

# BUSINESS SYSTEM AS AN EQUILIBRIUM OF INTENTION AND CAUSALITY

Václav Řepa

## Introduction

More than 20 years after the idea of business process-driven management was born there still exist essential insufficiencies in the understanding of how business processes should be modeled in order to respect this idea. These insufficiencies are manifesting themselves even in the existing modeling standards. As the modeling of business process is a necessary first step in the process of implementing the process-driven management idea, this situation seems to be a main barrier of the needed putting this idea into the real life.

On the other hand, there are existing theories, methods and ideas, already known in other contexts, which directly address the problems and features behind the above mentioned insufficiencies of the current state. As usually, the root of the problem is not the insufficient basic knowledge but rather the insufficient realization of the proper context.

The aim of this article is to draw attention to the essential features of the business processes and business systems in the context of their modeling. The needed reflection of these essential features in the modeling language and methodology will be analyzed. In addition, the basic insufficiencies of the contemporary approaches to the business process modeling will be pointed out together with the outline of possible ways of their overcoming.

As a basis of the main principles used in considerations in this article we use the MMABP methodology Řepa (2003), OpenSoul project (2015). MMABP (Methodology for Modeling and Analysis of Business Processes) is a 'language independent' methodology based on the set of meta-models which define the basic concepts and express the basic principles of the methodology, and completed with the set of techniques, consistency rules and patterns. MMABP is generally open in terms

of principal ability to be completed with newer concepts, principles, techniques, etc. if they are consistent with its principles and the meta-models. As the MMABP is based on meta-models instead of particular languages it can be also used as a basis for the evaluation of any modeling language towards the principles.

The root idea, this article is based on, is that business process is an equilibrium of intention and causality. This idea is based on the fact that business process always represents the way of following some intention. Every business process has the goal which represents the meaning of the process. Every action performed in the process than should be targeted to its goal. The above described features of the business process mean that business process is always subjective. On the other hand, business process always exists in some environment which defines the conditions that have to be respected by the process actions. These conditions have different forms: the restrictions of possible actions, time and other types of limits, admitted of prohibited consequences, etc. Although the conditions become from different resources: the nature itself, legislation, habits, or specific local restrictions or features, they all have to be respected in the process unconditionally. From the process point of view all conditions given by the process environment are objective. Therefore, we call the summary of all these mandatory conditions, no matter where they come from, the causality of the business system.

Concluding from previous two paragraphs one can see the business process as a way of following some intention in the environment which particularly restricts its actions. The correct business process thus must balance between its purpose and the given causality of the environment.

The article is divided into five sections. In the second section, after this introduction, we

follow the root ideas of cybernetics in order to explain the concept of intentionality and its consequences in business system modeling as well as its impact on the business process modeling methodology and language. Third section is focused on the concept of causality in business systems. Relevant informatics theories and techniques for modeling the real world causality are introduced in this context. In the fourth section the relationships between both basic types of the business system models (intentional business process versus causal ontology model) are discussed together with the methodological consequences. In this section we also introduce basic consistency rules from the MMABP methodology. In the final section we summarize the main ideas from the article, discuss connected insufficiencies of the current approaches and outline some basic task for the future development.

### 1. Intentionality in Business System

In the legendary article Rosenblueth, Wiener and Bigelow (1943) authors expressed the idea which essentially influenced the later development of cybernetics: *'all purposeful behavior may be considered to require negative feed-back'*. The concept of negative feed-back is explained there as follows: *'...the behavior of an object is controlled by the margin of error at which the object stands at a given time with reference to a relatively specific goal. The feed-back is then negative, that is, the signals from the goal are used to restrict outputs which would otherwise go beyond the goal'*.

According to the basic work in the field of process-driven management (Hammer, & Champy, 1993) business process always follows some goal. The goal is a fundamental attribute of a business process as it is regularly used in matured methodologies like in Eriksson and Penker (2000) for instance. That means that business process is always an intentional process. By the term intentional process we mean the process of purposeful behavior of interested object following some goal.

Concluding from previous two paragraphs one can find that the business process, as it is an intentional kind of process, have to have some negative feed-back which ensures restriction of its outputs in order to keep them in the margins of its goal. This characteristic strongly distinguishes the business process from the process in general (i.e. in just technical

/physical sense) as well as from processes which do not need any feed-back like machine-managed or automated processes running without a contact with their environment.

In the case of business process the feed-back is represented by the input to the process from its environment which is causally connected with some process output. The value of the input should influence the following behavior of the process in terms of keeping it in the margins of its goal. This means that 'intermediate' inputs to the process (i.e. none-starting inputs to the process coming between its starting and end points) are critically important parts of the business process distinguishing it from other, non-intentional (i.e. non-business), processes. Working with processes we have to take into the account even the time dimension; every input to the process from its environment has to be synchronized with the process run. Thus, in each part of the process where some input which influences the following process run is expected the process state has to be placed. Process state means such point in the process structure where nothing can be done before the input to the process occurs, i.e. point of waiting for the input.

#### 1.1 The Concept of Process State in the MMABP Methodology

The ideas presented in this article are systematically elaborated in the MMABP methodology (Methodology for Business Processes Analysis and Design). MMABP distinguishes between two main types of models: business process versus business object models. In both types of model the methodology recognizes the global model (system view) and the detailed model. The modeling tools used by the methodology are based on common standards BPMN (Business Process Model and Notation (2011)), UML (Unified Modeling Language™ Infrastructure Specification version 2.4.1 (2015)), and Eriksson/Penker Notation (Eriksson and Penker (2000)). The essence of the methodology is defined in the formal meta-model (Business System Modeling Specification (2015)) as a part of the development project OpenSoul (OpenSoul project (2015)). The meta-model consists of three main packages which correspond to the main dimensions of the business system model: Business Substance Meta-model, Business Process Meta-model, and Business Models Consistency Model.

The 'intentional' part of the business system is a subject of the Business Process Meta-model (see Fig. 1). Business Process Meta-model defines the essential concepts in the field of business processes and their relationships, and express the principles which the modeling of business processes should be based on.

As it is visible from the meta-model the concept of 'state' is, together with 'stimulus' and 'activity', one of the basic building blocks of the business process description. Meta-model also expresses its essential relationships to other basic concepts:

- State is a mandatory consequence of both types of the process activity (processing and control, the difference is only in the cardinality of the relationship which depends on the type of the activity). This principle expresses the fact that every process activity (i.e. an action performed in the process) needs to be followed by waiting for the reaction from outside the process (a feed-back).

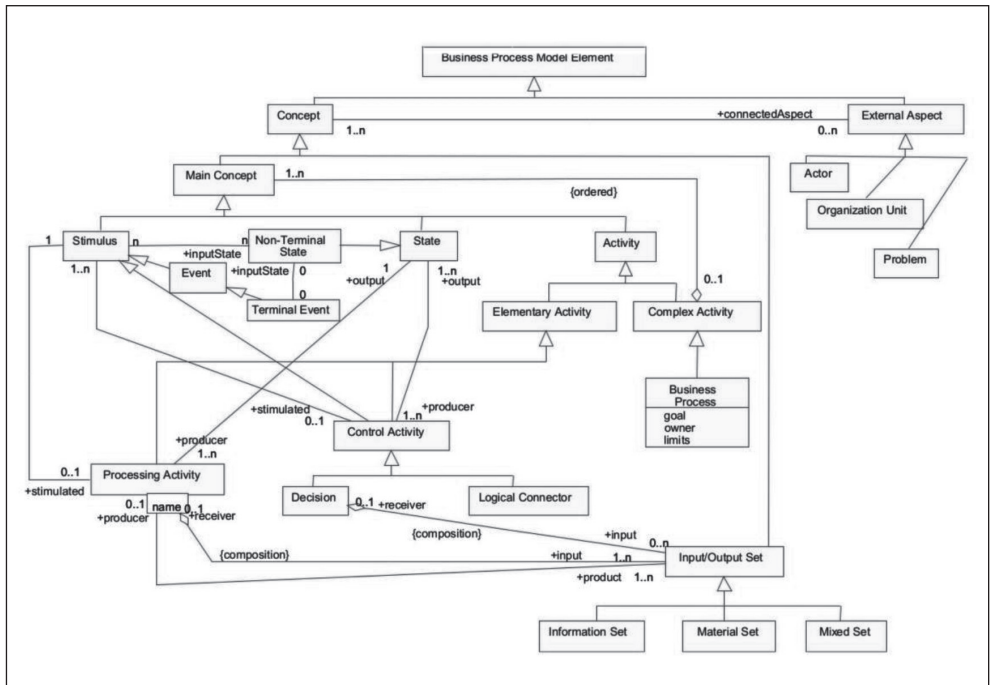
- State has to be connected with at least one 'event' type of 'stimulus'. This principle expresses the fact that following process activity can be performed after the reaction from outside the process (a feed-back) occurs. The feed-back comes to the process in the form of 'event'.

The definition of basic principles in the meta-model is absolutely exact, nevertheless (and therefore) it is very hardly readable. Moreover, the meta-model in just defining the principles but does not explain their meaning and purpose. Therefore, the meta-models in the MMABP methodology are completed with partial, not so exact but more easily readable, definition of some principles in the form of so-called process patterns.

There are two main kinds of Business Process Patterns in the MMABP:

- Basic Business Process Flow Pattern** which defines the basic procedure and decision points of the process of model

Fig. 1: MMABP Business Process Meta-model



Source: OpenSoul project (2015)

creation. This pattern is essential in the MMABP, it expresses the main principles, rules, and other aspects of the MMABP approach to the business process modeling. It also defines and explains the role of the process state in the process description and therefore it is explained in detail in this article.

- **BP patterns for particular situations** which cover typical situations frequently occurring in business process models where it is possible to find some generally valid structures, principles and constructions which should be fulfilled by the process description undoubtedly. There are several most general process patterns in the MMABP:
  - **Complementary Events Pattern,**
  - **Repeating State Pattern,**
  - and some other specific patterns.

The first two mentioned complementary patterns for particular situations are also connected with the problem of process intentionality through their focus on two main concepts relevant to the problem: state and

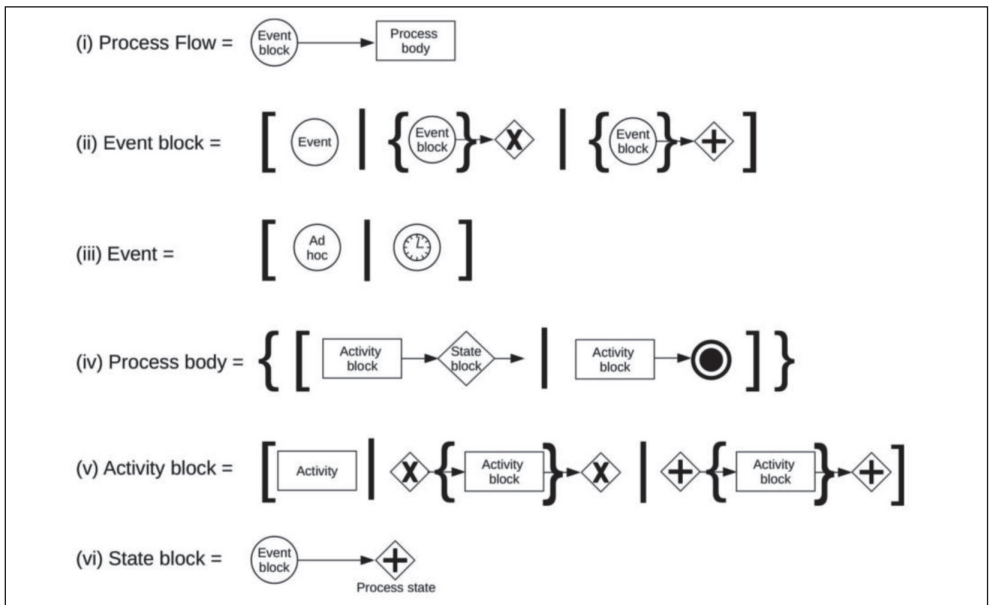
event. Thus, these two patterns are also explained in detail in following text.

### 1.2 Basic Process Flow Pattern

Basic Process Flow Pattern expresses the basic structure of the process model which respects the essential rules of the MMABP methodology. These rules express the ‘technical’ necessities which mainly follow from the general theory of algorithms as well as the specific aspects of the business process which distinguish the business process from a process in general (i.e. just technical) sense. The lately mentioned rules follow from the theory of BP management and re-engineering which is anchored already in the basic work in this field: Hammer and Champy (1993), Davenport (1993). Basic Process Flow Pattern (see Fig. 2) expresses the essence of the process flow using three methodically essential types of the process elements: events, activities, and states.

According to the Basic Process Flow Pattern the business process should be described as a sequence of *Activity blocks* interrupted by *State blocks* starting with just one *Event block* (starting event) and resulting in

**Fig. 2: Definition of basic blocks and concepts of the BPF Pattern**



Source: Business System Modeling Specification (2015)

one or more *End states*. The definition is written in the semi-formal metalanguage based on the simplification of the standard Extended Backus-Naur Form (for details and explanation of the EBNF see Syntactic Metalanguages ISO/IEC International Standard 14977 (1996)). Used meta-symbols have following meanings:

- $A = [element1 | element2 | element3]$  means that the item *A* can be either *element1* or *element2* or *element3* exclusively.
- $A = \{ elementX \}$  means that the item *A* consists of one or more *elementsX*.

Particular definition sentences can be read as follows:

- Def (i): Process flow begins with starting Event block followed by the Process Body.
- Def (ii): Event block is either a single event, or structure of mutually exclusive Event blocks, or structure of mutually synchronized Event blocks.
- Def (iii): Event can be either an ad-hoc event or a timer.
- Def (iv): Process body consists of one or more pairs where each pair consists of an Activity block followed by either State block or End State. If the pair ends with State block the description should continue with another pair (see the arrow after the State block). End state always means the end of the process.
- Def (v): Activity block is either a single Activity, or structure of mutually exclusive Activity blocks, or structure of parallel Activity blocks.
- Def (vi): State block is a synchronization of internal process flow with expected event(s) expressed as an Event block (in other words: waiting for the event(s)).

**Event block** represents the external influence which the process always has to respect. It works either as a trigger or a limiter of the process. In both cases it has to be unambiguous which means, among others, that is has to represent a single point of time. Therefore, it can be either a single event or a time-elementary structure of events. If it is a structure it can express either the synchronization of parallel events (event blocks) or the set of possible mutually exclusive

alternative events (event blocks) in order to be time-elementary.

**Activity block** represents an action element of the process. It can be either a single activity or a structure of activities (activity blocks). Similarly, as in the case of event also an activity should be unambiguous. Therefore, if it is a structure, it can express either the synchronization of parallel activities (blocks) or the set of possible mutually exclusive activities (blocks). It cannot express a sequence of activities as it would be a violation of the elementariness rule. The methodical reasons and meaning of the need for the elementariness of activities in the process description is discussed in more detail below.

**State block** represents the essential need to synchronize the process run with expected events. This need follows from the fact that the event is always an objective external influence and thus it must be respected. From the physical point of view such respect means synchronization – waiting for the event (event block). As the BPMN notation do not recognize the concept of process state there is no other way than to express the process state with the general symbol for synchronization – the ‘AND gate’. In order to distinguish between the general synchronization and its specific meaning as a process state we complete the BPMN with the stereotype <<process state>>.

One of the most important ideas expressed in this pattern is that there *cannot be a sequence of process activities uninterrupted by the process state*. This rule reflects the essence of the definition of an elementary process activity:

(a) the process activity is regarded as elementary if there is no objective reason for its interruption,

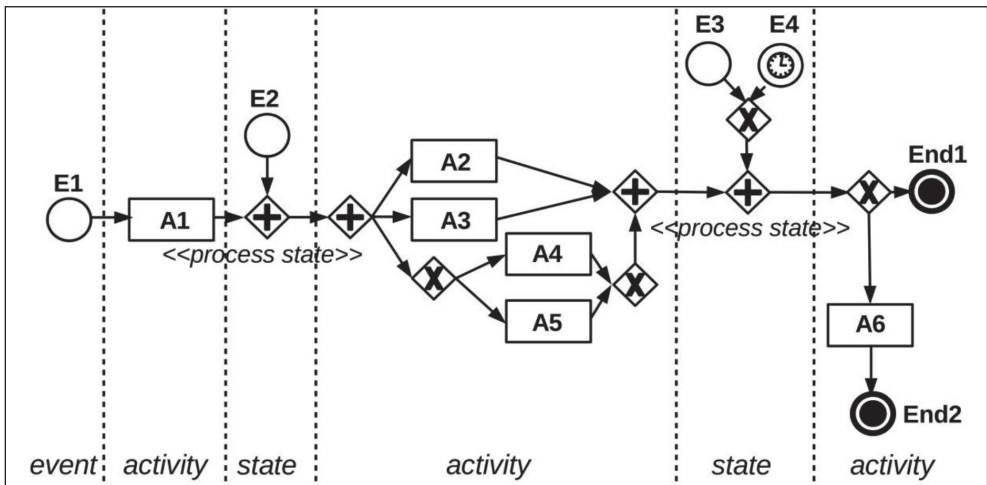
(b) the reason for the interruption of the activity is objective if it comes from outside of the process.

Rule (b) of this definition means that each objective reason for the process interruption is represented by an event (external influence) in fact. Thus, any activity of the process, no matter how technically complex it is, must be regarded as elementary if there does not exist an external influence (event) which the process has to respect (i.e. wait for). This consequence well illustrates the fact that the elementariness of a business process activity is not only its physical but much more a functional attribute as

the business process itself is always more than a physical process (algorithm) only. This way the methodology prevents the analyzer from the pointless unlimited dividing of the process activities which is a frequent mistake in the field of business process modeling. The necessity of such safety fuse in the methodology against the unlimited division of activities is given by the fact that in the field of process-oriented modeling the aggregation is a dominating type of abstraction (unlike in the field of object-oriented modeling

where the generalization is a dominating type of abstraction). This fact manifests itself in the principally unlimited possibilities of division of activities known as a rule: any single process activity can be decomposed into the structure of sub-activities – a process (as it is also defined by the process meta-model at Fig. 1). As the division of activities is physically unlimited the methodology has to define some logical – functional definition of the very low level: the level of the process elementariness.

Fig. 3: Correct business process flow example



Source: own

Fig. 3 shows the symbolic example of the process which can be regarded as correct according to the Basic Business Process Flow Pattern. The process can be seen as a sequence of several parts each representing one block of a particular basic type (see the division of the whole process by vertical dashed lines). It is beginning by the starting event block which consists of just single event E1 in this case – the starting event of the process. The starting event is followed by the activity block consisting of just single processing activity A1. According to the pattern the activity block is followed by the state block in the form of synchronization of the process run with just a single event E2. Following activity block represents more complex structure of activities: it consists of

the structure of three parallel activities where the first two are single processing activities A2 and A3, and the last one is a structure of two alternative processing activities A4 or A5. Following state block represents the waiting for two alternative events E3 or E4. The last activity block is a structure of two alternatives: the processing activity A6 followed by the process end End2 or the immediate process end End1.

The example in the Fig. 3 illustrates that and how any algorithmic structure of the process can be checked whether it fulfills the basic definition of the business process expressed by the Basic BP Flow Pattern: *business process is a sequence of Activity blocks interrupted by State blocks starting with just one Starting Event block and resulting in one or more End states.*



In the MMABP the Basic Process Flow Pattern is completed by a number of process patterns which model the specific typical situations which are often occurring in business process models. These situations are generalized and should be therefore instantiated for the use in the particular process.

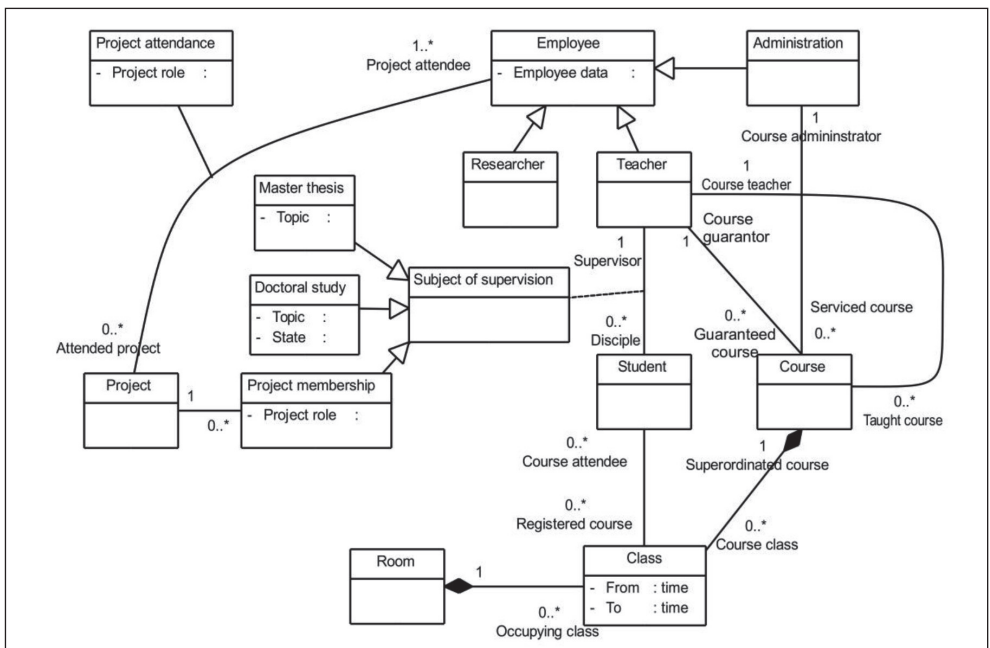
## 2. Causality of Business System

By the term 'causality' we understand an aggregate of basic facts, conditions, rules, relationships, presumptions, consequences, and other aspects of the given business system which can restrict the possible actions in a business process. As it follows from this definition, the causality of the business system is always superordinate to any action in the system, to any activity of any business process.

The traditional informatics technique for description of the real world causality is called Conceptual Modeling. The origin of conceptual modeling is closely connected with the database technology. The original purpose of the *technique for data modeling* was to create

the concept of the database whose structure is so flexible to be able to accept as much as possible needed future changes. The need for future changes of the database comes from the changes in the real world. As the real objects and their relationships are naturally changing, the contents of the database should change as well. The root motivation for data modeling is the need to design the database which would be able to accept as much as possible necessary changes in the real world without the need to change its own structure. Chen (1976) expressed the crucial idea that these criteria best fulfill such structure of the database which is similar as much as possible to the structure of the real world concepts. Therefore, the modeling technique has to be primarily aimed on analyzing the structure of the real world concepts. For that purpose he proposed a data model, called the *entity-relationship model*, which incorporates the important semantic information about the real world by modeling the real world concepts and their relationships. *Conceptual Modeling* was born.

Fig. 4: Example of the system model of objects (Class Diagram)



Source: own

Since the mid-nineties the *Class Diagram* from the UML (Unified Modeling Language™ Infrastructure Specification version 2.4.1 (2015)) has been accepted by the informatics community as a standard tool for conceptual modeling. Class Diagram describes the static structure of the system in the form of classes and relationships among them. Class Diagram is a basic diagram of the whole UML repertoire of diagrams. In the following decades there have come the new generation of the Real World modeling which can be also regarded as a successor of the traditional conceptual modeling: *ontology modeling* and *ontology engineering*. A comprehensive description and explanation of relationships among the conceptual modeling, ontology modeling and the UML can be found in Guizzardi (2005).

We understand the term 'system of business objects' to mean the global view of objects. This model expresses which objects in which mutual relationships form the business system. For an example of a particular system of business objects see Fig. 4 above. In principle, the system view of objects can simply recognize their existence and mutual context, not their dynamic details. This model is a conceptual model in the traditional meaning, in fact. Particular object classes in the model represent concepts which identify possible real objects from the business system. Relationships among object classes then identify possible links among real objects from the business system. Nevertheless, both real objects and their mutual links are naturally dynamic in the real world; they are changing in time. These dynamic aspects cannot be described in this model as it principally only represents a static view of objects. For the description of object's dynamics the detailed objects model (life cycles of objects) is intended.

One of the main reasons for using the UML Class Diagram as a standard is the fact that this diagram, as a part of the consistent system of diagrams UML, allows direct linking with the detailed object models – object life cycles in the form of UML State Charts.

## 2.1 Modeling Dynamics of Objects (Object Life Cycles)

Like in business process models, even in the field of business objects there is a need for a detailed view on some particular objects. Like in the case of business processes, even

the detailed view of a business object means viewing the object as a process. This process represents everything that can happen during the lifetime of the particular instance of the object class. Therefore, this detailed model of an object is called the *object life cycle*.

The object life cycle expresses the internal dynamics of each object of given class. It describes the mechanism of the object evolution during the time. As the tool for the Object life cycle description, the MMABP methodology uses the State Chart diagram from the UML (Unified Modeling Language™ Infrastructure Specification version 2.4.1 (2015)). MMABP regards the State Chart as a most suitable tool from the Unified Modeling Language for the purpose of the object life cycle description. Nevertheless, the State Chart has not been originally intended as a tool for description of life cycle. Its roots are in the field of state machines theory, and it is closely connected with the concept of so called 'real-time processing'. However, the concept of the state machine in general is not substantially reducible to just the area of real-time processing. There is also a need for recognizing the states and transitions among them in the area of data processing. The best proof of this idea is the concept of the object life cycle itself – once we think about the objects generally (i.e. in terms of their classes), then we have to strongly distinguish between the class and its instance. In the case of the object life this requires determining those points in the life of all objects of the same class, which we will be able to identify, and which it is necessary to identify in order to describe the synchronization of the object life with life cycles of other objects. Such points of the object life are its states. So each object instance lives its own life while the common structure of lives of all instances of the same class is described as a common life cycle of the object class.

Fig. 5 shows the example of the life cycle of the object class Teacher from the model at Fig. 4. For the understanding of the life cycle contents the relationships of the Teacher to other object classes from the ontology model are essential. There are four main states in the life of the Teacher:

- The state *Vacant* means that the teacher does not supervise any student and does not teach or guarantee any course (i.e. has no relationships to objects *Student* and *Course*). There are two possible ways from



this state: either the new student or course is assigned to the teacher which causes the transition to the state Occupied or the teacher is released (transition to the finite state Released).

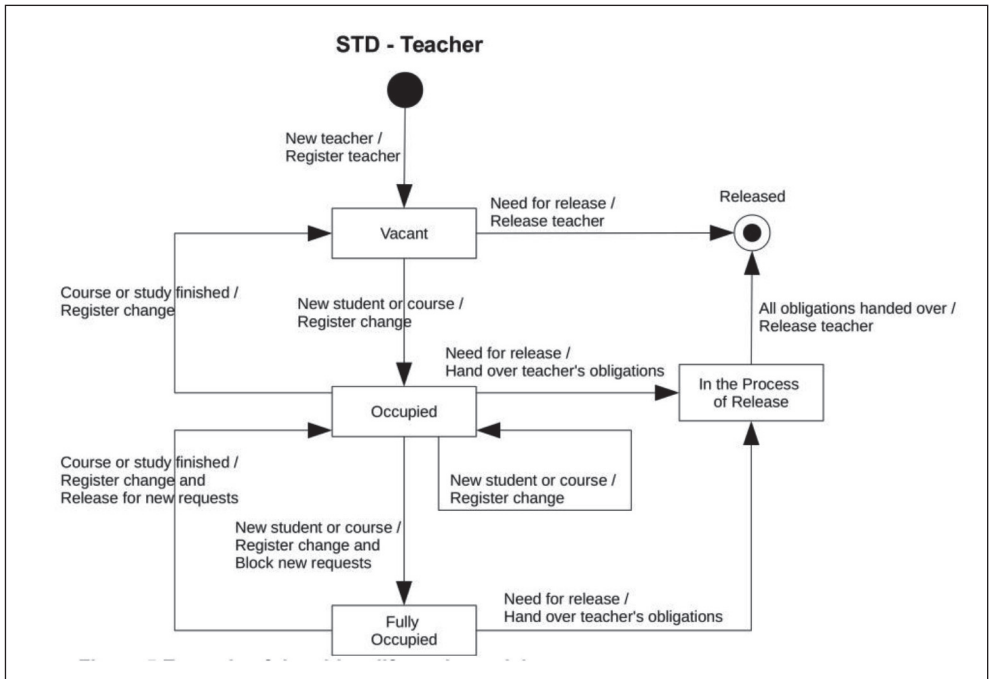
- The state *Occupied* represents the situation when the teacher supervises some students and/or teaches or guarantees some courses but he/she still has a free capacity for more students and/or courses (i.e. has some relationships to objects *Student* and *Course* but still in the given borders of his/her capacity). From this state the teacher can transit to any state including itself (see the self-transition in the case of the change which does not overflow the teacher's capacity and does not mean lose of the last student and/or course).
- The state *Fully Occupied* represents the situation when the teacher has no more capacity for any student and/or course. From this state the teacher can transit just back to the state *Occupied* or to the finite

state *Released* through the mid-state *In the process of release*.

- The state *In the process of release* is a necessary mid-step from the states *Occupied* and *Fully Occupied* to the finite state *Released* as the release of the teacher requires prior hand over of all supervised students as well as taught and guaranteed courses (i.e. redirecting all existing relationships to objects *Student* and *Course* to other *Teachers*).

Each described life cycle has to correspond to the particular object class in the Class Diagram. In such way the State Chart specifies the general mechanism of the life of all possible instances of the given class. Described states and transitions among them consequently correspond to the attributes and methods of the class. In fact, life cycle states represent the specific attribute of the class (no matter whether this attribute is present in the class description or not it always exists by

Fig. 5: Example of the object life cycle model



Source: own

the definition – it is necessary to distinguish among particular states/values of this ‘hidden’ attribute). Each transition between life cycle states then represents the use of the particular class method.

Every transition between states is described by the pair of attributes divided by slash (see Fig. 5). The first one represents the reason for the transition (why), the second one represents the way of the transition (how). The reason for the transition is an event (i.e. some external influence of the given object) while the way of the transition is a method of the given object (i.e. some action performed by or applied to the given object). The concept of events, as a common concept existing in both main points of view on the Real World dynamics, allows linking of the description of object life cycles with the description of business processes. Events used as reasons for transitions in object life cycles are the same events as those which trigger business processes. This way the *object life cycle serves as a bridge between intentional models of business processes and the description of the system causality in the form of the ontology model.*

Although the object life cycle as well as the business process are both process descriptions (description of the dynamics), there is a dramatic difference between them in the meaning of the ‘process’ concept. During the Real World modeling it is necessary to clearly distinguish between the ‘business process’ and ‘process in general’ concepts. On one hand it is necessary to model just the Real World processes and not the infrastructure processes (i.e. ‘software processes’, organizational procedures, performance of IS, etc.). On the other hand, the model of objects also describes the behavior – in the form of entity life algorithms (ordering of methods). Such behavior is seen from the point of view of objects and their relationships. It does not represent any intention. So, the ‘behavior of objects’ should be regarded as a structural aspect of the real world, i.e. something completely different from the business process which is intentional in principle. In the form of the process the Object life cycle thus describes just the set of rules which are given by the essence of the business and therefore have to be respected by all objects of the given class (so-called business rules). Unlike the process of the object life (object life cycle) representing the general internal logic of the object behavior,

the business process always represents some external business goal or any other form of intention. From the point of view of the object the business process thus expresses the *intentional combination of actions from the object’s life.*

### 3. Harmonizing Intentional Processes with the Causality of Business System

The main idea this article is based on is that business process should be taken as an equilibrium of intention and causality. Equilibrium is never axiomatic. Keeping a system in the equilibrium state requires permanent effort. Ensuring the equilibrium of intention and causality in business system models requires permanent *harmonization of both main business models* in order to ensure the *consistency of the process model with the business ontology.*

General overview of the consistency of business system models can be found at Fig. 6. There are two main general kinds of consistency:

- Correctness.
- Completeness.

**Completeness** means that nothing essential is missing in the model.

**Correctness** means that nothing in the model is in contradiction with anything in this model as well as in other models of the same system.

Fig. 6 also shows that the concept of consistency is relevant for a single model (i.e. internal consistency of the model, consistency of different parts of the same model) as well as for the whole system of models. Nevertheless, the difference between completeness and correctness can be clearly distinguished just in the single model. In the case of cross-models consistency (consistency of the parts of different models) both types of consistency coincide.

For instance, completeness of the object life cycle is defined as follows: ‘life cycle has to cover the whole life of the object since the moment of its nativity until all different kinds of its possible death’. Similarly, completeness of the business process is defined: ‘business process has to cover the whole process since the moment of the starting event until all possible ends of the process’. General completeness of the conceptual model is defined: ‘every class defined in the conceptual

model has to be associated to at least one another class of the same model'. In all bilateral and even more in the trilateral intersection of models the completeness is always relative to other model(s) and therefore may be viewed as a specific kind of correctness. For example, completeness of object roles (intersection of the Class Diagram and Business Process Diagrams) means that all objects from the Class Diagram have to occur in at least one role in at least one process. In the same time it means that every role from business processes has to be represented as a class or relationship, or the combination of both in the Class Diagram. Completeness thus always depends on what is expressed in other connected diagrams, it cannot be more regarded as absolute and the clear difference between completeness and correctness consequently disappears.

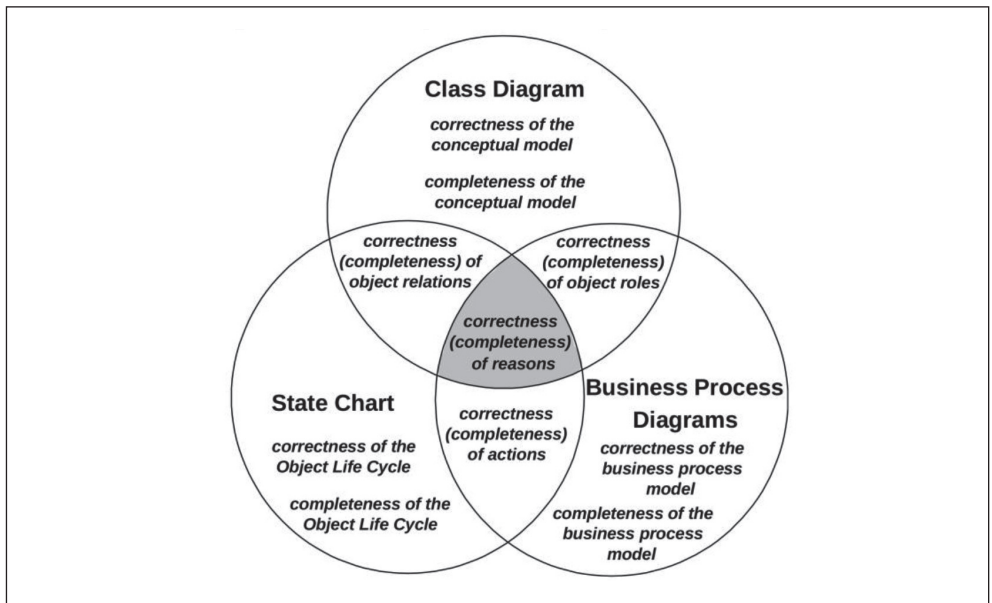
Despite this, from the methodology point of view the classification of these two main meanings of consistency remains the powerful tool for creating particular consistency rules.

Rules for internal completeness and correctness in a single model always follow from the basic definitions in the given domain:

ontology, business processes, and life cycles (see mutually exclusive parts of sets at Fig. 6. In contrary, the cross-models consistency rules are always connected with specific meaning of the common concept of given set of models (see mutual intersections of the sets at Fig. 6):

- Common concept of the global model of objects and detailed view on the particular object class is the *relation*. Relationships to other objects how they are visible in the Class Diagram are manifesting themselves in transitions of states in the object's life cycle. The transition usually means creation, deletion or some modification of the relation to some particular other object.
- Common concept of the global model of objects and detailed view on the particular connected business process is the *role*.
- Common concept of the particular object life cycle and detailed view on the particular connected business process is the *action*. The meanings of actions which are the basic building blocks of the business process directly correspond to the meanings of actions by whom the object can transit between states (methods of the object).

Fig. 6: Consistency of business system models



Source: own

- Common concept of all three basic views on the business system is the *reason*. Reason is represented by the concept of *event* which is the only concept existing in all views: as a trigger of the process actions, trigger of transitions among states in the object life cycles, as well as the trigger of any change in the ontological status of the business system (creation, expiration or modification of objects and/or their relationships).

### 3.1 Structural Consistency

Besides the regular consistency rules based on the existence of diagram elements MMABP recognizes also the specific kind of consistency – so-called *structural consistency*. In this kind of consistency we work not only with particular elements of models but with their structures. The idea of structural consistency is based on the Jackson's idea of the *correspondence of structures* (Jackson, 1997; Jackson & Cameron, 1983). Jackson (1997) introduced the method for designing the computer programs based on keeping the correspondence between the structure of data and the structure of program actions. He shows the essential dependency of the ordering of processing actions on the ordering of processed data. This idea is even more elaborated in (Jackson & Cameron, 1983) for the purpose of designing the program systems working with databases.

In MMABP we generalize this idea in order to make it applicable in all possibly relevant areas. As a basis for this generalization we use the general classification of two basic types of hierarchy of concepts:

- Generalization* where the superordinate concept is a generic concept which covers all subordinated concepts as its specific variants.
- Aggregation* where the superordinate concept is an aggregate (according to Jackson a *collective concept*) which covers all subordinated concepts as its parts.

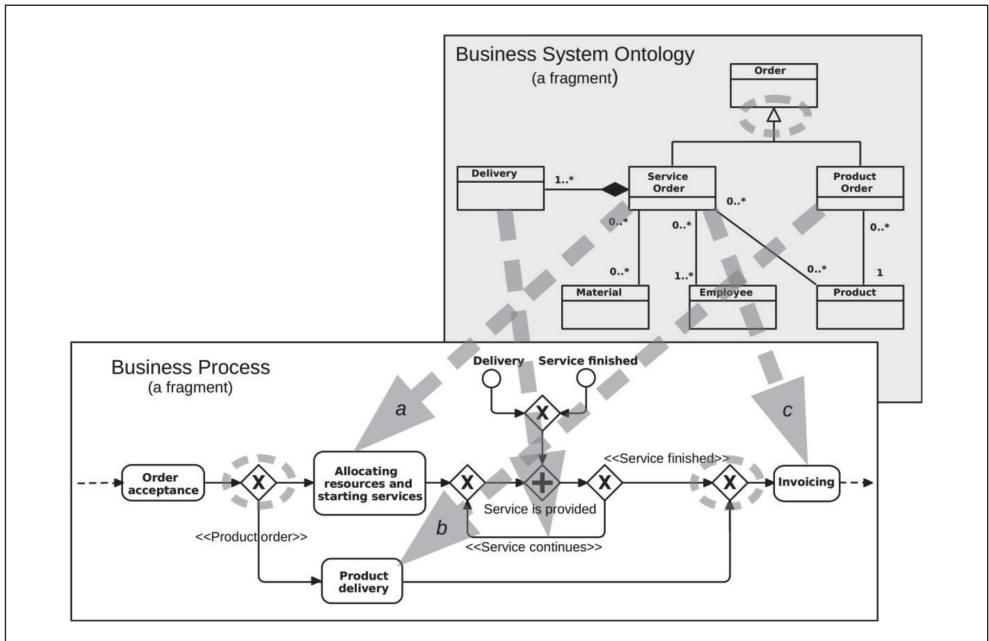
These two basic types of hierarchy can be watched in various forms in various models: as a cardinality (aggregation) or optionality (generalization) of relationships of concepts, as a fork (generalization) or a cycle (aggregation) in the process structure, etc. Any structure of elements in one model has to correspond to the structure of elements of the same generic type (generalization / aggregation) in other connected models (see example in the Fig. 7).

The example in the Fig. 7 shows how the business process structure should meet the structural aspects of the business system. There are fragments of two mutually connected models at the example: business process model and business system ontology (conceptual model). According to the business system ontology there are two mutually exclusive types of order: service order and product order. While the delivery of the *product* is a one-off action as the *product order* always contains just one *product* (see the cardinality of the association between these concepts), the *service order* can be fulfilled by a number of *deliveries* (according to the relationship between them the *service order* is an aggregate of *deliveries*). This difference is reflected in the process by two mutually exclusive ways of handling these two basic types of the *order* (see the thick gray arrows a and b between the models). The fact that the concept *order* is a generic concept which covers both concepts *service order* and *product order* corresponds to the way of handling these objects in the process. The actions *order acceptance* and *invoicing* which are connected with the generic concept *order* are principally common for both order types while actions connected with specific sub-concepts *allocating resources*, *product delivery* etc. are specific just for the given type of the *order* (see the thick gray ovals in both models and the gray arrow c). Similarly, the fact that the *service order* consists of potentially more *deliveries* is reflected in the process by the possible repetition of the state *service is provided*, i.e. repeated waiting for the delivery (see the remaining thick gray arrow).

This example also illustrates that the business system ontology is principally superordinate to any business process as the causality of the Real World is principally superordinate to any intention of its actors. For example, the fact that two basic types of *order* are mutually exclusive can never be changed by any intention, by any business process. Conversely, this fact has to be respected in the structure of any business process otherwise the process allows generally incorrect behavior of business actors.

On the other hand, the Real world causality is also a subject of evolution which the actors collaborate on as well. Thus, the intentions represented by business processes and their eventual conflicts with the Real World causality are important signals for needed (wanted / possible,

Fig. 7: Structural consistency (example)



Source: own

etc.) changes in the Real World. For this purpose the consistency rules which can emphasize these differences are very powerful tools.

For other views and classifications of the consistency see (Bruckner, Řepa, & Chlapek, 2014), relatively complete overview of the particular consistency rules can be found in Řepa (2010).

### Conclusions

In section 2. Intentionality in Business System we discussed the essential need to respect the principle of negative feed-back in process modeling languages. In case of business process the feed-back means that there is an input to the process from its environment which is causally connected with some process output. This part of the process which represents the communication with the environment in terms of its contents as well as time is called 'process state'. Process state means such point in the process structure where nothing can be done before the input to the process occurs, i.e. point of waiting for the input.

The concept of process state is present just in some process modeling standards (like IDEF, see (Mayer, Menzel, Painter, deWitte, Blinn, & Perakath, 1997) or Petri Nets based languages (Billington, Christensen, van Hee, Kindler, Kummer, Petrucci, Post, Stehno, & Weber, 2003)), partially present in some others (like ARIS methodology (Scheer, 1992) which mixes the concept of process state with the concept of event which is confusing and may even contradict with the idea of negative feed-back.), some standards do not support it. Widely accepted process modeling standard BPMN (Business Process Model and Notation, 2011) does not recognize this concept at all.

Regarding the importance of the above outlined problem together with the insufficient support in most of process modeling standards it can be said that the primary task for every process modeling methodology is to *allow the modeling of process states* in order to ensure critically important presence of the negative feed-back no matter which notation and/or modeling standard is used.

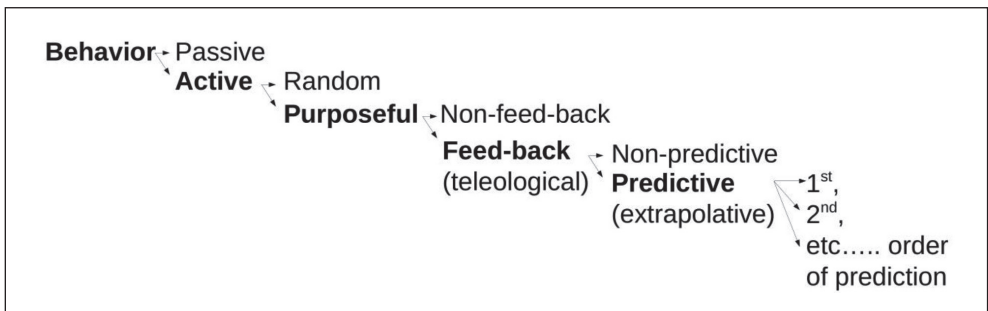
In the above mentioned essential article (Rosenblueth, Wiener, & Bigelow, 1943) the authors generally classify behavior in terms of explain its teleological meaning (see Fig. 8). Regarding the purposeful behavior from this point of view we have to take into the account even the feed-back. As the figure shows the feed-back may be either non-predictive or predictive. Non-predictive type of feedback represents just primitive reactions on events without the ability of the future improvement of the behavior based on the gathered experience. The authors use the example of the predator which follows its target exactly the same way as it runs. On the other hand, predictive feed-back requires from the subject of behavior the ability of some extrapolation of the future events.

The authors classify predictive behavior in different hierarchical orders. The first order of prediction can be understood as a simple use of the information about the followed target to predict the future changes connected with it. The second level of prediction means the ability of the subject to predict the changes of the chased target together with the consequences of its own behavior. The second order behavior allows making shortcuts to catch the target, for instance. The authors state that *'Predictive behavior requires the discrimination of at least two coordinates, a temporal and at least one spatial axis.'* In their essay they are solely focused on the movement of behaving objects in space. For the purpose of our analogy in the field of business processes we can better speak about a temporal and factual axes. Instead of just the spatial information the management of the business process generally requires

working with the information about important facts of any kind. So we can generalize this definition as follows: *'Predictive behavior requires the discrimination of at least two coordinates, a temporal and at least one factual axis.'* Particular orders of prediction then differ according to the number of 'factual axes', i.e. information sets (about different factual aspects) simultaneously considered in decisions. The topic of orders of prediction, generalized to the topic of orders of intentionality, became popular in the field of philosophy and related fields (see Dennett (1988) and Heyes (1987) for instance). This work offers a huge amount of inspiration for the field of business process management as well. Nevertheless, in this article we would like to outline just some basic challenges for the nearest development of the business process management methodology inspired just by the original classification of the first two orders of prediction from (Rosenblueth, Wiener, & Bigelow, 1943) which we regard as basic.

Business process modeling languages sometimes have problems even with the basic dimension – a time axis. Some languages do not support the modeling of this kind of process aspects sufficiently. Especially BPMN does not suppose a need for discrimination of this axis as it primarily takes the process as an automata (which is in direct contradiction with the main idea of the process-driven management, in fact). Consequently, it does not support well the coordination with other processes from the process internal point of view as it is discussed in the beginning of this section as a problem of modeling process states. This problem, as a consequence of the insufficient respect to

**Fig. 8: Classification of behavior from the teleological perspective**



Source: Rosenblueth, Wiener, & Bigelow (1943)



the time aspects of the intentional character of business processes, appears also in some other business process modeling languages.

Prediction of facts ('factual axis') always requires some knowledge about the causality of the real world. Even for the first order prediction it is necessary to know the supposed further consequences of the given facts. From the modeling methodology point of view it requires the permanent connection between the process and the ontology models. This need is a challenge more for the methodology and supporting computer tools than for languages. Currently, the connections between ontology and process models is possible just in some modeling tools and just on the basic – general level. It is possible to address objects from the ontology model in process models or to address in the ontology model the process activity connected with the given object, for instance. Nevertheless, most modeling tools do not support connections between UML and process models at all.

To achieve higher orders of prediction the business process has to permanently monitor itself and predict the consequences of its actions in the real world. Even this need is a challenge mainly for the methodology (how to model self-monitoring and which methodological consequences it has) and also for computer tools aimed on supporting the real-time management of process instances.

Summarizing the conclusions it is obvious that the idea of business system as an equilibrium of intention and causality brings following challenges in the field of business process modeling methodology and languages:

- Business process has to be always taken as a purposeful process respecting all consequences of this fact, mainly:
  - *Importance of the process goal.* This fact requires to strongly distinguish between the business process and the process in general. The need to act in order to achieve the process goal excludes regarding the business process to be automata for instance. Once the set of actions is automated it cannot be more regarded as a business process. From the business process point of view it is just a single action as there is no way to change its run therefore the only meaning of it lies in its function (i.e. predefined result). This fact also

requires two different points of view on business processes. Besides the process view when we see the process as a structure of actions there is the need for the contextual view when we see the process as a part of the system of processes (Global Process Model). Process goal and other global, relatively stable, process attributes (like output, information responsibility and others) have to be independent of the structure of actions which has to be dynamic (i.e. the subject of possible change) by definition as it follows from the idea of process-driven management.

- *Need for communication with its environment.* This need mainly supposes to distinguish between the process 'interior' and its environment ('exterior'). Therefore, it is necessary to regard any events as an external influence to the process. Besides the fact that it also excludes regarding the business process to be automata, it leads to the need for process states as points in the internal process structure where the communication with the environment occurs in the form of awaited event.
- Achieving the second and higher orders of prediction (as a main condition for needed purposefulness, intentionality) particularly requires the *self-perception of the process.*
  - The basic level of such self-perception is provided by the above mentioned process states which represent the information about what has been done in the process and what could be done in the future. Moreover, the process state represents the point of *sharing the responsibility for the process result with other processes* in both factual and time dimensions: the quality and time of the future process results are determined by the quality of awaited inputs and the particular time when the awaited event will come.
  - From the factual point of view the process also needs to use some additional general information about the environmental facts often called 'business rules'. Therefore, the *close relationship with the model of business causality - ontology model is necessary* as well. Even this relationship should cover both factual and time dimensions

of causality. Thus, it is necessary to include to the ontology model even the *process aspects of the causality in the form of so-called object life cycles*.

Many of above mentioned methodological challenges are reflected in the MMABP methodology (OpenSoul project, 2015; Řepa, 2003), other challenges will be reflected in its future development. Some reflection of them can be also find in other methodical resources in the field of business systems modeling (methodologies, languages and tools) Nevertheless, regarding the general validity of the root idea of this paper, the reflection of them should mainly become the regular part of the professional standards in the field.

*The article was processed with contribution of long term institutional support of research activities by Faculty of Informatics and Statistics, University of Economics, Prague.*

## References

Billington, J., Christensen, S., van Hee, K., Kindler, E., Kummer, O., Petrucci, L., Post, R., Stehno, C., & Weber, M. (2003). The Petri Net Markup Language: Concepts, Technology, and Tools. In *Application and Theory of Petri Nets, 2003, 24th International Conference, LNCS 2679* (pp. 483-505).

Bruckner, T., Řepa, V., & Chlapek, D. (2014). Consistency Issues in Large Business Process Model Environment, a Case Study. In *Proceedings of the BIR 2014 Conference, Lund, 2014*.

Business Process Model and Notation. (2011). *OMG Document Number: formal/2011-01-03*. Retrieved from: <http://www.omg.org/spec/BPMN/2.0>.

Business System Modeling Specification. (2015). Retrieved from: <http://opensoul.panrepa.org/meta-model.html>.

Chen, P. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems, 1*(1), 9-36.

Davenport, T. H. (1993). *Process Innovation: re-engineering Work through Information Technology*. Boston, MA: Harvard Business School Press.

Dennett, D. C. (1988). *The Intentional Stance*. MIT Press.

Eriksson, H. E., & Penker, M. (2000). *Business Modeling with UML: Business Patterns at Work*. John Wiley & Sons.

Guizzardi, G. (2005). *Ontological Foundations for Structural Conceptual Models* [Telematica Instituut Fundamental Research Series No. 15].

Hammer, M., & Champy, J. (1993). *Re-engineering the Corporation: A Manifesto for Business Revolution*. London: Nicholas Brealey Publishing.

Heyes, C. M. (1987). Contrasting Approaches to the Legitimation of Intentional Language within Comparative Psychology. *Behaviorism, 15*(1), 41-50.

Jackson, M. A. (1997). *Principles of Program Design*. Academic Press.

Jackson, M. A., & Cameron, J. (1983). *Systems Development*. Englewood Cliffs, NJ: Prentice Hall.

Mayer, R. J., Menzel, C. P., Painter, M. K., deWitte, P. S., Blinn, T., & Perakath, B. (1997). *IDEF3 Process Description Capture Method Report*. Knowledge Based Systems.

OpenSoul project. (2015). Retrieved from: <http://opensoul.panrepa.org>.

Řepa, V. (2003). Business System Modeling Specification. In H. Chu, J. Ferrer, T. Nguyen, & Y. Yu (Eds.), *Computer, Communication and Control Technologies (CCCT '03). Orlando: IIS, 2003* (222-227).

Řepa, V. (2010). Consistency in the Business System Model. In T. Družovec, H. Jaakkola, Y. Kiyoki, T. Tokuda, & N. Yoshida (Eds.), *Information Modelling and Knowledge Bases XXI*. Amsterdam: IOS Press (247-257).

Rosenblueth, A., Wiener, N., & Bigelow, J. (1943). Behaviour, Purpose and Teleology. *Philosophy of Science, 10*, 18-24.

Scheer, A. W. (1992). *Architecture of Integrated Information Systems: Principles of Enterprise Modeling*. Berlin, Heidelberg: Springer Verlag. doi:10.1007/978-3-642-97389-5.

Syntactic Metalanguages ISO/IEC International Standard 14977. (1996), ISO/IEC.

Unified Modeling Language™ (OMG, UML) Infrastructure Specification version 2.4.1 (2015). Object Management Group (OMG). Retrieved from: <http://www.omg.org>.

**prof. Ing. Václav Řepa, CSc.**

University of Economics, Prague  
Faculty of Informatics and Statistics  
Department of Information Technologies  
[repa@vse.cz](mailto:repa@vse.cz)

## Abstract

**BUSINESS SYSTEM AS AN EQUILIBRIUM OF INTENTION AND CAUSALITY****Václav Řepa**

*The article is aimed to draw the attention to the essential features of the business processes and business systems in the context of their modeling. We follow the root ideas of cybernetics in order to explain the concept of intentionality and its consequences in business system modeling as well as its impact on the business process modeling methodology and language. Possible way of reflecting these ideas in the business processes modeling methodology is outlined using the example of the process meta-model and business process patterns from the MMABP methodology. Then the concept of causality in business systems is explained and relevant informatics theories and techniques for modeling the real world causality are introduced in this context. Particular attention is paid to the topic of relationships between both basic types of the business system models: intentional business process and causal ontology model. General rules and principles of the consistency of models are discussed together with their methodological consequences. Basic types of the consistency of models – completeness and correctness – are identified and also the specific topic of 'structural consistency' is introduced in this context. In the conclusions section the needed reflection of these essential features in the modeling languages and methodology is analyzed and the basic insufficiencies of the contemporary approaches to the business process modeling are pointed out together with the outline of possible ways of their overcoming. As the main challenges in the field of business process modeling methodology and languages we particularly identify the need for respecting all consequences of the fact that business process has to be always taken as a purposeful process as well as the need for implementing the self-perception of the process in order to allow it achieving the higher orders of prediction.*

**Key Words:** Business process management, business process modeling, conceptual modeling, ontology, object dynamics, cybernetics.

**JEL Classification:** M10.

**DOI:** 10.15240/tul/001/2017-4-015