

Comparison of Selected Modeling Notations for Process, Decision and System Modeling

Krzysztof Kluza, Piotr Wiśniewski, Krystian Jobczyk, Antoni Ligeża
AGH University of Science and Technology
al. A. Mickiewicza 30, 30-059 Krakow, Poland
E-mail: {kluza,wpiotr,jobczyk,ligeza}@agh.edu.pl

Anna Suchenia (Mroczek)
Cracow University of Technology
ul. Warszawska 24, 31-155 Krakow, Poland
Email: asuchenia@pk.edu.pl

Abstract—System specifications can be modeled using various types of notations and diagrams regarding applications of the particular model. In this paper, we present an overview of the existing solutions, focusing on UML, BPMN and DMN models and the diagrams provided by these notations. We perform a comparison of these approaches and provide examples of representing system requirements in these notations.

Index Terms— Software Engineering, UML, BPMN, DMN, Unified Modeling Language, Business Process Model and Notation, Decision Model and Notation

I. INTRODUCTION

SOFTWARE engineering aims to produce effectively good quality software. Various methods and processes are at the heart of software engineering [1]. In practical software design, parts of systems are specified using visual models. The standard for modeling software applications is Unified Modeling Language (UML). It provides diagrams to capture requirements, collaboration between parts of the software that realize them, the realization itself and models which show how everything fits together and is executed [2].

Business Process Management [3], in turn, is a modern approach to improving organization’s workflow, focused on reengineering of processes to obtain optimization of procedures, increase efficiency and effectiveness by constant process improvement. Business Process Model and Notation (BPMN) is the standard for designing business process models. BPMN can get along with with UML [4], but it does not support modeling of some concepts such as rules. Decision Model and Notation (DMN) provides a standard for modeling decisions and supports decision management and business rules.

The rest of this paper is organized as follows: In Sections II–IV, UML, BPMN, and DMN are introduced. Section V presents the comparison of the notations with the focus on the comparisons from the 4+1 view model architecture perspective. Contributions of the paper are summarized in Section VI.

II. UNIFIED MODELING LANGUAGE (UML)

Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering. Modeling is about capturing a system as models [2], which can be depicted as sets of diagrams. Such diagrams describe the system (or a part of it). UML 2 defines a variety of diagrams divided

The paper is supported by the AGH UST grant.

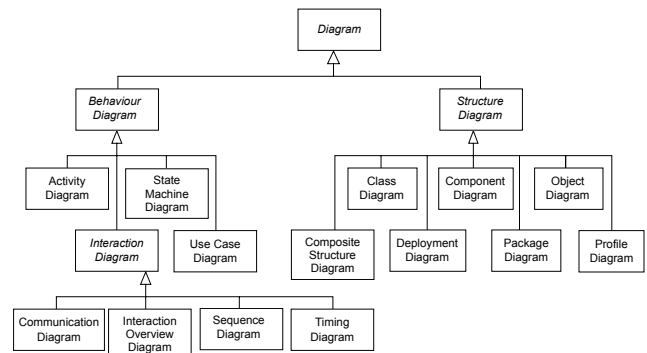


Figure 1: Hierarchy of UML diagram types [5]

into two main categories: *structure diagrams* – containing diagrams representing the structure of a modeled application, and *behavior diagrams* – contain diagrams representing general types of behavior. A tree showing the classification of the UML diagram types [5] is presented in Fig. 1.

The complete system can be described by a number of models describing the system from different angles, often on various levels of abstraction. By design, each UML diagram should be consistent with any other diagram representing the same model. But inconsistency is highly likely to occur in models. Some issues can be resolved using formal methods [6]–[8] or ontologies [9]–[11], but there are also other modeling problems such as exceptions [12] or using reverse engineered models [13].

UML itself is not a design method or a software process. It is only a notation which can be useful within a software process or designing. Another issue is a methodology which indicates how to apply a design. UML itself does not require any specific method, but mostly it is used with an object-oriented design method.

III. BUSINESS PROCESS MODEL AND NOTATION (BPMN)

Business Process Model and Notation (BPMN) [14] is the most widely used notation for modeling business processes. As the notation is quite complex, it has many application areas that may be found in [15]–[20].

The current BPMN 2.0 specification [21] provides four different types of diagrams:

- 1) Process diagram (describing the ways in which operations are carried out to accomplish the intended objectives of an organization),
- 2) Collaboration diagram (presenting the collaborative public Business 2 Business process),
- 3) Conversation diagram (which specifies the logical relation of message exchanges),
- 4) Choreography diagram (defining the expected behavior between two or more interacting business participants in the process).

In most cases, using only the process model is sufficient. The process model uses four basic categories of elements to model BPs: flow objects (activities, gateways, and events), connecting objects (sequence flows, message flows, and associations), swimlanes, and artifacts as shown in Figure 2.

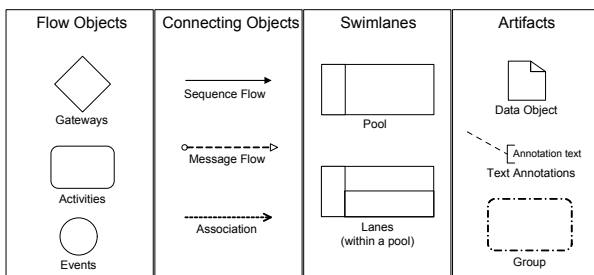


Figure 2: BPMN core elements of Process diagram

In the case of flow object elements, activities denote tasks that have to be performed, events indicate something that happens during the lifetime of the process, and gateways determine forking and merging of the sequence flow between tasks in a process, depending on some conditions. The sequence flow between flow objects is used to model the flow of control in a process. The message flow between selected elements is used to model the flow of messages between participants of a process (which are depicted as different pools).

BPMN 2.0 defines more than 100 elements, thus practitioners differentiate them based on the degree of model detail. Three levels of models can be distinguished [22]: a descriptive level, which is the basic level that uses a very intuitive subset of BPMN to reflect a “happy path” scenario and all major activities in a process; an analytical level, dedicated to analysts, modelers and business architects that use complex structures and elements to design fully representative processes, and an executable level for technicians in which execution details can be captured in the model. Additionally, many different extensions of BPMN were proposed to capture other aspects of business processes [23]–[28].

IV. DECISION MODEL AND NOTATION

DMN [29] is a brand new OMG standard for decision modeling. Such a decision determines the result (or selects some option) based on some input data. Its goal is to provide the notation for decision modeling so the decision can be easily presented in diagrams and understandable by business users [30]. The main purposes of the notation are:

modeling human decision-making, modeling the requirements for automated decision-making, and implementing automated decision-making [29]. Such decision models can be integrated with BPMN models or exist separately [31].

There are four types of core elements in DMN: Decision, Business Knowledge Model, Input Data, and Knowledge Source (see Fig. 3). Decision elements are used to determine an output from a number of inputs using some decision logic. Business Knowledge Model elements denote functions encapsulating business knowledge (like decision table, business rules or analytic models). Input Data elements are used for modeling the input of a Decision or Business Knowledge Model when values are defined outside of the decision model. Knowledge Source elements model authoritative knowledge sources in a decision model. These elements can be connected using different requirement connectors. There are three different types of them: Information, Knowledge, and Authority.

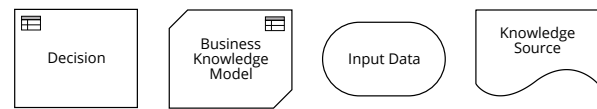


Figure 3: The types of DMN elements

The decision model is usually represented as *Decision Requirements Graph (DRG)*. DRG can be split into one or more *Decision Requirements Diagrams (DRD)* presenting a particular view of the model [29].

DMN provides a wide range of tools (various types of decision logic representation, Elements, and Requirements) to implement decision-making, automated or not. It can be easily adjusted and understood. It fills the gap in the market of decision modeling and is often used with BPMN.

V. COMPARISON OF UML, BPMN AND DMN

We compare the diagrams of the UML, BPMN and DMN notations using the evaluation Framework for BPM/ISM technique [33] and comparing the diagrams in terms of system specification views, especially focusing on the “4+1” view model architecture [32].

Software architecture deals with abstraction, with composition and decomposition. To describe such architecture, a “4+1” model is often used. The model was designed by Philippe Kruchten and used for “describing the architecture of software-intensive systems, based on the use of multiple, concurrent views” [32]. The views are used to describe the system from the viewpoint of different users (end-users, developers and project managers) [32], [34]. The “4+1” view model supports five main views, as shown in Figure 4 and in Table:

1. *Logical View* – an object model of the design.
2. *Process View* – concurrency and synchronization aspects.
3. *Development View* – static organization of the software.
4. *Physical View* – mapping of the software to the hardware.
- +1 *Use-cases view* – various usage scenarios.

The Logical View (Object-oriented Decomposition) and the Process View are at a conceptual level and are used from analysis to design [35]. This view focuses on realizing an application’s functionality in terms of structural element, key abstractions and mechanisms, distribution of responsibilities and separation of concerns. Users-architects use this view for functional analysis [35]. The Process View (process decomposition) [36] captures the concurrency and synchronization aspects of the design. Development View describes the static organization of the software in its development environment [35]. The Physical view (mapping software to hardware) describes the mapping(s) of the software onto the hardware and reflects its distributed aspect [36]. Use case view presents functionality of the system, its external interfaces, and principal users of the system.

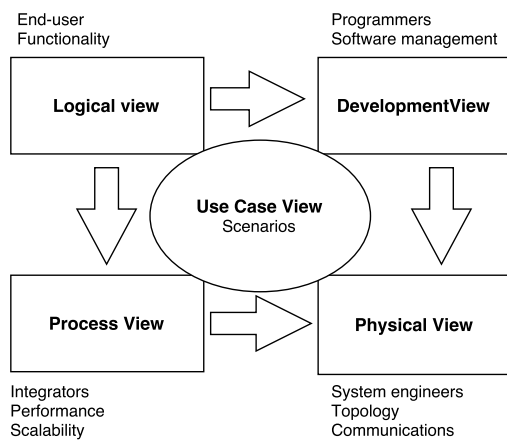


Figure 4: The "4+1" view model architecture

Similarly, the evaluation framework for BPM/ISM [33] is not intended to be rigid, as the lines between depth and breadth of modeling are blurred and hard to be separated.

Our evaluation of the UML, BPMN, and DMN notations in terms of the Giaglis evaluation framework is presented in Table II. Table III presents the comparison of the diagrams in these notations, especially focusing on their application in “4+1” view model architecture [32].

VI. SUMMARY

This paper has given an overview and provided a comparison of the most popular notations for modeling systems, processes, and decisions, i.e. the UML, BPMN and DMN notations. These results have been presented in terms of the "4+1" view model architecture. Further specification of the contribution is going to be a subject of future research to find the appropriate modeling methods for particular systems.

REFERENCES

- [1] Y. Dittrich, “What does it mean to use a method? towards a practice theory for software engineering,” *Information and Software Technology*, vol. 70, pp. 220–231, 2016.
- [2] D. Pilone and N. Pitman, *UML 2.0 in a Nutshell*. O’Reilly, 2005.
- [3] M. Weske, *Business Process Management: Concepts, Languages, Architectures 2nd Edition*. Springer, 2012.
- [4] L. Aversano, C. Grasso, and M. Tortorella, “Managing the alignment between business processes and software systems,” *Information and Software Technology*, vol. 72, pp. 171–188, 2016.
- [5] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition*. Addison Wesley, 2003.
- [6] T. Szmuc and M. Szpyrka, “Formal methods—support or scientific decoration in software development?” in *Mixed Design of Integrated Circuits & Systems (MIXDES), 2015 22nd International Conference*. IEEE, 2015, pp. 24–31.
- [7] R. Klimek, “Towards deductive-based support for software development processes,” in *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*. IEEE, 2013, pp. 1389–1392.
- [8] P. Szwed, “Efficiency of formal verification of archimate business processes with nusmv model checker,” in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*. IEEE, 2015, pp. 1427–1436.
- [9] Z. Rybola and R. Pergl, “Towards OntoUML for software engineering: transformation of rigid sortal types into relational databases,” in *Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on*. IEEE, 2016, pp. 1581–1591.
- [10] G. Nalepa, M. Slazynski, K. Kutt, E. Kucharska, and A. Luszpaj, “Unifying business concepts for smes with prosecco ontology,” in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*, Sept 2015, pp. 1321–1326.
- [11] P. Ziemia, J. Jankowski, J. Wątróbski, W. Wolski, and J. Becker, “Integration of domain ontologies in the repository of website evaluation methods,” in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*. IEEE, 2015, pp. 1585–1595.
- [12] R. Klimek, P. Skrzynski, and M. Turek, “On some problems with modelling of exceptions in UML,” in *Software Engineering: Evolution and Emerging Technologies*, 2005, pp. 87–98.
- [13] A. M. Fernandez-Saez, M. Genero, M. R. Chaudron, D. Caivano, and I. Ramos, “Are forward designed or reverse-engineered uml diagrams more helpful for code maintenance?: A family of experiments,” *Information and Software Technology*, vol. 57, pp. 644–663, 2015.
- [14] M. Chinosi and A. Trombetta, “BPMN: An introduction to the standard,” *Computer Standards & Interfaces*, vol. 34, no. 1, pp. 124–134, 2012.
- [15] T. Kruzel and J. Werewka, “Application of BPMN for the PMBOK standard modelling to scale project management efforts in IT enterprises,” in *Information systems architecture and technology: information as the intangible assets and company value source*. Z. W. et al., Ed. Wroclaw: Oficyna Wydawnicza Politechniki Wroclawskiej, 2011, pp. 171–182.
- [16] A. Ligeza, “A note on a logical model of an inference process : from ARD and RBS to BPMN,” in *Knowledge acquisition and management*, 232nd ed., ser. Research Papers of Wrocław University of Economics, M. L. O. Małgorzata Nycz, Ed. Wrocław : Publishing House of Wrocław University of Economics, 2011, pp. 41–49, ISSN 1899-3192.
- [17] D. Lubke, K. Schneider, and M. Weidlich, “Visualizing use case sets as BPMN processes,” in *Requirements Engineering Visualization, 2008. REV ’08.*, 2008, pp. 21–25.
- [18] M. Szpyrka, G. J. Nalepa, A. Ligeza, and K. Kluza, “Proposal of formal verification of selected BPMN models with Alvis modeling language,” in *Intelligent Distributed Computing V. Proceedings of the 5th International Symposium on Intelligent Distributed Computing – IDC 2011, Delft, the Netherlands – October 2011*. Springer, 2011, vol. 382, pp. 249–255.

Table I: Diagram and view name based on Kruchten [32]

View	Detail	Stakeholders	Comments
Logical	Subsystems Classes	End Users	Functionality
Implementation	Components, Packaging, Layering	Developer, Project, Manager	Used to be called Development View
Deployment	Topology, Mapping to Platforms	System Engineer	Used to be called, Physical View
Process	Performance, Throughput, Concurrency	System Integrator	It is a Computer Engineering term
Use case	Architecture, Discovery, View Validation	Analyst, Tester	Sometimes called Scenarios

Table II: Comparison of selected modeling approaches

Breadth Depth	Understanding & Communicating	Process Improvement	Process Management	Process Development	Process Execution
Functional	(UML) DMN	(UML) DMN	BPMN DMN	UML BPMN DMN	UML BPMN DMN
Behavioural	BPMN (DMN)	BPMN (DMN)	BPMN (DMN)	BPMN (DMN)	BPMN DMN
Organizational	(DMN)	(BPMN)	(BPMN)	(UML)	—
Informational	UML BPMN DMN	UML BPMN DMN	UML BPMN DMN	UML BPMN DMN	UML BPMN DMN

Table III: Comparison of diagrams

	UML										BPMN			DMN					
	Class diagram	Component diagram	Composite structure diagram	Deployment diagram	Object diagram	Package diagram	Profile diagram	Activity diagram	Communication diagram	Interaction overview diagram	Sequence diagram	State diagram	Timing diagram	Use case diagram	Process diagram	Collaboration diagram	Conversation diagram	Choreography diagram	Decision Requirements diagram
Structure	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Behaviour																			
Requirements																			
Implementation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4+1 Logical view	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4+1 Process view																			
4+1 Development view	(✓)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4+1 Physical view			(✓)	✓												(✓)			
4+1 Scenarios					(✓)		(✓)		(✓)	(✓)			✓	(✓)	✓	(✓)		(✓)	✓

✓ – the diagram supports or is used in the particular view (✓) – the diagram partially supports or can be used as an additional element in the view

[19] C. Arevalo, M. Escalona, I. Ramos, and M. Domínguez-Muñoz, “A metamodel to integrate business processes time perspective in bpmn 2.0,” *Information and Software Technology*, vol. 77, pp. 17–33, 2016.

[20] M. Trkman, J. Mendling, and M. Krisper, “Using business process models to better understand the dependencies among user stories,” *Information and Software Technology*, vol. 71, pp. 58–76, 2016.

[21] OMG, “Business Process Model and Notation (BPMN): Version 2.0 specification,” Object Management Group, Tech. Rep. formal/2011-01-03, January 2011.

[22] B. Silver, *BPMN Method and Style*. Cody-Cassidy Press, 2009.

[23] A. Yousfi, C. Bauer, R. Saidi, and A. K. Dey, “ubpmn: A bpmn extension for modeling ubiquitous business processes,” *Information and Software Technology*, vol. 74, pp. 55–68, 2016.

[24] R. Martinho, D. Domingos, and J. Varajão, “Cf4bpmn: a bpmn extension for controlled flexibility in business processes,” *Procedia Computer Science*, vol. 64, pp. 1232–1239, 2015.

[25] R. M. Pillat, T. C. Oliveira, P. S. Alencar, and D. D. Cowan, “Bpmnt: A bpmn extension for specifying software process tailoring,” *Information and Software Technology*, vol. 57, pp. 95–115, 2015.

[26] K. Kluzza, K. Jobczyk, P. Wiśniewski, and A. Lięza, “Overview of time issues with temporal logics for business process models,” in *Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on*. IEEE, 2016, pp. 1115–1123.

[27] R. Klimek, “Towards formal and deduction-based analysis of business models for soa processes,” in *Proceedings of 4th International Conference on Agents and Artificial Intelligence (ICAART 2012), 6–8 February, 2012, Vilamoura, Algarve, Portugal*, J. Filipe and A. Fred, Eds., vol. 2. SciTePress, 2012, pp. 325–330.

[28] R. Klimek, “A system for deduction-based formal verification of workflow-oriented software models,” *International Journal of Applied Mathematics and Computer Science*, vol. 24, no. 4, pp. 941–956, 2014.

[29] OMG, “Decision model and notation. beta.1,” Object Management Group, Tech. Rep. dtc/2014-02-01, 2014.

[30] J. Taylor, A. Fish, J. Vanthienen, and P. Vincent, *iBPMS: Intelligent BPM Systems: Intelligent BPM Systems: Impact and Opportunity*, ser. BPM and Workflow Handbook Series. Future Strategies, Inc., 2013, ch. Emerging standards in decision modeling – An introduction to decision model & notation, pp. 133–146.

[31] T. Debevoise, J. Taylor, J. Sinur, and R. Geneva, *The MicroGuide to Process and Decision Modeling in BPMN/DMN: Building More Effective Processes by Integrating Process Modeling with Decision Modeling*. CreateSpace Independent Publishing Platform, 2014.

[32] P. B. Kruchten, “The 4+ 1 view model of architecture,” *IEEE software*, vol. 12, no. 6, pp. 42–50, 1995.

[33] G. M. Giaglis, “A taxonomy of business process modeling and information systems modeling techniques,” *International Journal of Flexible Manufacturing Systems*, vol. 13, no. 2, pp. 209–228, 2001.

[34] R. Wendler, “Development of the organizational agility maturity model,” in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*. IEEE, 2014, pp. 1197–1206.

[35] V. Muchandi, “Applying 4+ 1 view architecture with uml 2,” *FCGSS White Paper*, 2007.

[36] M. Salehie, “Software architecture,” 2004. [Online]. Available: http://cic.javerianacali.edu.co/wiki/lib/exe/fetch.php?media=materias:mazeiar-kruchten-4_1.pdf