Controlling Business Object States in Business Process Models to Support Compliance

Ludmila Penicina

Institute of Applied Computer Systems, Riga Technical University, 1 Kalku, Riga, LV-1658, Latvia
{ludmila.penicina}@rtu.lv

Abstract. The doctoral thesis addresses the existing gap between business process models and states of business objects. Existing modelling methods such as BPMN and ArchiMate lack an explicitly declarative approach for capturing states of business objects and laws of state transitions. This gap hinders the compliance of business process models with regulations imposed internally or externally, and can result in potential legal problems for organizations. Also this gap makes the ability of BPMN and ArchiMate to capture real-world phenomena questionable and drives modellers to employ additional standards and models (e.g., state-machine diagrams). The research goal of the doctoral thesis is to propose a formalized solution for closing the gap between business process models and states of business objects by using Bunge-Wand-Weber model. The solution approach includes means for explicit definition of states of business objects, automatic generation of conceivable state space at a process model design-time, automatic generation of lawful state space, and compliance checking at a process run-time.

Keywords: Business process modelling, BWW model, BPMN, states, compliance.

1 Introduction

Business processes are valuable assets of any organization. In organizations business process modelling has become a main activity for capturing, analysing, and improving business processes. Business process modelling comprises two aspects – the control-flow perspective and data-flow perspective [1]. Control-flow perspective defines possible execution paths of a business process, while data-flow perspective represents how business objects are manipulated and change states during a process. Data in business process models are usually declared in terms of business objects (physical or virtual), and usually there are prescribed allowed or legal (further in text – lawful) states of business objects described in internal business policies, external legislative documents, standards, reference models, and other regulations.

Nowadays there is an increased pressure on organizations to guarantee compliance of their business processes with various legislative and regulatory requirements, other externally imposed constraints, and internal business policies [2]. For an organization engaged in business process modelling this might mean to employ controlling states of business objects in business process models. This requires the following prerequisites for business process modelling:

1. It has to be possible to associate activities in business process models with business objects representing inputs or outputs.
2. It has to be possible to define possible state spaces (lawful and unlawful states) and state transitions of business objects in business process models.
3. It has to be possible to represent a state of a business object at a given point of time.
4. It has to be possible to detect if a state of a business object is lawful, and to encourage action if it is not.

By compliance the author does not mean the soundness of the process – correctness criteria that a process model has to fulfil, e.g., deadlock or livelock patterns. Compliance can be checked during or after the execution of the business process, called compliance by detection; or compliance can be checked while modelling the business process, called compliance by design [3].

Nowadays organizations employ industry modelling standards like BPMN [4] and ArchiMate [5] to understand and improve business processes. Business Process Model and Notation (BPMN) is the de-facto standard for representing in a very expressive graphical way the processes occurring in virtually every kind of organizations [6]. However, BPMN has its limitations when it comes to modelling other aspects of organizations such as organizational structure and roles, functional breakdowns, data, strategy, business rules, and technical systems [7].

Information about Enterprise Architecture (EA) is needed to create real-world business process models. To provide a uniform representation for diagrams that describe EA, ArchiMate modelling language has been developed [8]. The core of ArchiMate language consists of three main types of elements: active structure elements, behaviour elements, and passive structure elements (objects). Some tools like ARIS [9] and QPR [10] allow linking BPMN and ArchiMate models in their modelling environments. Linkage between BPMN models and ArchiMate models provides possibilities to complement BPMN models with enterprise architecture aspects, and ArchiMate models with detailed process descriptions. However, existing BPMN-ArchiMate linkage solutions do not address the gap between state spaces of business objects and linked BPMN-ArchiMate models. In this paper the author particularly addresses linked BPMN and ArchiMate models, which, for simplicity reasons, are called business process models.

1.1 Research Methodology

The proposed methodology for the doctoral thesis is a design science method using deductive research approach. The validation of the proposed solution will be conducted using Delphi estimation method combining expert judgement. Delphi estimation will include individual estimates, sharing the estimates with experts, and having several rounds until consensus is reached. The steps of the proposed methodology are as followed:

1. Identification of existing gaps and problem statement.

2. Literature analysis.
3. Definition of research goal.
4. Determining solution approach.
5. Defining solution scope.
6. Requirements analysis.
7. Design of solution prototype.
8. Assessment of proposed solution.
9. Validation of proposed solution.

1.2 Motivation and Problem Statement

The author motivates the research with the following: controlling states of business objects in business process models: (1) can assist organization in compliance to ensure that organization will not violate laws and there will be no potential legal problems for the organization, and (2) can contribute to consistency in collaborative business processes and customer satisfaction. A number of studies exist that show the importance of addressing data and states of data in business process models, e.g., in [11] authors indicate the importance of data-driven process structures in large engineering processes such as assembling of a car or an airplane, and according to [12] in order to achieve safe execution of a process model it must be ensured that every time a task attempts to access a data object, the data object is in a certain expected data state (legal state). And, since not all possible transitions of states are meaningful, restrictions on object state transitions are also required. In this paper the author intentionally uses the term “business objects” and not “data objects”, since active structure elements (such as actors or application components) are also capable of assuming a state which can be illegal and should be also monitored.

1.3 Identified Gap and Research Goal

The previous research has shown that BPMN and ArchiMate lack in ability to describe flow of business objects in business process models, and explicitly declare states of business objects and state transition laws imposed by regulations (see [13], [14], and [15]). This gap hinders compliance of business process models with external and internal regulations. This paper continues the research presented in [14] and [15] where the evaluation of BPMN and ArchiMate against Bunge-Wand-Weber model (further in text – BWW model) was presented. Based on the results presented in previous works, we can conclude that BWW model defines a set of elements that are supported by BPMN and ArchiMate modelling language as well as a set of elements that are not supported by these modelling languages.

Majority of BPMN and ArchiMate core elements can be mapped to BWW constructs. However, it is necessary to supplement BPMN and ArchiMate modelling languages with the missing elements in order to be able to maintain a set of object states in business process models. As in BPMN and ArchiMate there is no explicit representation for object’s *State*, *Conceivable State Space*, *Lawful State Space*, *State*

Law, *Conceivable Event Space*, *Lawful Event Space*, and *History* – the resulting BPMN and ArchiMate models may be irrelevant and modellers may need to incorporate additional modelling techniques to overcome these defects. It may be impossible to detect from BPMN and ArchiMate models which states should be expected to occur and which states can occur but are illegal (unlawful). Another important aspect is lacking of element *History* which chronologically describes state changes of business objects. This deficiency can lead to problems regarding maintaining system's log and recovery.

These gaps hinder lawfulness of business process models, because lawful and unlawful states of business objects are not explicitly defined in business process models, models might contain meaningless states, since a set of conceivable states are not depicted, and, as a result, business process models do not represent real-world processes and can lead to business process incompliance with regulations. Also, since BPMN proclaims to be directly executable, omitting states and state transition laws may hinder correct automated execution.

The doctoral thesis aims to provide a solution and a prototype of a tool for supporting explicit declaration of states of business objects at design-time, and compliance checking of business objects states at run-time.

2 Related Works

The lack of consistent theoretical foundation for building information systems urged Wand and Weber to build a set of models for the evaluation of modelling techniques [16]. Wand and Weber have extended the ontology presented by Mario Bunge [17] and developed a formal foundation called BWW model for modelling information systems. Elements in BWW model (in the text shown in italics) can be organized in the following groups (adapted from [18]):

1. *Thing* – including *Properties, Classes* and *Kinds* of *Things*. *Thing* is an elementary unit in BWW. *Things* possess *Properties*, which defines *States* of a *Thing*. *Things* can belong to *Classes* or *Kinds* depending on a number of common *Properties*. A *Thing* can act on another *Thing* if its existence affects the *History* of the other *Thing*. *Things* are coupled if one *Things* acts on another.
2. *State of Thing* – *Properties* of *Things* define their *States*. *State Law* restricts *Values* of *Properties* of *Things*. *Conceivable State Space* is a set of all *States* a *Thing* can assume. *Lawful State Space* defines *States* that comply with *State Law*. *Stable State* is a *State* in which a *Thing* or a *System* will remain unless forced to change by a *Thing* in the *System Environment*. *Unstable State* is a *State* that will be changed into another *State* by the *Transformations* in the *System*. *History* is the chronologically-ordered *States* of a *Thing*.
3. *Transformation* – transformation between *States of Things*. *Transformation* is a mapping from one *State* to another. *Lawful Transformation* defines which *Events* in a *Thing* are lawful.
4. *Event* – event is a change in *State* of a *Thing*. *Conceivable Event Space* is a set of all *Events* that can occur to a *Thing*. *Lawful Event Space* is a set of all *Events* that are lawful to a *Thing*. *Events* can be *Internal Events* and *External Events*. *Events*

can be *Well-Defined* - an *Event* in which the subsequent *State* can be predicted - or *Poorly-Defined* – an *Event* in which the subsequent *State* cannot be predicted.

5. *System* – a set of coupled *Things*. *System Composition* are *Things* in the *System*. *System Environment* is *Things* outside the *System* interacting with the *System*. *System Structure* is a set of couplings that exists among *Things*. *Subsystem* is a *System* whose composition and structure is a subset of the composition and structure of another *System*. *System Decomposition* is a set of *Subsystems*. *Level Structure* is an alignment of the subsystems.

The authors of [5] propose a notion of "weak conformance" which checks conformance of a process model with respect to data objects. This notion can be used to tell whether in every execution of a process model each time a task needs to access a data object in a particular state, it is ensured that the data object is in the expected state or can reach the expected state and, hence, the process model can achieve its goals.

In [19] authors identify that consistency between business process models and object life cycle is required, however, their relation is not well understood. Authors clarify this relation and propose an approach to establish the required consistency by explicitly defining object states in business process models and then generating life cycles for each object type in the process. The authors of [19] indicate that object life cycle modelling is valuable at the business level. However, we propose to consider states of objects also at the application and technology levels of enterprise architecture since objects can be hidden and specified in sub-process structures at different levels of an enterprise.

The authors of [20] use object life cycle as a common means for explicitly modelling allowed state transitions of an object during its existence and propose a technique for generating a compliant business process model from a set of given reference object life cycles.

The notion of a "legal state" is also mentioned in [21] where authors indicate that the representation of legal states in a model of a trade procedure is essential because organizations should be able to derive their obligations, rights, and duties at each point during the execution of the trade procedure and propose to annotate the states in Petri nets.

In [2] authors investigate the use of temporal deontic assignments on activities as a means to declaratively capture the control-flow semantics that reside in business regulations and business policies. In object-oriented paradigm, state machines are extensively used for representation of states of objects [21].

In [22] the authors propose logic based formalism for describing the semantics of business contracts and the semantics of compliance checking procedures and close the gap between business processes and business contracts.

In [3] the author focuses on compliance by design and extends artifact-centric approach to model compliance rules using Petri nets and show how compliant business processes can be synthesized automatically from the point of view of the involved business objects.

The doctoral thesis differs from the related work in that it uses BWW model as a theoretical foundation for generating conceivable and lawful state spaces from a business process model and applies it to nowadays de-facto modelling methods BPMN and ArchiMate.

3 Proposed Solution Approach

Wand and Weber [16] built a set of models for the evaluation of modelling techniques based on an upper ontology defined by Bunge [17]. They extended Bunge's ontology and applied it to the modelling of information systems (BWW model). BWW model consists of constructs present in the real world that must be represented in information systems. BWW model allows straightforwardly addressing (further in text BWW elements are in italics): (1) *states* of *things*, (2*) lawful state space* and *lawful event space* of *things*, (3) *conceivable state space* and *conceivable event space* of *things*, (4) *state law* that restricts values of the properties of *things* to a lawful subset, and (5) *lawful transformations* that define which events in *things* are lawful.

To be able to control whether a business object has assumed an unlawful state it is necessary: (1) to provide means explicitly defining states of business objects in business process models, (2) to generate lawful and conceivable state spaces for business process models at business process model design-time, and (3) to control compliance of business process with lawful state spaces at run-time.

For a theoretical foundation purpose we propose to use BWW model [16], since BWW model complements BPMN and ArchiMate for what they are lacking – explicit representation of business objects, their states, and state transition laws. Using BWW model will potentially support creating business process models compliant with regulations, since missing BWW elements are addressed. The solution approach includes the following:

1. Explicitly defining *Properties* of business objects in business process models using formal definitions and indicating whether business object is an input or output parameter of an activity.
2. Explicitly defining *States* of business objects in business process models using formal definitions.
3. At business process design-time generating automatically *State Law*, *Conceivable State Space* and *Conceivable Event Space* directed graphs based on formal definitions and explicitly defined *Properties* and *States* of business objects.
4. Controlling compliance of business process with lawful states of business objects at run-time. At business process run-time based on a particular process scenario or case, generating automatically *Lawful State Space*, *History*, and *Lawful Event Space* directed graphs.
5. Using rules for object life cycle generation presented in [19] for automatically generating conceivable and lawful state spaces. Rules for object life cycle generation presented in [19] are based on patterns that are matched in the business process model and used to create object life cycle with state transitions from initial state to possible end states.

The proposed solution for controlling lawful states of business objects in business process models requires a repository-based modelling tool that accommodates BPMN, ArchiMate, and BWW. The solution approach includes means for explicit definition of states of business objects, automatic generation of conceivable state space at a process model design-time, and automatic generation of lawful state space and compliance checking at a process run-time.

In the doctoral thesis the author intends to apply compliance by detection method to check during the execution of the business process if states of business objects are compliant with the lawful state space. However, it is also intended to generate a space of conceivable states for business objects at design time of business process model.

4 Conclusions and Contributions

The previous research has shown that BPMN and ArchiMate lack in ability to describe flow of business objects in business process models and explicitly declare states of business objects imposed by regulations (see [13], [14] and [15]). This gap hinders compliance of business process models with external and internal regulations. Research presented in [18] describes how BPMN and ArchiMate support BWW model. There are 6 BWW model elements that are not supported by these modelling languages, namely, *State Law* (SL), *Conceivable State Space* (CSS), *Lawful State Space* (LSS), *History* (H), *Conceivable Event Space* (CES), and *Lawful Event Space* (LES): or a tuple {SL, CSS, LSS, H, CES, LES}. The doctoral thesis focuses on the use of BWW elements {SL, CSS, LSS, H, CES, LES} in designing solution to bridge the gap between business process models and object states to support compliance.

Definition of object states in business process models are especially required in data-driven processes – in any process model that is based on data and manipulates with business objects. This paper presents an ongoing doctoral thesis research towards supporting controlling states of business objects in business process models. BWW model is used as the foundation, since it allows straightforwardly addressing the lawful and conceivable state spaces of business objects.

The main contribution of the doctoral thesis will be a formalized solution prototype for controlling states of business objects in business process models that has a capacity to support organizations in ensuring compliance between business process models and regulations.

Acknowledgments

The supervisor of the doctoral thesis is Professor, Dr.sc.ing Marite Kirikova. The author is grateful to Professor Marite Kirikova for valuable and constructive input. This research is funded in part by the Latvian Council of Science grant for project No 342/2012, and in part by the Latvian National research program SOPHIS under grant agreement Nr.10-4/VPP-4/11.

References

1. Weske, M.: Business Process Management. Springer Berlin Heidelberg, Berlin, Heidelberg (2012).
2. Goedertier, S., Vanthienen, J.: Designing compliant business processes with

obligations and permissions. Bus. Process Manag. Work. 5–14 (2006).
3. Lohmann, N.: Compliance by design for artifact-centric business processes. Inf. Syst. 7241, (2013).
4. OMG: Business Process Model and Notation 2.0, www.bpmn.org.
5. The Open Group: ArchiMate 3.0 Specification, https://www2.opengroup.org/ogsys/catalog/C162
6. Chinosi, M., Trombetta, A.: BPMN: An introduction to the standard. Comput. Stand. Interfaces. 34, 124–134 (2012).
7. Silver, B.: BPMN Method and Style with Implementer's Guide. Cody-Cassidy Press (2011).
8. Group, T.O., Forum, A.: An Introduction to ArchiMate, an Open Group Standard. (2012).
9. Softwareag: ARIS ArchiMate Modeler, http://goo.gl/WI76E.
10. Enterprise Architecture modeling software QPR EnterpriseArchitect, http://goo.gl/8Ots6h.
11. Dominic, M., Müller, D., Reichert, M., Herbst, J.: A new paradigm for the enactment and dynamic adaptation of data-driven process structures. Adv. Inf. Syst. Eng. 48–63 (2008).
12. Meyer, A., Polyvyanyy, A., Weske, M.: Weak Conformance of Process Models with respect to Data Objects. ZEUS. 847, 74–80 (2012).
13. Rosemann, M., Recker, J., Indulska, M., Green, P.: A Study of the Evolution of the Representational Capabilities of Process Modeling Grammars. Adv. Inf. Syst. Eng. 447–461 (2006).
14. Penicina, L., Kirikova, M.: Towards Completeness and Lawfulness of Business Process Models. Perspect. Bus. Informatics Res. Lect. Notes Bus. Inf. Process. Vol.158, 63–77 (2013).
15. Penicina, L.: Linking BPMN, ArchiMate, and BWW: Perfect Match for Complete and Lawful Business Process Models? Short Pap. Proc. 6th IFIP WG 8.1 Work. Conf. Pract. Enterp. Model. (PoEM 2013). Vol-1023, 156–165 (2013).
16. Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. Inf. Syst. J. 3, 217–237 (1993).
17. Bunge, M.: Treatise on Basic Philosophy: Vol. 4: Ontology II: A World of Systems. (1979).
18. Recker, J., Indulska, M., Rosemann, M., Green, P.: Do Process Modelling Techniques Get Better ? A Comparative Ontological Analysis of BPMN. Campbell, Bruce, Underwood, Jim, Bunker, Deborah 16th Australas. Conf. Inf. Syst. (2005).
19. Ryndina, K., Küster, J., Gall, H.: Consistency of business process models and object life cycles. Model. Softw. Eng. 80–90 (2007).
20. Küster, J., Ryndina, K., Gall, H.: Generation of business process models for object life cycle compliance. Bus. Process Manag. 165–181 (2007).
21. Bons, R.W.H., Lee, R.M., Wagenaar, R.W., Wrigley, C.D.: Modelling inter-organizational trade using Documentary Petri Nets. Proceedings of the Twenty-Eighth Hawaii International Conference on System Sciences, vol.3. pp. 189–198. IEEE Comput. Soc. Press (1996).