Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1355–1358
DOI: 10.15439/2016F376
ACSIS, Vol. 8. ISSN 2300-5963

Spreadsheet-Based Business Process Modeling

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

***Abstract*—Business Process models help to visualize processes of an organization. In enterprises, these processes are often specified in internal regulations, resolutions or other law acts of a company. Such descriptions, like task lists, have mostly form of enumerated lists or spreadsheets. We present a method how to generate a BPMN process model from a spreadsheet-based representation. In contrast to the existing approaches, our method does not require explicit specification of gateways in the spreadsheet, but it takes advantage of nested list form.**

***Index Terms*—Business Process Model and Notation (BPMN), process modeling, spreadsheets, spreadsheet-based modeling**

I. Introduction

PROCESS models constitute a useful knowledge representation. They are commonly used by organizations to depict the workflow of the company, especially to specify alternative flows of tasks and events. Such aspects are often specified using textual description in internal regulations, resolutions or other companies law acts. These descriptions consist of the specification of the steps taken to achieve the specific goal. Such steps can be easily specified using a spreadsheet or an enumerated list (an ordered list of steps can be almost directly transformed into a spreadsheet format).

In this paper, we present a method of generating BPMN process models from spreadsheet-based representation (see Fig. 1). A process can be described using one of the spreadsheet applications like MS Excel, Google Docs or OpenOffice Calc. Based on the CSV file of the model exported from the application, a graphical process model in BPMN can be generated according to the specified transformation rules.

According to the studies [1], up to 60% of the time spent on process management projects can be consumed by the acquisition of process models, which mostly is done manually by process designers or business analytics. Thus, generating or transforming the existing representation to models can shorten this time.

In the field of transforming some kind of process description into a process model, there are various research directions.

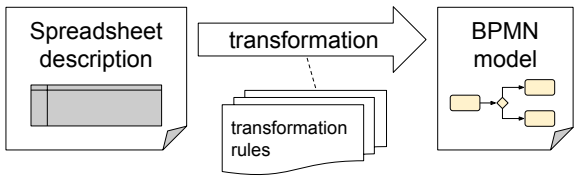


Figure 1. Overview of the spreadsheet-based approach

II. Related works

One of the existing methods is generating models from text description. A text can be provided in natural language [1] or in structured language. A part of the SBVR standard is SBVR Structured English [2]. There are methods of tranforming SBVR business rules into UML activity diagrams [3] (which are similar to BPMN models) or BPMN [4], [5]. Some papers consider the extended versions of SBVR, like SBPVR (Semantics of Business Process Vocabulary and Process Rules) [6] or sSBVRMM (simplified SBVR metamodel) [7]. There are also methods for analysis and validation of processes which use natural language generation [8] or spreadsheets [9].

Another method which provides good quality models is translation from other representation. An example of transformation approach is transformation of the Unified Modeling Language (UML) use case diagrams [10], [11]. The broad family of process model generation methods is process discovery, which is one of the process mining techniques. There are many existing algorithms, mostly implemented using the ProM process mining toolkit. Among them, there are algorithms to mine BPMN models [12].

The most similar approach to ours was presented by Krumnow and Decker in [13], [14]. They proposed three different approaches for business process modeling using spreadsheets:

1) Simple approach – this solution concerns simple processes modeled only using sequences of activities.
2) Branching approach – concerns not only sequences, but also more complex flow structures. It uses such elements like successors in order to represent complex control flow, as well as gateways and events. For the condition, the "Description" column is used. In the case of several successors, this can be realized by using a comma-separated string of row numbers as property value.
3) More Properties approach – extends the branching approach with additional specified explicitly properties like the assigned roles, input documents, etc. It allows for adding customized properties as new columns to the spreadsheet, which are not presented in the model.

The second and third approaches require from the user some kind of familiarity with business process modeling notation. According to these approaches, the user has to specified such elements like gateways or successors.

In our research, we propose a new approach, which solves the problem of familiarity with business process modeling notation. The solution is described in the following section.

Table I
SIMPLE EXAMPLE OF A XOR-GATEWAY IMPLEMENTATION

Order	Activity	Condition	(..)
4a	Send Ticket	Payment registered	
4b	goto 6	else	
5	...	...	
6	...	...	

III. TRANSFORMING SPREADSHEET INTO BPMN MODEL

The translation method presented in this section transforms a spreadsheet-based representation into a BPMN 2.0 model. In the following subsections, we describe the requirements and assumptions that we took into account during method development, the supported representation, as well as we provide the detailed description of transformations along with transformation procedure.

In order to make the solution widely applicable and simple to use, some assuptions were made. The following requirements need to be considered while providing a spreadsheet-based representation of a business model:

- The user should be able to create a business process model using his favourite popular spreadsheet application (e.g. MS Excel, Google Docs, OpenOffice Calc).
- A graphical model should be created using a CSV file (an exported spreadsheet).
- Only one pool is considered and the term "Who" is used instead of swimlane to distinguish different task executors.
- Logical gates are eliminated from the model. A process should be described as a set of phases. If two tasks are executed in the same phase, it means that there are parallel or connected by an inclusive or exclusive alternative (OR/XOR). The relationship between tasks is determined by a condition stored in a separate column (no condition – AND, two different conditions – OR, condition and „else" statement - XOR). Table I shows how logical gates can be represented.
- Loops are represented by "goto" tasks, which link one phase with another.

In the proposed solution, a business process is represented by a spreadsheet table, where rows correspond to tasks (or phases). Columns contain properties of the selected phase. Task and condition names should start with a capital letter, while all instructions are written in lowercase. Following column types are used:

- Order – number of the corresponding phase (starting from one). If two or more tasks are performed in parallel or alternatively, we use letters to distinguish different branches.
- Activity – name of the performed action or instruction. For example, if there is a need for a loop or skipping the phase, this field should be filled with a „goto" statement and number of the desired phase, e.g. „goto 7".
- Condition – a condition needed to perform the selected action. If the task executes every time this field should be left empty. In order to implement a XOR-gateway this field should be filled with an appropriate condition and the word „else" for the other task performed in the same phase. In order to implement an OR-gateway the fields mentioned before should be filled with two separate conditions.
- Who – a department or person that executes the task. This field corresponds to a swimlane in BPMN.
- Subprocess – information if the selected task contains a subprocess. If so, this field should be filled with „yes" and a sheet with a name of this subprocess should be created. Otherwise the field should remain empty.
- Terminated – information if the selected task terminates the process. If so, this field should be filled with „yes". Otherwise the field should remain empty.

Supported process elements and their spreadsheet model

1) Start events: In the proposed model an assumption is made, that a business process starts in one specified moment, which is called the „0 phase". First 3 rows of Table II present examples of none, message and timer start events.

2) End events: End events are represented by the statement „yes" in the column „Terminated". The field "Activity" can either contain the name of the last activity or remain blank. In 4th and 5th row of Table II the examples of none and message end events are presented.

3) Tasks: The name of a task is stored in the „Activity" column and should start with a capital letter. To skip a phase or go back to a previous phase it is possible to use the „goto" statement in the activity field.

4) Collapsed subprocesses: If a task is a collapsed subprocess, the „Subprocess" field should be filled with „yes" and the subprocess should be then modelled in a separate spreadsheet.

5) Parallel-, Exclusive- and Inclusive-gateways: Gateways are represented by alternative branches in the sequence (phase) flow. A phase preceded by a logical gateway is named as follows: NxM, where N is the (natural) number of the phase in the main process branch, x is a letter (a-z) corresponding to the alternative branch and M is the number of the phase in the selected branch. If the branch contains only one task M can be omitted. We assume that the branching is always ended by the same type of gateway that started it. It may be perceived as a limitation, but in fact it prevents model unconsistency.

The representation of a simple AND-gateway was presented in the 6th row of Table II. In order to implement an XOR-gateway, the field „Condition" should be filled in with the appropriate condition and with the word „else" for the other task performed in the same phase, but in another branch. An example of a simple XOR-gateway was presented in Table I. In order to implement an OR-gateway the fields mentioned before should be filled with two separate conditions. An example of an OR-gateway within multiple gateway spreadsheet representation was presented in the last row of Table II.

6) Pool/lane: Only one pool is considered. Different swimlanes are represented by the field „Who" containing the appropriate department or name.

Our spreadsheet-based approach is dedicated to people who are unfamiliar with business process modelling and that is

Table II
TRANSFORMATION RULES FROM SPREADSHEET-BASED REPRESENTATION TO BPMN PROCESS MODEL STRUCTURE

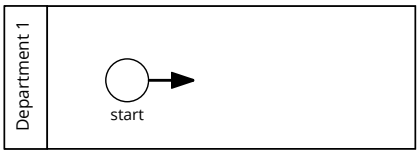
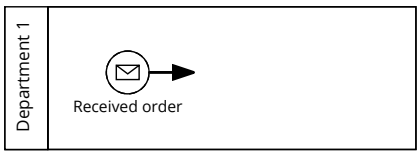
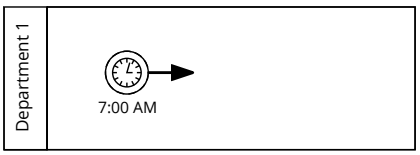
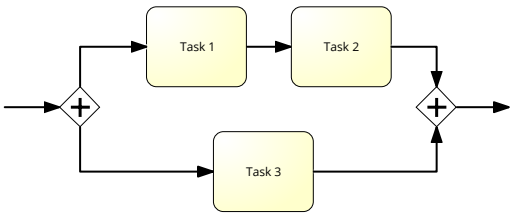
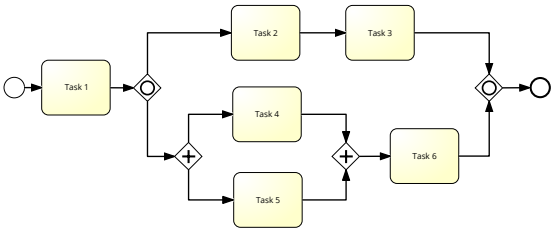
Spreadsheet representation	BPMN Element
<table><tr><td>Order</td><td>Activity</td><td>Condition</td><td>Who</td><td>(...)</td></tr><tr><td>0</td><td>start</td><td></td><td>Department 1</td><td></td></tr></table>	
<table><tr><td>Order</td><td>Activity</td><td>Condition</td><td>Who</td><td>(...)</td></tr><tr><td>0</td><td>message Receive Order</td><td></td><td>Department 1</td><td></td></tr></table>	
<table><tr><td>Order</td><td>Activity</td><td>Condition</td><td>Who</td><td>(...)</td></tr><tr><td>0</td><td>timer 7:00 AM</td><td></td><td>Department 1</td><td></td></tr></table>	
<table><tr><td>Order</td><td>Activity</td><td>Condition</td><td>(...)</td><td>Terminated</td></tr><tr><td>99</td><td>Request completed</td><td></td><td></td><td>yes</td></tr></table>	
<table><tr><td>Order</td><td>Activity</td><td>Condition</td><td>(...)</td><td>Terminated</td></tr><tr><td>99</td><td>message Send Report</td><td></td><td></td><td>yes</td></tr></table>	
<table><tr><td>Order</td><td>Activity</td><td>Condition</td><td>(...)</td></tr><tr><td>2a1</td><td>Task 1</td><td></td><td></td></tr><tr><td>2a2</td><td>Task 2</td><td></td><td></td></tr><tr><td>2b</td><td>Task 3</td><td></td><td></td></tr></table>	
<table><tr><td>Order</td><td>Activity</td><td>Condition</td><td>(...)</td></tr><tr><td>0</td><td></td><td></td><td></td></tr><tr><td>1</td><td>Task 1</td><td></td><td></td></tr><tr><td>2a1</td><td>Task 2</td><td>Condition 1</td><td></td></tr><tr><td>2a2</td><td>Task 3</td><td></td><td></td></tr><tr><td>2b1a</td><td>Task 4</td><td>Condition 2</td><td></td></tr><tr><td>2b1b</td><td>Task 5</td><td></td><td></td></tr><tr><td>2b2</td><td>Task 6</td><td></td><td></td></tr><tr><td>3</td><td>(end)</td><td></td><td></td></tr></table>	

Table III
COMPARISON TO THE EXISTING APPROACHES

Element type	Simple approach	Branching approach	More Properties approach	Our approach
Task	●	●	●	●
Events	❍	●	●	●
Collapsed Subprocess	❍	●	●	●
AND, OR and XOR Gateways	❍	●	●	●
Pool, Lane	❍	◑	◑	◑
Data Object	❍	◑	●	❍
Sequence Flow	◑	●	●	●
Message Flow	❍	❍	❍	❍

why it is important to transform the created spreadsheet into a BPMN diagram in a simple way. Such a diagram is a correct BPMN model, however, it can still contain some behaviour anomalies if the spreadsheet contains some loops [15].

The proposed set of transformation rules can use a CSV file that can be created from any spreadsheet software. The table containing information about the business process is represented as an array of structures, where each row is a unique part of structure with some element fields in the row.

Table III presents the overall comparison of our approach to the existing approaches in terms of supported elements by these solutions. One can noticed that our approach supports fewer elements than such complex approaches as Branching or More Properties approaches. However, the other approaches support the elements in a straightforward way, requiring the deeper knowledge of business process elements, even in the case of such simple structures like gateways.

Moreover, there are several points, in which our approach has the advantage over the existing approaches:

- We do not use explicit notion of XOR/OR/AND gateways, thus the user does not have to think about the kind of flow branching. Instead, in our model it is modeled in the implicit way.
- For describing condition, we use the ”condition” column, what is more clear than using the ”description” field.
- We do not use the notion of ”Successor” as it does not always show clearly the flow in the spreadsheet representation. Thus, in our opinion, a well-known jump statement ”goto” (one-way control flow transfer), existing in many computer programming languages, in this particular usage performs better, as it is clearly visible in the Activity field.

IV. CONCLUSIONS

In the paper, we present a new method of generating a BPMN process model based on its spreadsheet-based representation. In contrast to the existing approaches, our method does not require explicit specification of gateways in the spreadsheet, but it takes advantage of the spreadsheet with numbered rows in the form of a nested list. Thanks to this, our method does not require the knowledge of the BPMN notation or any business process notation.

As future works, we plan to extend the method in order to support multiple pools, message flows, as well as more type of events, including boundary events.

REFERENCES

[1] F. Friedrich, J. Mendling, and F. Puhlmann, “Process model generation from natural language text,” in *Advanced Information Systems Engineering*, ser. Lecture Notes in Computer Science, H. Mouratidis and C. Rolland, Eds. Springer Berlin Heidelberg, 2011, vol. 6741, pp. 482–496.

[2] F. Levy and A. Nazarenko, “Formalization of natural language regulations through sbvr structured english,” in *Theory, Practice, and Applications of Rules on the Web*, ser. Lecture Notes in Computer Science, L. Morgenstern, P. Stefaneas, F. Levy, A. Wyner, and A. Paschke, Eds. Springer Berlin Heidelberg, 2013, vol. 8035, pp. 19–33. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-39617-5_5

[3] A. Raj, T. V. Prabhakar, and S. Hendryx, “Transformation of SBVR Business Design to UML Models,” in *Proceedings of the 1st India Software Engineering Conference*, ser. ISEC ’08. New York, NY, USA: ACM, 2008, pp. 29–38.

[4] O. C. Tantan and J. Akoka, “Automated transformation of Business Rules into Business Processes,” in *Proceedings of the Twenty-Sixth International Conference on Software Engineering and Knowledge Engineering*, 2014, pp. 684–687.

[5] K. Kluza and K. Honkisz, “From SBVR to BPMN and DMN models. proposal of translation from rules to process and decision models,” in *Artificial Intelligence and Soft Computing*, ser. Lecture Notes in Computer Science, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, Eds. Springer International Publishing, 2016, vol. 9693.

[6] A. Raj, A. Agrawal, and T. V. Prabhakar, “Transformation of Business Processes into UML Models: An SBVR Approach,” *International Journal of Scientific and Engineering Research*, July 2013.

[7] B. Steen, L. Pires, and M.-E. Iacob, “Automatic Generation of Optimal Business Processes from Business Rules,” in *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2010 14th IEEE International*, Oct 2010, pp. 117–126.

[8] L. Henrik, J. Mendling, and A. Polyvyanyy, “Supporting process model validation through natural language generation,” *IEEE Transactions on Software Engineering*, vol. 40, no. 8, pp. 818–840, 2014.

[9] J. Saldivar, C. Vairetti, C. Rodríguez, D. Florian, F. Casati, and R. Alarcón, “Analysis and improvement of business process models using spreadsheets,” *Information Systems*, vol. 57, pp. 1–19, 2016.

[10] J. R. Nawrocki, T. Nedza, M. Ochodek, and L. Olek, “Describing business processes with use cases,” in *BIS*, 2006, pp. 13–27.

[11] D. Lubke, K. Schneider, and M. Weidlich, “Visualizing use case sets as bpmn processes,” in *Requirements Engineering Visualization, 2008. REV ’08.*, 2008, pp. 21–25.

[12] A. A. Kalenkova, M. de Leoni, and W. M. van der Aalst, “Discovering, analyzing and enhancing bpmn models using prom?” in *Business Process Management-12th International Conference, BPM*, 2014, pp. 7–11.

[13] S. Krumnow and G. Decker, *Business Process Modeling Notation: Second International Workshop, BPMN 2010, Potsdam, Germany, October 13-14, 2010. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ch. A Concept for Spreadsheet-Based Process Modeling, pp. 63–77.

[14] S. Krumnow, “Spreadsheet-based process modeling,” *Business Processes in the Real World*, pp. 55–70, 2010.

[15] A. Mroczek and A. Ligęza, “A note on bpmn analysis. towards a taxonomy of selected potential anomalies,” in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*. IEEE, 2014, pp. 1097–1102.