, tel. 082-888-326, e-mail: instanev@gmail.com

Katalina Grigorova, prof, PhD, Informatics and Information Technologies Department, University of Ruse, e-mail: k.grigorova@gmail.com

Докладът е рецензиран.