Overcoming expressiveness deficit of business process modeling languages[1]

Igor G. Fiodorov
Associate Professor, Department of Applied Information Technologies and Information Security
Plekhanov Russian University of Economics
Address: 7, Nezhinskaya Street, Moscow, 119501, Russian Federation
E-mail: Igor.Fiodorov@mail.ru

Abstract

Y. Wand and R. Weber have suggested that the ontological clarity of the modeling language can be evaluated by comparing the alphabet of this language with the constructs of top level ontology known as Bunge-Wand-Weber (BWW). According to them, one of the key success factors of using a given language is its ability to provide the users with a symbol set, which can directly reflect appropriate ontology concepts. However, the ontology is not limited to a thesaurus; it also covers the structure of relations between concepts. It may be assumed that the modeling language must be able to convey these relationships. Therefore, the approach of Y. Wand and R. Weber can be significantly enhanced if the structural relationships among BWW ontology concepts are studied. This paper also makes an attempt to extend the BWW ontology as applied to business process modeling, since in its current form it does not make it possible to represent logical operators and the temporal characteristics. We enhance the BWW ontology with transformations which change mutual properties, they correspond to logical operators. The interpretation of the event concept is modified such that it designates the moment in time when the object state changes. It is demonstrated that external events are connected to each process operation. Thus, the items of temporal logic: the moment in time and time interval between two consecutive events are added. The investigation of relations among enhanced BWW ontology concepts made it possible to substantiate five perspectives of the process model and identify formalisms used for their description, i.e. informational – entity-relation diagram; behavioral – state transition diagram; transformational – dataflow diagram; temporal – event graph; logical – ordinary Petri nets. Multiple research shows that process modeling languages and notations are not able to display immediately all BWW ontological model concepts, but only part of them. Moreover, the authors of these researches focus their attention on a percentage ratio of modeled and unmodeled concepts, calculate a relative degree of deficit, redundancy, excess and overload. For overcoming the deficit, this paper proposes to model a business process not in one notation but in several correlated diagrams, so that each diagram reveals separate perspectives, and all together they form a coordinated, integrated process description.

Key words: business process modeling, Bunge-Wand-Weber ontology, expressiveness deficit, process model perspectives.

Citation: Fiodorov I.G. (2016) Overcoming expressiveness deficit of business process modeling languages. *Business Informatics*, no. 3 (37), pp. 62–71. DOI: 10.17323/1998-0663.2016.3.62.71.

Introduction

A variety of languages and notations, namely: UML [1], BPMN [2], EPC [3], ebXML [4], BPEL [5], Petri Nets [6] are used for business process modeling. Hence, the question often arises to carry out a comparative analysis in order to determine which is better-suited for business process modeling [7]. Y. Wand and R. Weber have suggested that the ontological clarity of the modeling language can be evaluated by comparing the alphabet of this language with the constructs of the top level ontology known as Bunge-Wand-Weber (BWW) [8]. One of the key success factors of using a given language is its ability to provide the users with a symbol set (modeling

[1] This work was executed with the support of the Ministry of Education and Science of Russia within the basic part of government task No. 2014/122, reference number 2966.

primitives), which can directly reflect appropriate ontology concepts (abstracts). They identify the following correspondence options between an alphabet of the modeling language and a set of ontology concepts (*Figure 1*):

✦ construct equivalence: each symbol of an alphabet can be associated with exactly one concept;

✦ construct deficit: separate concepts have no corresponding symbol;

✦ construct excess: the ontology concept cannot be associated with any symbol;

✦ construct redundancy (synonymy): one concept can be represented directed in several symbols;

✦ construct overload (homonymy): several concepts correspond to one symbol.

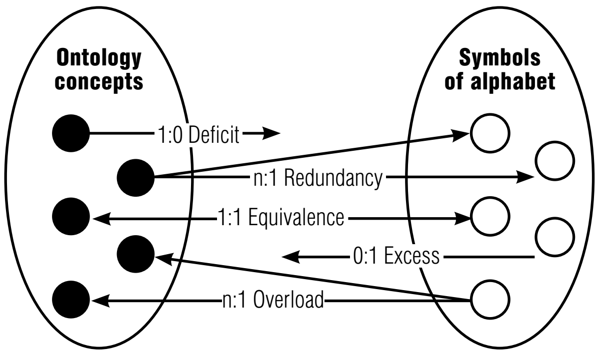

Fig. 1. Relationship between modeling language primitives and BWW ontology concepts

The essence of the approach proposed by Y. Wand and R. Weber consists in checking an equivalence of two sets, i.e. symbols of an alphabet and ontology concepts. The research demonstrates that all known business process modeling languages have an expressiveness deficit [9], so that overcoming this deficit is an important and urgent task. However, the ontology is not limited to a thesaurus, it also covers the structure of relations between concepts [10]. It may be assumed that the modeling language must be able to convey these relationships. Therefore, the approach of Y. Wand and R. Weber can be significantly enhanced if the structural relationships among BWW ontology concepts are studied. This paper also makes an attempt to extend the BWW ontology as applied to business process modeling, since in its current form it does not make it possible to represent logical operators and the temporal characteristics.

1. Enhanced BWW ontological model

The model proposed by Y. Wand and R. Weber is based on the ontology proposed by M. Bunge [11]. The world is made up of things, which are usually treated as a "separate object of the tangible world with relative independence, objectivity and stability of existence" [12], therefore, in what follows the term "object" will be used as a synonym of a thing. The object has properties which are its attributes; so a property cannot have properties. The object state is defined as a set of all values of all its attributes at a given time. Moreover, not all states are considered as acceptable and not all transitions between states are considered lawful [13]. The object state transits due to transformation, which is always implemented by a predetermined rule called the transformation law. Transformation can be interpreted as a work changing the object, or an operation being performed on the object.

Let us pay attention to the fact that M. Bunge differentiates between the intrinsic object properties inherent thereto and distinguishing one entity instance from another one (for example, the color and shape characterize each object on an individual basis) and mutual properties, which characterize one object relative to another (for example, distance is a property of a pair of objects). Speaking about the transformation, M. Bunge has in mind a change of intrinsic properties of the object. We will interpret the transformation in a more comprehensive sense, and also consider a change of mutual properties. For example, the process operation changes the intrinsic properties of the object, while the logical operator in the process diagram route the object along one of several processing paths, changing its relative position, whereas the intrinsic properties of the object remain unchanged. Therefore, by partitioning the transformations which change the intrinsic properties of the object and the transformations which modify the mutual properties, we complement the ontology with a capability to represent logical process operators [14].

The fact of changing the object state is called an event, irrespective of the cause of occurrence. Meanwhile, it remains not quite clear what is the difference between the event and the state. In current interpretation the event has a meaning "for this reason" and represents a cause-and-effect relationship: the next operation can start because of the completion of the previous one. Therefore, it emerged that the terms state and event are hard to differentiate. The event interpretation proposed by us is different from the interpretation proposed by M.Bunge. By the definition of E.A. Babkin, an event is something that is happening at some instant per saltum, step-wise and is considered as a state change of a certain object [15]. Yu.N. Pavlovsky interprets an event as an instant in time designating a change of the object states [16]. Therefore, we will link an event with a moment in time when a change of state of a certain object occurred; it has the meaning of "afterwards" – later in the chronological order. Thus, an internal event establishes the fact and the moment in time when the object passed into the following state and is ready for execution of the next operation. The occurrence of an internal event is insufficient for the beginning of execution of the next operation. In case of an interactive operation the execution begins following the interference of the actor and the latter is treated as an external object relative to the system. If the operation is automatic, then it start after a signal from the external control device. Therefore, external event represents the fact and moment in time of changing the state of the object external to the system, which initiates the execution of the operation and record the moment when the transformation began. Thereby, the terms of temporal logic are added to the ontology: a moment in time and time interval between two consecutive events [14]. The time interval between the occurrence of an internal event indicating readiness to processing, and an external event indicating the real beginning of work will be interpreted as the waiting time, the time interval between the occurrence of an external event indicating the beginning of work and internal event indicating the end of processing will be interpreted as the execution time. An external event not only initiates the execution of the process operation, but can also stop it. For example, a customer placed an order – this event initiates the process, and if the customer canceled the order, further processing may not be reasonable. The external event may imply the occurrence of an abnormal situation and require special processing. Thus, we enhanced the BWW ontology, added it with transformations which change mutual object properties they correspond to logical operators, changed the event concept interpretation such that it designates the moment in time when the object state changes, and demonstrated that the external events are related to each process operation.

An important conclusion that can be made from the analysis of BWW enhanced ontology is in specifying a set of concepts. Among these are (*Figure 2*):

✧ the object to be processed – it has an internal structure describing a set of inherent properties of the object;

✧ transformations changing intrinsic properties of the object that result in a change of its state;

✧ transformations which route the object, but do not change its state;

✧ internal events which designate a moment in time when the object is ready for execution of the next operation;

✧ external events which designate a moment in time when operation starts.

Now we have to analyze the relationships between the individual concepts of the ontology.

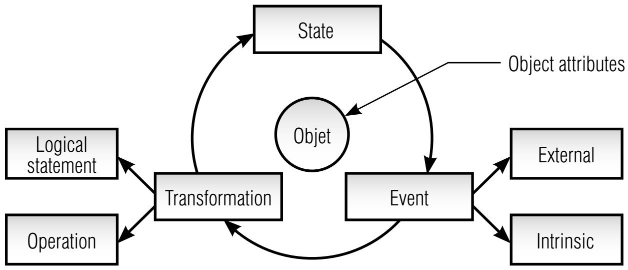

Fig. 2. Basic concepts of the process model

2. Structure of BWW ontological model

Let us consider what formal models enable us to describe relationships between separate concepts of the BWW ontology. The adapted ontological model includes six concepts, and respectively we have to consider a graph having vertexes of six types. In this graph (*Figure 3*), vertex sets of different types do not overlap; there are no arcs linking vertexes of similar type – it can be classified as sextuple. One have to note that vertex sets are linked only pair-wise: state with event, event with transformation, transformation with state. It can be seen that the above-listed relationships can be reflected by virtue of well-known modeling formalisms: the entity-relationship diagram (ER) [17], state diagram (STD) [18], data flow diagram (DFD) [19], event graph [20], and Petri nets [21]. Let us consider how the above diagrams describe relationships among pairs of concepts. We will consider only basic formalisms which have no extensions. We have to note what each diagram is capable of modeling and what it uses as a reference to another diagram.

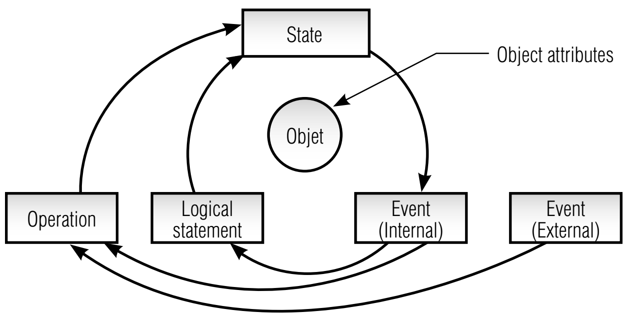

Fig. 3. Sextuple graph describing the BWW ontological model

2.1. Entity-relationship diagram (ER)

The entity-relationship diagram (ER) is used to describe information objects, its attributes and relationships between them. Retrieving the basic concepts of the domain area, one can find objects to be processed, each of which forms an appropriate process. The diagram uses the terms: entity, which is taken to mean any distinctive object, and attribute which is taken to mean a named inherent (property) of the entity [17]. The object state is determined by values which take its attributes. We can associate each named object's state with a definite set of attributes and their values. Therefore, the ER diagram models the object and its structure.

2.2. State transition diagrams (STD)

The state transition diagram is a traditional approach to describe the behavior of an object. It is customary to distinguish control and computational states [22]. For example, a control state "work is in progress/completed" reflects the status of a separate process operation. The computational state is associated with an object, it reflects success or failure of the operation. For example, operation "check the bill" can result in a success – the bill is accepted, or a failure – the bill is rejected. The subject of our discussion is the computational state of the object. Inasmuch as many variables and control flows can exist within a large application program, it is conventional to specify the state variables [23]. To simplify the analysis, the changes of one state variable are considered at any specific time, which determines the state of the entire system [22].

The state diagram shows transitions between the acceptable states of the object. It uses named object states, but does not display values of relevant attributes – this information can be obtained by reference from the ER-diagram. The state diagram does not allow us to model transformations which result in a state transition; instead, it contains a reference to the data flow diagram, where the relevant information is available.

2.3. Data flow diagram (DFD)

The data flow diagram describes the processing of information objects [24]. It is conventional to call it transformational, since it depicts the operations which transform the input data to output, but does not show those actions which do not change the object [25], so that it does not make it possible to model logical operators. Let us note that the diagram indicates a logical name of transformation, and as such the transformation algorithm is contained in the mini-specification which describes transformation of concrete attribute of the object. A DFD diagram shall be consistent with STD and ER diagrams: the initial and final states of the object shall differ in particular by those attributes which are changed by this transformation. DFD does not contain information on the moment in time when transformation can be initiated; for this purpose a reference to the event graph is used, which will be addressed below.

The question as to whether a data flow diagram is formal depends on the method of description of the mini-specification converting inputs into outputs. If the mini-specification can be defined in a strict mathematical form, the model is considered to be formal. In our case, the mini-specification can be described formally using the notion of a target value of an object and its attributes. As a result of transformation, the object shall pass into a target state, which is characterized by a certain set of target values of the attributes of this object. If transformation succeeds all target values are achieved, this means that the target state of the object is obtained, and if the target value is not achieved, then it is considered as a failure. Therefore, it is possible to abstract from specific values and describe the transformation formally using Boolean logic.

2.4. Event graph

The event graph shows the temporal relationship between the events [20]. Its nodes represent moments in time when the object changes its state. The events can be internal, associated with changes of the object under control and external ones associated with changes of other objects, which are outside of the process' control. The diagram arcs represent a sequence of events and, therefore, the event diagram depicts a temporal relationship of consequence of events. If we associate the arc length with the time interval which passes between two consecutive events, we will get a Gantt chart. For example (*Figure 4*), event E0in reflects the moment of completion of the previous operation: the object is ready for execution of the next operation, however, it does not begin immediately, but with some delay – let us call it a waiting time of the execution. External event E1ex, which is associated with the external control device initiates the execution of the next operation. The fact of completion of the next operation is reflected as internal event E1in: the object is ready for execution of the next operation; it will be again in a waiting state until external event E2ex occurs. The Gantt chart is depicted in the same figure.

2.5. Petri nets

It is commonly supposed that Petri nets enable us to model the execution behavior of the process [26]; however, this is not quite true, inasmuch as simple Petri nets have a limited expressiveness and are not able to

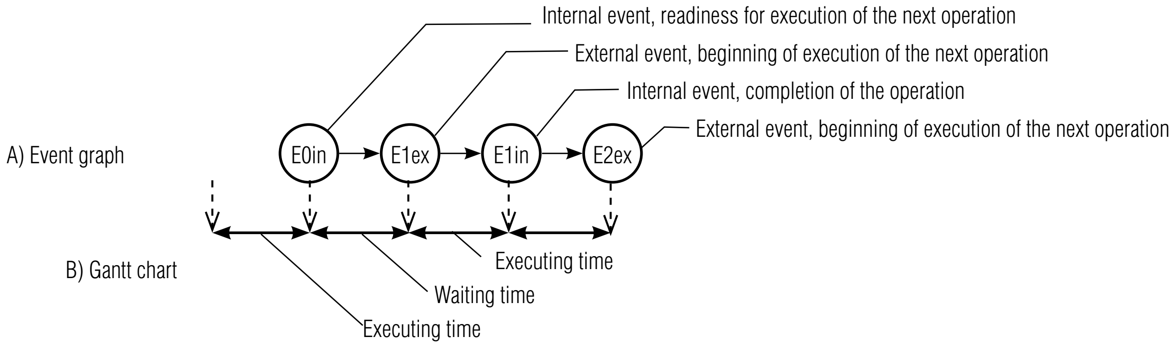

Fig. 4. Event diagram and Gantt chart

reflect the object's state. The graphical representation of the Petri net is a bipartite directed graph containing two types of nodes – places and transitions interrelated by arcs, where the nodes of the same type cannot be directly connected. Places can accommodate tokens capable of moving through arcs via transitions. In case of process modeling, a tokens is associated with a certain material object or information entity. The transition is associated with the work or operation, it moves the token from one places to another. The place is passive, it does not change and does not move the token, only keeps it between two transitions. The state or marking of Petri net at any moment in time is determined by distribution of tokens over the places. The token doesn't have state, so the change of the objet in response to operation is not analyzed. We have to note that transitions of ordinary Petri nets cannot reflect the transformation algorithm, since they do not contain a mini-specification; they can not represent transformation duration, because they occur immediately, and positions do not reflect the object state. The tokens reflects the current "spatial position" of the "control point" on the process chart as a result of routing by logical operators. Thus, ordinary Petri nets are not capable of modeling the behavior, but are suitable for modeling the process logic. This task is urgent, because a certain combinations of simple logical operators may result in collisions preventing a normal termination of the process. For example, as consequence of chaining "OR" (split) with "AND" (join) a deadlock occurs, the process stops and cannot be terminated [27].

2.6. Structure of relationships between ontology concepts

It may be concluded that relationships between the ontological concepts of the BWW model are described by five diagrams. Those familiar with engineering drawings are aware that a model of a mechanical part has three projections. Thus, in the absence of at least one of them the drawing is incomplete, and it is impossible to fabricate a part. The diagrams presented can be considered as a projections of the process model: each displays separate relations between ontological concepts, and all together they form the complete model. We have to distinguish the following perspectives and formalisms used for their description, namely, informational – the entity-relationship diagram; behavioral – the state diagram; transformational – the data flow diagram; temporal – the event graph; and logical – the ordinary Petri nets. *Table 2* presents process's perspectives and proposes formal models; symbol "M" shows a parameter modeled by an appropriate diagram, and symbol "R" is a parameter which is used as a reference to another diagram.

Table 1.

What enables us to model the diagrams

Concept / Diagram	Object and its structure	Object state	Operation		Event		Perspective
			Transformation	Routing	Internal	External	
ER	M	-	-	-	-	-	Informational
STD	R	M	R	-	-	-	Behavioral
DFD	R	R	M	-	-	-	Transformational
Petri Net	-	-	R	M	-	-	Logical
Event Graph	-	R	-	-	M	M	Temporal

3. Structural analysis of business processes modeling languages

A large body of research reveals that process modeling languages and notations are not capable of reflecting BWW ontological model concepts all at once, but only part of them. Moreover, the authors of investigations focus their attention on a percentage ratio of modeled and unmodeled concepts, calculate a relative degree of deficit, redundancy, excess and overload. *Table 2* shows the results of similar research [9]. One is compelled to ask: to what extent a language having a 10% of deficit is better than another language having a 15% expressiveness deficit?

Table 2.

Comparative analysis of modeling languages

Modeling notation	Relative degree			
	Deficit	Overload	Redundancy	Excess
BPMN 1.0	51%	35%	28%	25%
BPML 1.0	29%	65%	28%	3%
EPC	3%	62%	43%	28%
WSCI 1.0	29%	49%	18%	8%
ebXML 1.01	15%	13%	14%	5%
BPEL 1.1	32%	49%	13%	6%

Let us suggest that a requirement of equivalence of language symbols set and BWW ontology concepts is too strict, that the overload, redundancy and excess make the modeling language unsuitable for modeling. However, the expressiveness deficit of the language is acceptable, because it can be overcome. *Table 2* shows a comparison of the EPC and BPMN expressive power in order to represent various perspectives of the process model. Both notations do not model the structure of information object; thus, they do not reflect the information perspective. The symbol "event" in EPC notation reflects a state acquired by an object as a result of execution of the process operation. It makes it possible to show a sequence of state transitions and thus model objects behavior; however, no place for state mapping is foreseen in BPMN notation. Both notations represent names of the operations which transform the information object, but it is necessary to refine them using mini-specifications, to specify the properties to be changed in order to achieve a target state. The EPC diagram contains no means to indicate time intervals; therefore, it does not represent a temporal perspective — such means are available in BPMN notation. Both diagrams enable us to reflect logical process statements. In summary, it can be seen that none of the business process modeling notations are able to represent the process model perspectives all at once, but only part of them. In other words, both notations have an expressiveness deficit.

Table 3.

Comparative analysis of EPC and BPMN notations expressiveness

Notation	Model perspectives				
	Informational	Behavioral	Transforma-tional	Temporal	Logical
EPC	–	+	+	–	+
BPMN	–	–	+	+	+

In order to overcome the deficit, this paper proposes to model the business process not in one notation, but in several coordinated diagrams, so that each diagram identifies separate perspectives of the model, and all together they form an integrated description. For example, EPC notation should be supplemented with the information model and Gantt chart, and the model in BPMN notation should be supplemented with the information model and the state diagram. Diagrams depicting individual perspectives of the process model shall be well coordinated. For example, the transformational perspective should describe a change of only those properties that characterize an appropriate target state of the object.

4. Discussion

The idea that the process model consists of several perspectives was addressed by different researchers. For example, a well-known Zachman model includes six perspectives [28]. Architecture CIMOSA identifies four perspectives: functional, informational, resource, organizational [29]. The integrated model of ARIS information systems addresses four perspectives, where three — informational, organizational and functional — being considered as basic, and the choice of the fourth perspective is determined by the choice of modeling objective, i.e. for the information system modeling a resource representation is used, and for business modeling the management perspective is applied [30]. The proposal formulated by B. Curtis includes four perspectives: functional, behavioral, informational, organizational [18]. It can be seen that a number of perspectives in various researchers is different, so an objective comes up concerning justification of a list of model perspectives.

Let us agree to distinguish the terms “process” and “business process”. The difference is as follows. According to M. Bunge, the process is a history of a certain object which change its states due to the execution of transformations initiated by events. Let us consider what sense we put into the term “process” when we add the word “business”. Firstly, we mean that the controlled object is an informational one, otherwise, if the object is tangible, one should speak of a manufacturing process. Secondly, we assume that there is some technology interpreted as a method of obtaining a reproducible result of a required quality for a specified time with reasonable utilization of economic resources. The purpose of a business process is a reproducible output which can be achieved by formalizing the way of performing the operations. Thirdly, the business process require some economic resources, and if their consumption is higher than planned, we should speak about the procedural violation. The enhanced BWW ontology addressed by us in this paper includes five perspectives: informational, behavioral, transformational, temporal and logical, with each perspective having its own formalism. It does not contain concepts characterizing the economic result and taking into account the economic resources spent, so it describes the process model but not business process.

Let us turn to the above assumption that the expressiveness deficit of the process modeling language can be overcome. If some business processes modeling notation does not allow us to reflect individual perspectives, we can talk about an expressiveness deficit of a relevant language. The deficit can be overcome by modeling a process in several correlated diagrams so that each diagram depicts separate aspects, but all together they give a complete and integrated representation of all its aspects. Certain perspectives should be consistent, so that references show links between diagrams.

The result obtained is of great practical importance. The software modeling environment like ARIS and UML include plenty of notations, and the analyst is invited to make a choice, taking into account his personal preferences. It is left aside that having selected the basic modeling notation the analyst should complement it with such diagrams which all together cover all perspectives of the process model.

The approach proposed in the paper is generally consistent with the suggestions made by E. Jordan, who within the structural modeling method proposed sequential modeling in three diagrams DFD, STD and ER [31]. E. Jordan did not set a goal to design a real-time system, so his structural method omits the temporal aspect and Gantt chart; the business logic is not modeled, so no Petri nets are missing. A selection of perspectives by E Jordan is not theoretically justified. Since we consider the most general case, we added our model so as to take into account all the relationships between the concepts. Similar comments are true if we consider the executable xUML, since it uses the same set of diagrams as E. Jordan [32].

Conclusion

The novelty of the analysis performed in this paper is in the adaptation of Bunge-Wand-Weber ontological model for process modeling. Additional concepts are identified and a new interpretation is given to them. The relationships between concepts are studied, five perspectives of the process model: informational, behavioral (state), transformational, logical and temporal are theoretically justified. For each perspective, a formalism is defined. A difference between the process model and business process model is demonstrated.

A practically important result is obtained, proving that none of the known business process modeling languages is capable to represent all BWW ontological concepts at once, but only part of them. Thus, all known modeling notations have an expressiveness deficit. This paper proposes a method for overcoming the deficit consisting in the use of an integrated process model which includes a number of perspectives, each showing some aspects of the process model, and all together they form a complete, coordinated representation.

The result explain why the executable business process model requires rather much programming. Firstly, the executable model in BPMN notation is not capable of representing separate process model perspectives. Secondly, it can happen that perspectives are insufficiently integrated with each other at a model level. Both shortcoming has to be compensated with an additional software code. A method is proposed to overcome the ontological expressiveness deficit which consists in process modeling in several diagrams, so that each of them “covers” separate perspectives of the process model, and all together they enable us to create a complete and comprehensive integrated description of the process. That will eliminate a need in additional programming. ■

References

1. Opdahl A., Henderson-Sellers B. (2002) Ontological evaluation of the UML using BWW model. *Software and Systems Modeling*, vol. 1, no. 1, pp. 43–67.
2. Recker J., Rosemann M., Krogstie J. (2007) Ontology versus pattern-based evaluation of process modeling languages: A comparison. *Communications of the Association for Information Systems*, vol. 48, no. 20, pp. 774–799.
3. Green P., Rosemann M. (2000) Integrated process modeling. An ontological evaluation. *Information Systems*, vol. 25, no. 2, pp. 73–87.
4. Green P., Rosemann M., Indulska M. (2005) Ontological evaluation of enterprise eystems interoperability using ebXML. *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 5, pp. 713–725.
5. Green P., Rosemann M., Indulska M., Manning C. (2007) Candidate interoperability standards: An ontological overlap analysis. *Data & Knowledge Engineering*, vol. 62, no. 2, p. 274–291.
6. Rosemann M., Green P., Indulska M., Recker J. (2009) Using ontology for the representational analysis of process modelling techniques. *International Journal of Business Process Integration and Management*, vol. 4, no. 4, pp. 251–265.
7. Fiodorov I.G. (2011) Sravnitel'nyy analiz metodov modelirovaniya biznes-protsessov [Comparative analysis of business processes modeling methods]. *Open Systems*, no. 8, pp. 28–30 (in Russian).
8. Wand Y., Weber R. (2002) Research commentary: Information systems and conceptual modeling – A research agenda. *Information Systems Research*, vol. 13, no. 4, pp. 363–376.
9. Recker J., Rosemann M., Indulska M., Green P. (2005) *Business process modeling: A maturing discipline?* // BPM Center Report BPM-06-20, 2005 37. Available at: http://bpmcenter.org/wp-content/uploads/reports/2006/BPM-06-20.pdf (accessed 15 January 2014).
10. Nayhanova L.V. (2005) Osnovnye aspekty postroeniya ontologiy verkhnego urovnya i predmetnoy oblasti [Main aspects of construction of high level ontologies and subject area]. *Internet Portals: Content and Technologies*. Moscow: Informika, Prosveshchenie, pp. 452–479 (in Russian).
11. Bunge M. (1977) *Treatise on basic philosophy ontology I: The furniture of the world*. Vol 3. Boston, MA: D. Reidel Publishing Company.
12. Uemov A.I. (1963) *Veshchi, svoystva i otnosheniya* [Things, properties and relations]. Moscow: USSR Academy of Sciences (in Russian).
13. Soffer P., Wand Y. (2005) On the notion of soft-goals in business process modeling. *Business Process Management Journal*, vol. 11, no. 6, pp. 663–679.
14. Fiodorov I.G. (2015) Adaptatsiya ontologii Bunge-Vanda-Vebera k opisaniyu ispolnyaemykh modeley biznes-protsessov [Adaptation of Bunge-Wand-Weber ontology to description of executable business processes models]. *Applied Informatics*, no. 4 (58), pp. 82–92 (in Russian).
15. Babkin E.A. (2010) O ponyatii sobytiya v diskretno-sobytiynom modelirovanii [On the concept of event in discrete-event modeling]. *Information Systems. Theory and Practice*. Kursk: Kursk State University, pp. C. 46–51 (in Russian).
16. Pavlovskiy Yu.N., Belotelov N.V., Brodskiy Yu.I. (2008) *Imitatsionnoe modelirovanie* [Simulation modeling]. Moscow: Academy (in Russian).
17. Kuznetsov S.D. (2005) *Osnovy baz dannykh* [Fundamentals of databases]. Moscow: INTUIT (in Russian).
18. Curtis B., Kellner M., Over J. (1992) Process modeling. *Communications of the ACM*, vol. 35, no. 9, pp. 75–90.
19. Bruza P.D., van der Weide T.P. (1993) The semantics of Data Flow Diagrams. Proceedings of the *International Conference on Management of Data*, Prentice Hall, pp. 1–13.
20. Schruben L. Simulation modeling with event graphs (1983) *Communications of the ACM*, vol. 26, no. 11, pp. 957–963.
21. Storrle H. (2000) *Models of Software Architecture. Design and analysis with UML and Petri-nets*. Munchen: Ludwig-Maximilians-Universitat.
22. Shalyto A.A. (1998) *SWITCH-tekhnologiya. Algoritmizatsiya i programmirovanie zadach logicheskogo upravleniya* [SWITCH-technology. Algorithmization and programming of logical control tasks]. Saint Petersburg, Nauka (in Russian).
23. Booch G., Maksimchuk R. (2008) *Ob"ektno-orientirovannyy analiz i proektirovanie s primerami prilozheniy* [Object-oriented analysis and design with applications]. Moscow: Williams (in Russian).
24. Kalashian A.N., Kalianov G.N. (2003) *Strukturnye modeli biznesa: DFD-tekhnologii* [Structured models of business: DFD technologies]. Moscow: Finance and Statistics (in Russian).
25. Whitten L., Bentley K., Dittman J. (2004) *Systems analysis and design methods*. McGraw-Hill Companies.
26. Ter Hofstede A., van der Aalst W., Adams M., Russell N. (2010) *Modern business process automation*. Springer Verlag.
27. Van der Aalst W., van Hee K., ter Hofstede A., Sidorova N., Verbeek H., Voorhoeve M., Wynn M. (2011) Soundness of workflow nets: Classification, decidability, and analysis. *Formal Aspects of Computing*, vol. 23, no. 3, pp. 333–363.
28. Zachman J.A. (2003) The Zachman Framework: A primer for enterprise engineering and manufacturing. Available at: http://www.zachmaninternational.com (accessed 15 January 2015).
29. Vernadat F. (1996) Enterprise integration: On business process and enterprise activity modeling. *Concurrent Engineering: Research and Applications*, vol. 4, no. 3, pp. 219–228.
30. Scheer A.W., Nüttgens M. (2002) ARIS Architecture and reference models for business process management. *Business Process Management. Springer Lecture Notes in Computer Science*, vol. 1806, pp. 376–389.
31. Yourdon E. (1988) *Modern structured analysis*. Prentice Hall.
32. Mellor S., Balcer M. (2002) *Executable UML. A foundation for model-driven architecture*. Boston, MA: Addison-Wesley.
33. Wand Y., Weber R. (1999) An ontological model of an information system. *IEEE Transactions on Software Engineering*, vol. 16, no. 11, pp. 1282–1292.

Преодоление дефицита выразительной способности языков моделирования бизнес-процессов[1]

И.Г. Фёдоров

кандидат технических наук
доцент кафедры прикладных информационных систем и информационной безопасности
Российский экономический университет им. Г.В. Плеханова
Адрес: 119501, г. Москва, ул. Нежинская, д. 7
E-mail: IFedorov@mesi.ru

Аннотация

Я. Ванд и Р. Вебер предположили, что «онтологическое» качество языка моделирования, можно оценить путем сравнения алфавита этого языка с конструкциями предлагаемой ими онтологии верхнего уровня, получившей название Бунге-Ванда-Вебера (BWW). Одним из главных факторов успеха использования языка они называют его способность предоставить пользователям набор знаков (примитивов моделирования), которые могут непосредственно отображать соответствующие концепты (абстракции) онтологии. Однако онтология не сводится к тезаурусу, она также включает спецификацию структуры соответствующей проблемной области. Можно предположить, что язык моделирования должен быть способен передать эти связи. Таким образом, подход Я. Ванда и Р. Вебера можно существенно развить, если исследовать структурные связи между концептами онтологии BWW. В работе предпринята попытка расширить онтологию BWW применительно к моделированию бизнес-процесов. Добавлены трансформации, которые изменяют взаимные свойства, им соответствуют логические операторы процесса, изменена трактовка концепта событие, таким образом, что оно фиксирует момент времени, когда происходит изменение состояния внешнего объекта. Показано, что внешние события связаны с каждой операцией процесса. Тем самым добавлены понятия темпоральной логики: момент времени и интервал времени между двумя последовательными событиями. Исследование связей между концептами расширенной онтологии BWW позволило обосновать пять перспектив модели процесса и выделить формализмы, используемые для их описания: информационную – диаграмма «сущность – связь»; поведенческую – диаграмма состояний; трансформационную – диаграмма потоков данных; темпоральную – граф состояний; логическую – обычные сети Петри. Многочисленные исследования показывают, что языки и нотации моделирования процессов не способны отобразить сразу все концепты онтологической модели BWW, но только их часть. При этом авторы исследований концентрируют внимание на процентном соотношении моделируемых и не моделируемых концептов, подсчитывают относительную степень дефицита, избыточности, неоднозначности и неразличимости. Для преодоления дефицита в данной работе предлагается моделировать бизнес-процесс не в одной нотации, а в нескольких согласованных диаграммах, так, чтобы каждая раскрывала отдельные перспективы модели, а все вместе они образовывали согласованное интегрированные описание.

Ключевые слова: моделирование бизнес-процессов, онтологии Бунге-Ванда-Вебера, дефицит выразительной способности, перспективы модели процесса.

Цитирование: Fiodorov I.G. Overcoming expressiveness deficit of business process modeling languages // Business Informatics. No. 3 (37). P. 62–71. DOI: 10.17323/1998-0663.2016.3.62.71.

Литература

1. Opdahl A., Henderson-Sellers B. Ontological evaluation of the UML using BWW model // Software and Systems Modeling. 2002. Vol. 1. No. 1. P. 43–67.
2. Recker J., Rosemann M., Krogstie J. Ontology versus pattern-based evaluation of process modeling languages: A comparison // Communications of the Association for Information Systems. 2007. Vol. 48. No. 20. P. 774–799.
3. Green P., Rosemann M. Integrated process modeling. An ontological evaluation // Information Systems. 2000. Vol. 25. No. 2. P. 73–87.
4. Green P., Rosemann M., Indulska M. Ontological evaluation of enterprise eystems interoperability using ebXML // IEEE Transactions on Knowledge and Data Engineering. 2005. Vol. 17. No. 5. P. 713–725.

[1] Работа выполнена при поддержке Министерства образования и науки России, в рамках базовой части государственного задания № 2014/122 шифр 2966.

5. Green P., Rosemann M., Indulska M., Manning C. Candidate interoperability standards: An ontological overlap analysis // Data & Knowledge Engineering. 2007. Vol. 62. No. 2. P. 274–291.
6. Rosemann M., Green P., Indulska M., Recker J. Using ontology for the representational analysis of process modelling techniques // International Journal of Business Process Integration and Management. 2009. Vol. 4. No. 4. 2009. P. 251–265.
7. Фёдоров И.Г. Сравнительный анализ методов моделирования бизнес-процессов // Открытые системы. 2011. № 8. С. 28–30.
8. Wand Y., Weber R. Research commentary: Information systems and conceptual modeling – A research agenda // Information Systems Research. 2002. Vol. 13. No. 4. P. 363–376.
9. Recker J., Rosemann M., Indulska M., Green P. Business process modeling: A maturing discipline? // BPM Center Report BPM-06-20, 2005 37 [Электронный ресурс]: http://bpmcenter.org/wp-content/uploads/reports/2006/BPM-06-20.pdf (дата обращения: 15.01.2014).
10. Найханова Л.В. Основные аспекты построения онтологий верхнего уровня и предметной области // В сб.: Интернет-порталы: содержание и технологии. М.: ФГУ ГНИИ ИТТ «Информика». Просвещение, 2005. С. 452–479.
11. Bunge M. Treatise on basic philosophy ontology I: The furniture of the world. Vol 3. Boston, MA: D. Reidel Publishing Company, 1977. 369 p.
12. Уемов А.И. Вещи, свойства и отношения. М.: Академия наук СССР, 1963. 184 с.
13. Soffer P., Wand Y. On the notion of soft-goals in business process modeling // Business Process Management Journal. 2005. Vol. 11. No. 6. P. 663–679.
14. Фёдоров И.Г. Адаптация онтологии Бунге-Ванда-Вебера к описанию исполняемых моделей бизнес-процессов // Прикладная информатика. 2015. № 4 (58). С. 82–92.
15. Бабкин Е.А. О понятии события в дискретно-событийном моделировании // В сб.: Информационные системы: Теория и практика. Курск: Курский гос. университет, 2010. С. 46–51.
16. Павловский Ю.Н., Белотелов Н.В., Бродский Ю.И. Имитационное моделирование. М.: Академия, 2008. 236 с.
17. Кузнецов С.Д. Основы баз данных. М.: ИНТУИТ, 2005. 488 с.
18. Curtis B., Kellner M., Over J. Process modeling // Communications of the ACM. 1992. Vol. 35. No. 9. P. 75–90.
19. Bruza P.D., van der Weide T.P. The semantics of Data Flow Diagrams // Proceedings of the International Conference on Management of Data. Prentice Hall, 1993. P. 1–13.
20. Schruben L. Simulation modeling with event graphs // Communications of the ACM. 1983. Vol. 26. No. 11. P. 957–963.
21. Storrle H. Models of Software Architecture. Design and analysis with UML and Petri-nets. Munchen: Ludwig-Maximilians-Universitat, 2000. 306 p.
22. Шалыто А.А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб: Наука, 1998. 628 с.
23. Буч Г., Максимчук Р. Объектно-ориентированный анализ и проектирование с примерами приложений. М.: Вильямс, 2008. 720 с.
24. Калашян А.Н., Калянов Г.Н. Структурные модели бизнеса: DFD-технологии. М.: Финансы и статистика, 2003. 234 с.
25. Whitten L., Bentley K., Dittman J. Systems analysis and design methods. McGraw-Hill Companies, 2004. 550 p.
26. Ter Hofstede A., van der Aalst W., Adams M., Russell N. Modern business process automation. Springer Verlag, 2010. 616 p.
27. Soundness of workflow nets: Classification, decidability, and analysis / W. van der Aalst [et al] // Formal Aspects of Computing. 2011. Vol. 23. No. 3. P. 333–363.
28. Zachman J.A. The Zachman Framework: A primer for enterprise engineering and manufacturing (electronic book). 2003. [Электронный ресурс]: http://www.zachmaninternational.com (дата обращения: 15.01.2015).
29. Vernadat F. Enterprise integration: On business process and enterprise activity modeling // Concurrent Engineering: Research and Applications. 1996. Vol. 4. No. 3. pp. 219–228.
30. Scheer A.W., Nüttgens M. ARIS Architecture and reference models for business process management // Business Process Management. Springer Lecture Notes in Computer Science. 2002. Vol. 1806. P. 376–389.
31. Yourdon E. Modern structured analysis. Prentice Hall, 1988. 688 p.
32. Mellor S., Balcer M. Executable UML. A foundation for model-driven architecture. Boston, MA: Addison-Wesley, 2002. 368 p.
33. Wand Y., Weber R. An Ontological model of an information system // IEEE Transactions on Software Engineering. 1999. Vol. 16. No. 11. P. 1282–1292.