DOI: 10.15439/2014F185
ACSIS, Vol. 2

A Note on BPMN Analysis. Towards a Taxonomy of Selected Potential Anomalies

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

Antoni Ligęza
AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Kraków, Poland
Email: ligeza@agh.edu.pl

Abstract—**Modeling based on a graphical notation understandable for different specialists has become very popular. Within the area of business processes, the most common one is the Business Process Modeling and Notation (BPMN). BPMN is aimed at all business users who design, analyze, manage and monitor business processes. Most papers in this area focus on making use of the possibilities that BPMN makes available, but there is lack of papers analyzing possible errors and ways of detecting and eliminating them. Specification of a BPMN diagram is relatively precise, but it is only a descriptive form presented at some abstract, graphical level. Hence, the main focus of this article is an attempt to analyze the topic of the anomalies which are likely to occur when modeling with use of BPMN.**

I. Introduction

CURRENTLY, the approach to modeling based on a graphical notations understandable for different specialists has become very popular. In the are of business processes the most common one is the Business Process Modeling and Notation (BPMN). BPMN is a business process modeling standard developed by Business Process Management Initiative. At present, BPMN is supported by the *Object Management Group* (OMG) because the two organizations merged in 2005.

In March 2011 the most recent specification of BPMN (BPMN 2.0) was released. The purpose of BPMN was to create a uniform notation of business processes that would be generally understandable — from professional process analysts, through managers to ordinary workers. According to [1], BPMN 'a standard Business Process Model and Notation (BPMN) will provide businesses with the capability of understanding their internal business procedures in a graphical notation and will give organizations the ability to communicate these procedures in a standard manner. Furthermore, the graphical notation will facilitate the understanding of the performance collaborations and business transactions between the organizations'. The main aims of BPMN include the following:

- process visualization which uses a graphical presentation of a business process. This form of visualization is much more effective than a textual representation;
- documentation through specification of process features;
- communication — provides a set of simple, commonly understandable notations.

BPMN is aimed at all business users, from the analysts, who create the initial process drafts, through the technical developers, whose responsibility it is to implement the technology performing those processes, and finally, to the business people, who will manage and monitor the aforementioned processes. The notation is clearly identified by various groups of experts, not only those connected with the IT industry. Yet, in spite of numerous endeavors, problems with unambiguous interpretation still exist. This fact stems from lack of a satisfactory BPMN interpreter. In fact as a consequence no semantics of BPMN components and connection is provided. Hence, various devices can interpret BPMN differently. The fact that there is no formal semantics may lead to misinterpretations and errors.

Most papers in the business process area focus on making use of the possibilities that BPMN brings, but there is lack of papers analyzing errors and ways of eliminating them. BPMN specification is precise but it is only a descriptive, graphical form. Hence, the subject of this article is an attempt to analyze the topic of the anomalies which are likely to occur in BPMN.

This paper is a kind of survey on anomalies in BPMN. An attempt has been made at presenting a taxonomy of possible problems, both of static, structural and dynamic nature. The research is based on literature analysis and some limited experience with BPMN models.

The article has been divided into five sections. The first one covers the basic elements of BPMN, the second presents an overview of the literature on anomalies. The consequent part touches potential misinterpretations and errors; it is based on examples. The last section contains the summary and conclusions.

II. ELEMENTS OF BPMN

BPMN model consists of simple diagrams made up of a limited set of graphical elements. Simplification of activity flows and processes is clearer for business users and developers. There are four main elements of BPMN, namely: Flow Objects, Connecting Objects, Swimlanes and Artifacts.

A. Flow Objects

Flow objects are the key elements describing BPMN. They consist of three core elements: events, activities and gateways [2].

An Event is represented by a circle and means something that happens (compared to an Activity, which is something

978-83-60810-58-3/$25.00 © 2014, IEEE

An Event is represented by a circle and means something that happens (compared to an Activity, which is something that has been done). The circular figures differ depending on the type of Event. Events may have an impact on a business process. An event can be an external or internal one. As long as they can influence the process being modeled, they should be modeled.

In general, there are three types of Events: Start, Intermediate and End. Start Event works like a trigger to a process. It is important for every process to have a Start Event to show the beginning of the business process. It allows readers to locate in the BPMN diagram where the process begins, and under what conditions. End event is used to indicate where the business process finishes. It presents the outcome of the process. Intermediate Event represents what happens in the gap between Start Event and End Event. It is responsible for driving a business flow based on the event it specifies.

An Activity is represented by a rounded-corner rectangle and describes a type of work that has to be completed within a business process. There are two kinds of Activities: Tasks and Sub-processes. Task means a single unit of work which is not or cannot by divided in the next stage of business processes specification; in certain sense a task is of an *atomic* nature. On the other hand, sub-process is used for complex work which can be divided into smaller units. It is applied in order to cover or uncover additional specification levels of business processes.

Gateways are elements used to monitor the way in which some business process flows interact with the others. A Gateway is represented by a diamond shape. Some of the typical types of gateways are the following ones:

- **Data-Based Exclusive Gateway** — it is used to control process flow based on given process data.
- **Inclusive Gateway** can be used to create parallel paths. (Fig. 1).

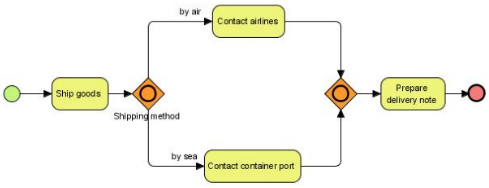

Fig. 1. Inclusive gateway

- **Parallel Gateway** — it is used to model the execution of parallel flows without the need of checking any conditions, all outgoing flows must be executed at the same time (Fig. 2).
- **Event-Based Gateway** — it is used to model alternative paths that are based on events (Fig. 3).

B. Connecting Objects

Flow objects are connected to each other using Connecting objects, which are of three types: sequences, messages, and associations [2]. Sequence Flow is used to show the order in which particular activities will be performed in a process. Message Flow is used to show the flow of messages between two process participants entitled to send and receive them. Association is used to link information and artifacts to activities, events, gateways and flows.

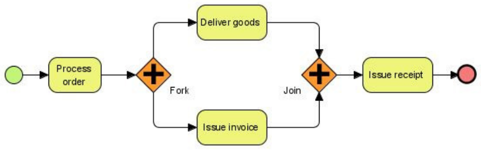

Fig. 2. Parallel gateway

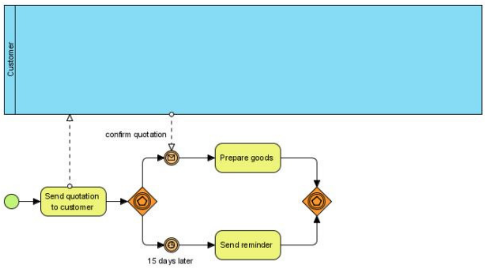

Fig. 3. Event-based gateway

C. Swimlanes

BPMN usually uses the concept of swimlanes in order to demonstrate what business function a particular activity is connected with or what system executes it. There are two types of swimlanes objects: lanes (sub-partition of pools) and pools (represent participants in a business process) [2].

Inside a pool, there are flow elements. It acts as a graphical container for partitioning a set of Activities from other Pools. Lanes can be used to represent specific objects or roles engaged in a process. They are used to organize and categorize activities in a pool, according to the function and role. They are represented by a rectangle extending either vertically or horizontally along the length of the pool. A lane contains flow objects, connecting objects and artifacts.

D. Artifacts

Artifacts are diagram elements used to display additional information relative to the process. They enable programmers to include more information in a model. In this way, the model becomes clearer. BPMN does not restrict the number of artifacts, though currently three have been defined [2]:

- *Data objects* are a mechanism whose aim is to show how data is prerequisite or result from activities. They are connected to activities though Associations.
- *Groups* can be used for analysis or documentation objectives but they do not affect the sequence flow.
- *Annotations* are a mechanism used in modeling to provide additional text information for the reader of BPD (Business Process Diagram).

III. THERMOSTAT: AN ILLUSTRATIVE EXAMPLE [3].

In order to provide intuitions, the theoretical considerations will be illustrated with a simple example process. The process goal is to establish the so-called *set-point* temperature for a thermostat system [4]. The selection of the particular value depends on the season, whether it is a working day or not, and the time of the day. A BPMN diagram of the process is specified in Figure 4.

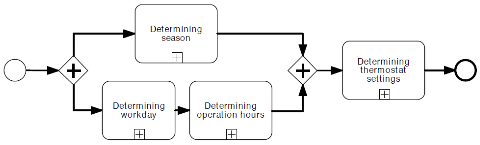

Fig. 4. An example BPMN diagram top-level specification of the thermostat system

After start, the process is split into two independent paths of activities. The upper path is aimed at determining the current season [1] *aSE*; it can take one of the values *sum, aut, win, spr*; the detailed specification is provided with rules 7-10. A visual specification of this activity with an appropriate set of rules is shown in Fig. 5.

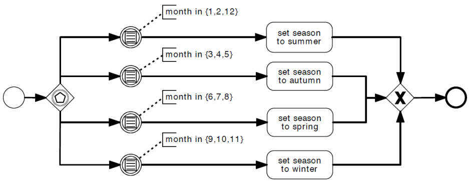

Fig. 5. An example BPMN diagram detailed specification a BPMN task

The lower path determines whether the day (*aDD*) is a workday (*aTD=wd*) or a weekend day (*aTD=wk*), both specifying the value of today (*aTD*); specification provided with rules 1 and 2, and then, taking into account the current time (*aTM*), whether the operation (*aOP*) is during business hours (*aOP=dbh*) or not (*aOP=ndbh*); the specification is provided with rules 3-6. This is illustrated with Fig. 5 and Fig. 6.

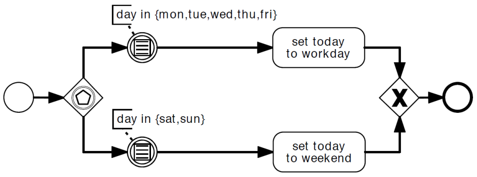

Fig. 6. An example BPMN diagram detailed specification of determining the day task

The whole process is formally specified with the following eighteen inference rules.

Rule 1:$\in \{,,,,\} \longrightarrow = wd.$

Rule 2:$\in \{,\} \longrightarrow = wk.$

Rule 3:$= wd \wedge \in (9,17) \longrightarrow = dbh.$

Rule 4:$= wd \wedge \in (0,8) \longrightarrow = ndbh.$

Rule 5:$= wd \wedge \in (18,24) \longrightarrow = ndbh.$

Rule 6:$= wk \longrightarrow = ndbh.$

Rule 7:$\in \{,,\} \longrightarrow = sum.$

Rule 8:$\in \{,,\} \longrightarrow = aut.$

Rule 9:$\in \{,,\} \longrightarrow = win.$

Rule 10:$\in \{,,\} \longrightarrow = spr.$

Rule 11:$= spr \wedge = dbh \longrightarrow = 20.$

Rule 12:$= spr \wedge = ndbh \longrightarrow = 15.$

Rule 13:$= sum \wedge = dbh \longrightarrow = 24.$

Rule 14:$= sum \wedge = ndbh \longrightarrow = 17.$

Rule 15:$= aut \wedge = dbh \longrightarrow = 20.$

Rule 16:$= aut \wedge = ndbh \longrightarrow = 16.$

Rule 17:$= win \wedge = dbh \longrightarrow = 18.$

Rule 18:$= win \wedge = ndbh \longrightarrow = 14.$

Let us briefly explain these rules. The first two rules define if we have today (*aTD*) a workday (*wd*) or a weekend day (*wk*). Rules 3-6 define if the operation hours (*aOP*) are during business hours (*dbh*) or not during business hours (*ndbh*); they take into account the workday/weekend condition and the current time (*hour*). Rules 7-10 define the season (*aSE*) is summer (*sum*), autumn (*aut*), winter (*win*) or spring (*spr*).

Finally, the results are merged together, and the final activity consists in determining the thermostat settings (*aTHS*) for particular season (*aSE*) and time (*aTM*) (the specification is provided with rules 11-18).

This is illustrated with Fig. 7.

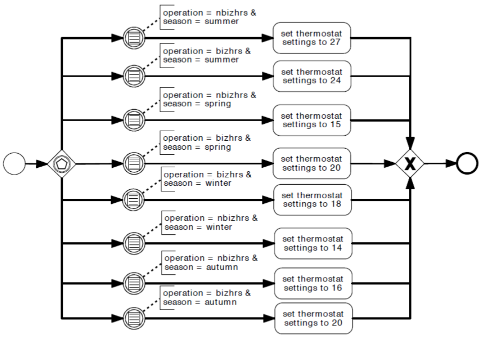
operation = nbizhrs & season = summer
operation = bizhrs & season = summer
operation = nbizhrs & season = spring
operation = bizhrs & season = spring
operation = bizhrs & season = winter
operation = nbizhrs & season = winter
operation = nbizhrs & season = autumn
operation = bizhrs & season = autumn
set thermostat settings to 27
set thermostat settings to 24
set thermostat settings to 15
set thermostat settings to 20
set thermostat settings to 18
set thermostat settings to 14
set thermostat settings to 16
set thermostat settings to 20

Fig. 7. An example BPMN diagram — detailed specification of the final thermostat setting task

Even in this simple example, answers to the following important questions are not obvious:

1) *Data flow correctness*: Is any of the four tasks/activities specified in a correct way? Will each task end with producing desired output for any admissible input data?
2) *Split consistency*: Will the workflow possibly explore all the paths after a split? Will it always explore at least one?
3) *Merge consistency*: Will it be always possible to merge knowledge coming from different sources at the merge node?
4) *Termination/completeness*: Does the specification assure that the system will always terminate producing some

[1] For technical reasons all attribute names used in this example start with lower-case 'a'.

temperature specification for *any* admissible input data?

5) *Determinism*: Will the output setting be determined in a unique way?

Note that we do not ask about *correctness* of the result; in fact, the rules embedded into a BPMN diagram provide a kind of *executable specification*, so there is no reference point to claim that final output is correct or not.

IV. Related work

There is a possibility of defining incoherent business logic specification and its interpretation. Even in basic processes anomalies are observed [5]. An improvement is required in the mechanism which provides cohesion in detecting anomalies in business processes [6]. Anomalies have been defined in numerous papers, yet a uniform definition was presented in [7] IEEE standard classification for Software Anomalies and it says: *"Each condition different from the expected is an anomaly"*.

In business logic an anomaly can be considered as every negative influence on modeling and models. There is a special kind of anomaly — a defect, which blocks the correct and efficient flow of objects completely.

A taxonomy of anomalies was created on the basis of literature. It concerns the flow control, bases and verification rules of data as well as flow accuracy. The taxonomy can make up a base for classification and research on anomaly possibilities.

The anomaly problem in BPMN is based on searching business logic for particular patterns. In [8], typical controls for anti-patterns are searched for by using a query language for BPMN. It is confirmed by deadlocks or livelock patterns which are used improperly. A similar thing happens in [9] where typical gateway constellations leading to problematic situations in the flow work diagram are presented. A comparable situation occurs in [10] as well, where an 'anomaly pattern' is used. This approach is based on detecting anti-patterns in the data flow. The whole thing is based on time logic using a real model control. By making use of different tools, position [11] is focused on various anomalies which stem from formalism or inadequacy of the tools. Yet another approach is a conception based on UML diagrams in development stages [12].

Control flow anomalies concern problems connected with flow control and gateways conditions [12]. In [13] , a problem was presented of control over many semantically identical connections between two work flow elements. This multiplicity complicates changes in the work flow, which is not desired.

Another element of flow control are gateways placed in the modeling center. It was stated in [14] that XOR-gateways with undefined gateway conditions can cause practical problems or even be a reason of an error. A similar thing happens when XOR-gateways conditions do not exclude each other and partly or fully overlap. What happens in flow control in case of lack of synchronization is multiple flow execution. For example, branches and some loop instruction cause such an anomaly [15].

Another situation is a flow deadlock. It is a situation in which the work flow is stopped in the current position of the path and cannot be accomplished. Another lock of flow is known as livelock. In [8] it is called an 'infinite loop'. Flow livelock keeps the operating work flow system in an infinite loop. The reason are bad modeling conditions, which prevent leaving the loop. Both cases — deadlock and livelock are described in [8], [15].

Rule-based anomalies are described in numerous papers [16], [17], [18], [19]. They involve mainly two problems connected with base rules. First, Rule-base Consistency are anomalies concerning coherence. Problems result from the set of rules, which have determined conditions but different outcomes at the same time. Rule-base livelocks, also called *circular rules* [17]. Rule-base livelocks and rule-base deadlocks describe a problem with creation rules, which are dependent on one another although they should not. This type of anomaly suggests that rule-base does not encompass the basic context in which it is used. Coverage anomalies concern the rules in which conditions can be fulfilled by the base context but conclusions are modeled in such a way that no effect will ever be seen. Another type of data flow anomaly is based on [20]. Such anomalies are influenced by those data elements which can be processed by workflow activities.

V. Anomalies in business processes

There are two kinds of business process anomalies which can occur while process modeling, namely [19]: Syntactic anomalies and Structural anomalies.

A. Syntactic Anomalies in Business Process

Analysis of Syntactic Business Process anomalies is important while designing a business process model. In this section examples of syntactic anomalies in business processes will be presented. A division into three groups has been made: **Incorrect usage of Flow Objects**, **Incorrect usage of Connecting Object** and **Incorrect usage of Swimlanes**.

B. Incorrect usage of Flow Object

The anomaly of this type result from improper use of the Event, Activity and Gateway.

Incorrect usage of Activities: Invalid use of Start Event or End Event. The BPMN specification defines the start and end events as optional. However, their usage is highly recommended, since each process starts and ends somewhere. Without explicitly using start and end events, a regular BPMN process might look the process in Fig. 8. This modeling approach is undesirable and could lead to misinterpretations.

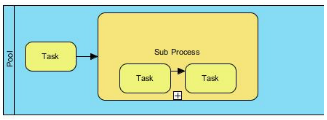

Fig. 8. Implicit process events

Depending on application, three anomalies can be distinguished. These are: Activities without Activation, Activities without Termination and Invalid use of Receive Task. Activities without Termination and Invalid use of Receive Task.

- **Activities without activation** If an activity is situated on a path that has no start, then this is an activity without activation. Even if a start of an event is used.
- **Activities without Termination**. An activity without termination happens when the activity cannot be brought to an end. Even if End Event is used.
- **Invalid use of receive task**. Receive Task Element is designed to wait for incoming messages from outside users in a business process.

Invalid use of Gateway. There are two groups of anomalies: invalid use of Data-Based XOR Gateway and invalid use of Event-Based XOR Gateway.

- **Invalid use of Data-Based XOR Gateway**. A data-based XOR Gateway relies on the arrival of a data token that has traversed the Process Flow. Data-based XOR Gateway must be date-based objects.
- **Event-Based XOR Gateway**. According to BPMN, the event-based gateway cannot be used as a merge gateway. It can only be used as a decision type gateway (multiple outputs).

Incorrect usage of Connecting Object. Anomalies concerning connecting objects stem from incorrect usage of their elements, that is message flow and sequence flow. As far as incorrect utilization of connecting objects is concerned, a few anomalies can be differentiated: the ones concerning incoming sequence flows, outgoing sequence flows, invalid use of conditional sequence flow. In this case there are two possible irregularities regarding the invalid use of a pool or lane [26].

C. Invalid use of Pool

When modeling multiple pools, a common mistake is that activities in a Pool are not connected with sequence flows. It is incorrect to use multiple pools as a single process and incorrectly interprets messages flows as way of indicating a sequence of activities (Fig. 9).

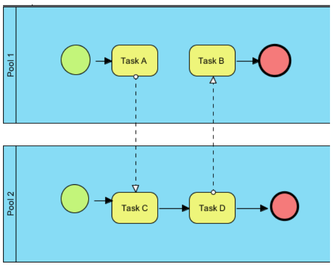

Fig. 9. Missing sequence flow

Another common problem when modeling multiple pools is the use of a set of pools as a single pool with multiple lanes. The end result will be an incorrect model (Fig. 10) that represents a single process that spreads over the boundaries of the pool.

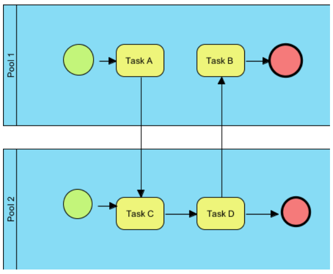

Fig. 10. A Sequence flow May not cross pools boundaries

D. Invalid use of Lane

Improper use of lane as a pool, thereby representing individual processes within separated lanes. This is wrong, because a lane is just a activity-classifying mechanism (Fig. 11).

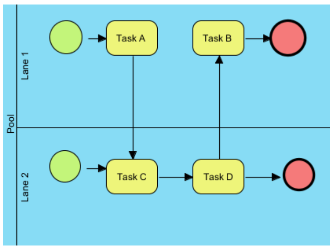

Fig. 11. Two Lances are used as two Pools

E. Structural Anomalies

Structural anomalies have been described in the literature [21], [22], [23], [24]; they are classified as four types: **Deadlock**, **Lack of synchronization**, **Dead Activity** and **Infinite Loop**.

Note that in fact all the above anomalies correspond to *wrong dynamic behavior*; all of them occur during execution of the process.

A process is sound [25] if and only if it is free of two control-flow errors: the deadlock and the lack of synchronization. First, deadlocks are blockings in the process model, which occur when gateways are used incorrectly. In this case, the links in the process where gateways were installed should be checked. Deadlocks occur when an exclusive gateway was picked for linking and this linkage was combined again with a parallel gateway. They may arise from added intermediate events or multiple exclusive start events, which should be checked again. There are two types of deadlocks: deterministic deadlock (Fig. 12) and non-deterministic deadlock (Fig. 13).

A deadlock is a reachable state of the process that contains a token on some Sequence Flow that cannot be removed in any possible future. A lack of synchronization (Fig. 14) is a reachable state of the process where there is more than one token on some Sequence Flow. To characterize the lack of synchronization, we follow the intuition that potentially

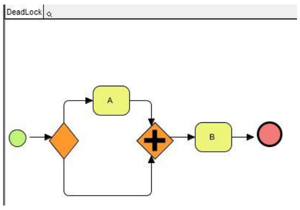

Fig. 12. Deterministic deadlock

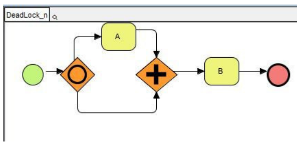

Fig. 13. Non-deterministic deadlock

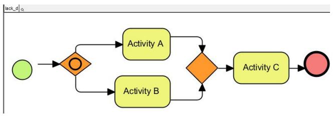

Fig. 14. Lack of synchronization

concurrent paths, paths starting with an IOR-split or an AND-split, should not be joined by XOR-join. In the following, we formalize this characterization and show that such structure always leads to lack of synchronization in deadlocks free acyclic workflow graphs [24]. While *Dead Activities* are activities which will never be executed. A last type of anomaly is Infinite Loop [27] , also called *closed loop*. A closed loop is a cycle without any split. Tokens that enter a closed loop are forever lost to the rest of the workflow. In our model, this leads to a deadlock, because each token entering the closed loop will have a synchronization copy of itself placed on the incoming edge of the initial join that loops back from the cycle. It is hard to imagine a sensible real-world example that contains a closed loop (the BPMN standard document admits this). Banning closed loops from workflows is thus not a serious restriction, especially since infinitely looping cycles are still possible as long as they are not closed [25].

VI. Conclusion

BPMN is a popular business process modeling language. The ability of using it is very important in the modeling stage. Yet, despite its advantages, the problem of effective anomaly detection still remains. There is a lack of a proper tool that would automate the process of detecting anomalies in business process modeling.

References

[1] The BPMN page: http://www.bpmn.org

[2] The BPMN documentation page: [2]http://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf

[3] A. Ligęza,'BPMN a logical model and property analysis. Decision Making in Manufacturing and Services', vol. 5 2011, pp. 57-67, http://dx.doi.org/10.1002/widm.11

[4] M. Negnevitsky, 'Artificial Intelligence. A Guide to Intelligent Systems', Pearson Education Paperback, 2002.

[5] J. Mendling, E. Verbee, B.van Dongen, W.van der Aalst and G.Neumann, 'Detection and Prediction of Errors in EPCs of the SAP Reference Model', *Data & Knowledge Engineering* vol. 64(1), 2008, pp. 312-329.

[6] A. Hallerbach, T. Bauer, R. Manfred, 'Capturing Variability in Business Process Models: The Provop Approach', *Journal of Software Maintenance and Evolution: Research and Practice*, in BPM 2008, vol. 22, 2010, pp. 519-546.

[7] The IEEE Computer Society, 'IEEE Standard Classification for Software Anomalies", 2010 https://standards.ieee.org/findstds/standard/1044-2009.html

[8] R. Laue, A. Awad, 'Visualization of Business Process Modeling Anti Patterns, Visual Formalisms for Patterns', *Electronic Communications of the EASST*, vol. 25, 2009.

[9] S. Kuhne, H. Kern, V. Gruhn, R. Laue, 'Business process modeling with continuous validation', *Journal of Software Maintenance and Evolution: Research and Practice*, BPM 2008 Vol. 22, 2010, pp. 547-566.

[10] N. Sidorova, N. Trcka, W. M. P. van der Aalst, 'Data-Flow Anti-patterns: Discovering Data-Flow Errors in Workflows', Springer, 2009, pp.425-439.

[11] N. Lohmann, K. Wolf, 'How to Implement a Theory of Correctness in the Area of Business Processes and Services', BPM 2010, Springer, 2010, pp. 61-77, dx.doi.org/10.1007/978-3-642-15618-2_7

[12] A. Stephen, 'Process Modeling Notations and Workflow Patterns', White, IBM Corp., United States, Nov. 2009. The BPMN documentation page: http://www.omg.org/bpmn/Documents/Notations_and_Workflow_Patterns.pdf

[13] L. Olkhovich, 'Semi-Automatic Business Process Performance Optimization Based On Redundant Control Flow Detection', AICT-ICIW '06,2006, pp. 146-146.

[14] The OMG documentation page: 'Documents Associated With Business Process Model And Notation (BPMN) Version 2.0', http://www.omg.org/spec/BPMN/2.0/, Jan. 2011.

[15] R. Liu, A. Kumar, 'An Analysis and Taxonomy of Unstructured Workflow', BPM 2005, 2005, pp. 268-284.

[16] X. Desheng, X. Kejian, Z. Dezheng, Z. Huangsheng, 'Model Checking the Inconsistency and Circularity in Rule-Based Expert Systems', *Computer and Information Science*, 2009, pp. 12-17.

[17] A. K. Zaidi, A. H. Levis, 'Validation and verification of decision making rules', *Automatica*, vol. 33, 1997, pp. 155-169.

[18] M. Do ohring, S. Heublein, 'Anomalies in Rule-Adapted Workflows - A Taxonomy and Solutions for vBPMN', *Software Maintenance and Reengineering* (CSMR), 2012, pp. 117 - 126, http://doi.ieeecomputersociety.org/10.1109/CSMR.2012.22

[19] A. Ligęza, G. J. Nalepa, 'A study of methodological issues in design and development of rule-based systems: proposal of a new approach', *Data Mining and Knowledge Discovery*, 2011, pp. 117-137, http://dx.doi.org/10.1002/widm.11

[20] A. Awad, G. Decker, N. Lohmann, 'Diagnosing and Repairing Data Anomalies in Process Models', BPM 2009, Springer, 2009, pp. 5-16, http://dx.doi.org/10.1007/978-3-642-12186-9_2

[21] G. Kim, J. H. Lee, J. H. Son, 'Classification and Analyses of Business Process Anomalies', *Communication Software and Networks*, ICCSN, 2009, pp. 433-437.

[22] H. Lin, Z. Zhao, Z. Chen, 'A novel graph reduction algorithm to identify structural conflicts', HICSS-35 Proceedings of the 35th Hawaii International Conference on System Sciences, 2002, pp. 289.

[23] H. Ling, Z. J. Bo, 'Research on workflow process structure verification', *e-Business Engineering*, ICEBE, 2005, pp. 158-165.

[24] W. M. P. van der Aalst, A. Hirnschall, H. Verbeek, 'An Alternative Way to Analyze Workflow Graphs', 14th International Conference, CAiSE 2002, Springer, 2002, pp. 535-552, http://dx.doi.org/10.1007/3-540-47961-9_37

[25] E. Boerger, O. Soerensen, B. Thalheim, 'On defining the behavior of OR-joins in business process models', *Journal of Universal Computer Science*, vol. 14, no. 1, 2009, pp. 3-32.

[26] The page: http://blog.goodelearning.com/

[27] The BPMI page, 'Business Process Modeling Notation Specification', http://www.omg.org/technology/documents/speccatalog.htm, 2006.