Event anomalies in modeling with BPMN

Anna Suchenia (Mroczek)*  
Antoni Ligęza**

*Cracow University of Technology, 31-155 Cracow Poland,
**AGH University of Science and Technology, 30-059 Cracow, Poland
*asuchenia@pk.edu.pl **aligeza@agh.edu.pl

Abstract

Very popular is a modeling based on a graphical notation, because it is understandable for different specialists. 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. BPMN is relevant from a practical point of view while at the same it offers many challenges for software developers and scientists. Specification of a BPMN diagram is relatively precise, but it is only a descriptive form presented at some abstract, graphical level. 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. Hence, the main focus of this article is an attempt to analyze the topic and example of the anomalies which are likely to occur when modeling with use of BPMN.

1. Introduction

This currently, the approach to modeling based on a graphical notation 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: a) process visualization which uses a graphical presentation of a business process. This form of visualization is much more effective than a textual representation; b) documentation through specification of process features; c) communication — provides a set of simple, commonly understandable notations.

They are four reasons why you should use BPMN in the work of the organization, they are [1]: a) Standard: The standard is supported by many software products; you are less dependent on any particular vendor’s products; b) Simplicity: The principle behind BPMN is rather simple which is why you can very quickly work with this notation; c) Power of expression: If necessary, with BPMN you can describe precisely how a process functions; d) Implementation in IT: BPMN has been primarily developed to support a technical implementation of processes.

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 afore-mentioned 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, no formal of BPMN processes was defined, and — 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.

2. 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 summary of most BPMN elements is shown in Fig. 1. Starting from BPMN 1.2 the number of the elements increases, even if most users use only a little subset of BPMN elements to model business processes. For a complete description of BPMN elements and features refer to [1].

![BPMN elements](image)

Fig. 1. BPMN elements.

Figure number 2 shows a simple example BPMN - working group meeting. Initial starting point is Working Group. Step one: Meeting at two pm on Friday. Step two: Check status of working group. Step three: Is working group still active? If NO then terminate workflow. If YES then go to step four. Step four: Receive information from issue list. Send current issue list then go back to step two.

![Simple example BPMN](image)

Fig. 2. Simple example BPMN.

2.1 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 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 be 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 (Fig.3).
Inclusive Gateway can be used to create parallel paths. The conditions of all outgoing flow are evaluated (Fig. 4). On the Figure presented an example of an inclusive, where a trip is planned.

Fig. 4. 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. 5). On the figure process starts with receiving and preprocessing an order. Then the parallel gateway triggers the execution of three activities. the inventory is updated, the goods are shipped, and the invoice is sent. The activities can be executed concurrently, because there are no execution constraints defined between these.

Fig. 5. Parallel gateway.

Event-Based Gateway — it is used to model alternative paths that are based on events (Fig. 6). On the figure presented process sending of an invoice by a reseller to customer.

Fig. 6. Event-Based Gateway.

2.2 Connecting Objects

Flow objects are connected to each other using Connecting objects, which are of three types (Fig 7.): 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. 7. Sequence and Message

2.3 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 swimlane 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.

In BPMN modeling, all events, activities and gateways are placed in pools or lanes. A pool usually represents an organization and a lane constitutes a department within this organization. A process flow can change the lane when different means are needed to fulfill the task. Who does what can be determined by placing the processes in pools or lanes. By doing the same with events, we can specify where they will appear. Finally, by positioning gateways in pools or lanes, we define
where decisions are made. Apart from an organization, a pool can represent other things like functions, applications, localization, classes or units.

2.4 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 [1, 2]: Data Object represent either data placed to the process, data resulting from the process, data that needs to be collected, or data that needs to be stored. They are connected to activities though Associations. They are four of types: Data Input, Data Output, Data Collection and Data Storage.

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).

3. Example- THERMOSTAT

In order to provide intuitions, the theoretical considerations will be illustrated with a simple example process [3]. 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 8.

After start, the process is split into two independent paths of activities. The upper path is aimed at determining the current season (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. 9.

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. 10 and Fig. 11.

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

Rule 1: $\forall \in \{, , , \} \mapsto wd.$
Rule 2: $\forall \in \{, \} \mapsto wk.$
Rule 3: $\forall wd \in (9,17) \mapsto dbh.$
Rule 4: $\forall wd \in (0,8) \mapsto ndbh.$
Rule 5: $\forall wd \in (18,24) \mapsto ndbh.$
Rule 6: $\forall wk \mapsto ndbh.$
Rule 7: $\forall \in \{, , , \} \mapsto sum.$
Rule 8: $\forall \in \{, , , \} \mapsto aut.$
Rule 9: $\forall \in \{, , , \} \mapsto win.$
Rule 10: $\forall \in \{, , , \} \mapsto spr.$
Rule 11: $\forall spr \land dbh \mapsto 20.$
Rule 12: $\forall spr \land ndbh \mapsto 15.$
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. 11.

![BPMN Diagram](image-url)

Fig. 11. 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 node? 4) Termination/completeness: Does the specification assure that the system will always terminate producing some 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.

4. 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.

5. Business Process Anomalies

There are two kinds of business process anomalies which can occur while process modeling, namely [19]: Syntactic anomalies and Structural anomalies [21].

5.1 Syntactic Anomalies in Business Process

This kind of anomaly is not dependent on the data type or tokens in activities. The problem is improper utilization of modeling elements. 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: a) Incorrect usage of Flow Objects; b) Incorrect usage of Connecting Object; c) Incorrect usage of Swimlanes.

5.1.1 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. 12.

This modeling approach is undesirable and could lead to misinterpretations.

Fig. 12. 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. The example (Fig. 13) shows a process in which after receiving an order, it shall be checked whether the order is valid. The control flow depends on the validity of the order.

Fig. 13. Data-Based XOR Gateway.

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). It is also possible that none of the awaited events will occur, so it is recommended to model also a Timer-Event with represents a Timeout situation. If you are not satisfied given conditions it is incorrect to use of Event-Based XOR Gateway (Fig. 14).
5.1.2 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 [22].

5.1.2.1 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. 15).

5.1.2.2 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. 17).

5.2 Structural Anomalies

Structural anomalies have been described in the literature [23], [24], [25], [26] they are classified as four types: a) Deadlock; b) Lack of synchronization; c) Dead Activity; d) 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 [27] 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. This example illustrates two types of deadlock: positive (Fig. 18) and negative (Fig. 19).

Fig. 14. Event-Based XOR Gateway.

Fig. 15. Missing sequence flow.

Fig. 16. A Sequence flow May not cross pools boundaries.

Fig. 17. Two Lances are used as two Pools.
A deadlock is a situation where the flow of the Process cannot continue because a requirement of the model is not satisfied. For example, if a Parallel Gateway is expecting a token from all of its incoming Sequence Flow and one never arrives, the process will be trapped with deadlock. There are two types of deadlocks: deterministic deadlock (Fig. 20) and non-deterministic deadlock (Fig. 21).

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. 22) 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 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 ), [26]. While Dead Activities are activities which will never be executed. A last type of anomaly is Infinite Loop [14], 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 [27].

6. Conclusions
Business process modeling is associated with the need for graphical representation of business processes for their optimization and archiving. Business Process Model and Notations (BPMN) a popular modeling language. BPMN is the global standard for process modeling and one of the most important components of successful Business. 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.

7. References