

Towards UML representation for BPMN and DMN models

Anna Suchenia^{1,*}, Paweł Łopata², Piotr Wiśniewski^{2,**}, and Bernadetta Stachura-Terlecka²

¹Cracow University of Technology, ul. Warszawska 24, 31-155 Krakow, Poland

²AGH University of Science and Technology, al. A. Mickiewicza 30, 30-059 Krakow, Poland

Abstract. Unified Modeling Language (UML) is a standardised Object Management Group (OMG) notation among software engineers. There have been many attempts to design alternatives to UML. Recently, new notations such as Business Process Model and Notation (BPMN) and Decision Model and Notation (DMN) have been proposed for modelling processes and decisions. These dedicated notations provide a simpler way of capturing functional requirements in a designer-friendly fashion. Moreover, some concepts as rules cannot be directly modelled in UML. Our research considers a method of translating business models such as BPMN and DMN into a set of consistent UML models, which can be later used by business analysts and developers to understand and implement the system. As a single notation design, it allows the user to take advantage of software supporting UML modelling and consistency checking, in addition using such translation can provide additional insights into OMG business models. Furthermore, the design provided in a single notation can be easier to follow and develop. This approach supports also visualisation of rules. It is important to mention that the proposed solution does not extend any custom UML artifacts and can be used with standard UML tools.

1 Introduction

Software and Knowledge Engineering use different methods and tools to model and build systems. In the case of process and rules (or decision) integration [1], there exist solutions for designing [2] and executing such models [3]. Currently, rules can be also specified using the OMG notations, such as DMN. Modelling decisions in DMN is not as popular as using UML in software engineering projects [4]. Thus we focus on discussing and analysing the potential for the practical application of UML in creating an integrated model of business processes and decisions. We would like to cover the whole model in UML. Both BPMN process and DMN decision should be represented in UML, including DMN decision logic.

The paper is structured as follows: Section 2 gives an overview of the selected three OMG notations, namely UML, BPMN, and DMN, which will be later used in the approach; Section 3 presents the existing related works concerning translations between these notations; Section 4 briefly explains the mapping between BPMN and UML models; Section 5 presents the translation from DMN to UML; Section 6 provides an illustrative case study; while the final section concludes the paper.

2 Object Management Group Notations

In this section, we briefly present the most important OMG notations, *i.e.*, UML, BPMN, and DMN.

2.1 Unified Modeling Language (UML)

Unified Modeling Language (UML) is a language of specification, visualisation, construction and

documentation of projects related to information systems that use software intensively [5]. The main purpose of UML is object-oriented software design. Nowadays, UML is widely used and is one of the most powerful and flexible notations available for software modelling and design [6]: it helps to manage complexity, reduce manufacturing time and improve system quality.

UML meets one of the requirements of business system modelling: it reflects different views of the business system to capture its various aspects. However, UML is a notation intended for computer engineers, so it can be difficult to understand by people not related to the field of computer science.

Activity diagrams are commonly used in the field of business or information flow design [7, 8], as well as in the behavioural design of software and systems embedded in concurrent hardware and software [9]. The designed systems are usually a set of activities performed by external actors or internal states [10].

Another diagram is a class diagram showing the structure of the system, which is the basis for its construction [11]. The class diagram presents a set of classes, interfaces and co-operations and their mutual relations (including inheritance, aggregation and association), as well as class operations and attributes. It is a graph composed of vertices (classes, interfaces, co-operations) and arcs (represented by relations) [12].

2.2 Business Process Model and Notation (BPMN)

Business Process Model and Notation (BPMN) is the most used notation for business process modelling [13].

* Corresponding author: asuchenia@pk.edu.pl

** Corresponding author: wpiotr@agh.edu.pl

Although processes can be modelled in various ways [14], BPMN is a tool for managing business processes and constitutes a modern approach to improving work organisation [15]. The current standard (BPMN 2.0) allows creating: process diagram, cooperation diagram, conversation diagram, choreography diagram.

BPMN has been created for business people and in this area it presents more aspects than the UML notation (Activity Diagram). BPMN and its application is limited to processes only. It describes in more detail the elements dedicated to business people who are not IT specialists. BPMN does not reflect the rules and the decision process, especially logic and decision tables, however, these are not native to either BPMN or UML.

2.3 Decision Model and Notation (DMN)

Decision Model and Notation (DMN) [16] is a brand new OMG standard used for modelling decisions. Based on some input data, this decision determines the result (or selects certain options). Its purpose is to provide notation for decision modelling so that the decisions presented in the diagram are understandable by business users [17]. Thanks to the use of DMN notation, business people have the opportunity to model and manage business logic. The main goals of DMN notation are: creating and modelling of human decision-making processes, automation and implementation of the decision-making process as well as monitoring and management of the decision-making process [16].

Such decision models can be integrated with BPMN models or can exist separately [18]. The decision model is usually represented as Decision Requirements Diagram (DRD). DRG can be divided into one or more decision requirement diagrams, which present a particular view of the model [16]. The decision model in DMN notation is created on two levels: the level of decision requirements and the level of decision logic. There are nine structural components in the DMN that are divided into three types: elements, requirements and artifacts. DMN provides a wide range of tools enabling the implementation of the decision-making process and its automatibility. DMN is easy to understand and adapt to users' needs. As can be seen, in comparison with the UML language or the BPMN notation, DMN notation is exceptionally simple. It fills a gap in the decision modelling market and is often used in combination with BPMN.

3 Related works

BPMN is currently one of the most popular business process modelling languages [15]. In turn, UML is a stable standard of software implementation. One of the basic UML diagrams is an activity diagram that has the same goal in BPMN notation, which is why these two can be considered as interchangeable. The BPMN business model and UML activity diagram have many common features. Both notations are used for modelling workflows or business processes. For each of these models, it is possible to translate to another language, although under certain conditions. However, it is difficult to choose which models, BPMN or Activity Diagram (AD), is better because in both cases there are

certain limitations. Therefore, in the design process, an effective method of translation from BPMN models to UML diagrams is necessary.

Available literature contains various works presenting methods for such a transformation. The paper [19] proposes a method for translating the BPMN model into a sequence diagram

In turn, the subject of translation from the BPMN model to the UML activity diagram is frequently undertaken, *e.g.* [20, 21, 22], because this method of translating the BPMN model into the UML model would bring many benefits to designers and the business environment (the company).

In [23], UML and BPMN were compared based on complexity. It was found that BPMN has a very high level of complexity as opposed to UML. The paper [24] presents the results of the comparison of both notations during the process of creating an application model by business users, which indicate that the Activity Diagram is in any case as useful as the BPMN model.

Typical problems associated with the transformation of the BPMN diagram into the UML activity diagram can be found in [25]. One of the key features of the described transformation is that it is independent of the tool. This feature was achieved by using the XML metadata exchange representation of both models as input and output and using XSLT transformation. In [26] complex transformation rules from BPMN to UML were presented and the mapping rules for the BPMN gate are described for the element of the UML Activity Diagram. A set of stereotypes to strengthen the semantic translated element was also presented. As well as the method of transforming the BPMN model into UML AD.

In [27] they proposed automatic translation from BPMN models into UML activity diagrams. The work also describes the challenges associated with the definition and implementation of such translation.

In the case of translation of rules to UML, there is a solution providing the algorithm of such translation, used in [28], which was introduced in [29].

4 Mapping BPMN to UML

This section describes the method for translating the BPMN model into the UML activity diagram. The presented method of translation allows the designer and people involved in the creation process to gain some benefits, like shortening the communication time or seamlessly synchronised BPMN model with the implementation. However, there is a gap between BPMN and UML, which can limit the method of translation.

The mapping specification, based between BPMN and UML is presented in Tables 1-4.

5 Mapping DMN to UML

In the case of DMN to UML mapping, if it comes to the DMN decision diagram, it is straightforward, as presented in Table 5.

Table 1. Mapping of activities.

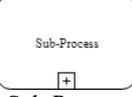
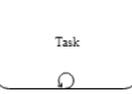
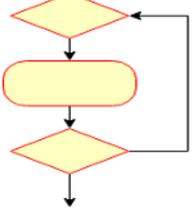
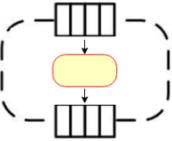
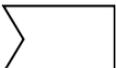
| BPMN | UML |
|---|--|
|  Task |  Action |
|  Sub-Process |  Call Action |
|  Loop |  |
|  Multiple instances |  |

Table 2. Mapping of events.

| BPMN | UML |
|---|--|
|  Start event |  Initial node |
|  End event |  Final node |
|  Terminate |  Destruction |
|  Message standard |  Accept Event Action |
|  Message Catching | |
|  Message End |  Signal |
|  Message Throwing | |

| | |
|--|---|
|  Timer Standard |  Time event |
|  Timer Catching | |

Table 3. Mapping of gates and nodes.

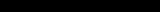
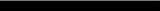
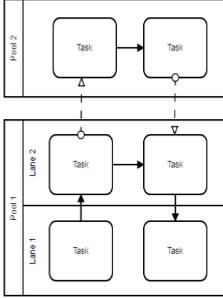
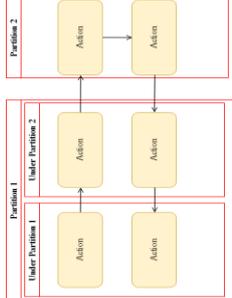
| BPMN | UML |
|---|--|
|  Exclusive gateway |  Condition |
|  Inclusive Gateway | |
|  Parallel Gateway |  |
|  Inclusive Gateway |  |

Table 4. Mapping of objects and flows.

| BPMN | UML |
|---|--|
|  Sequence Flow |  Activity Edge |
|  Message Flow | |
|  |  |

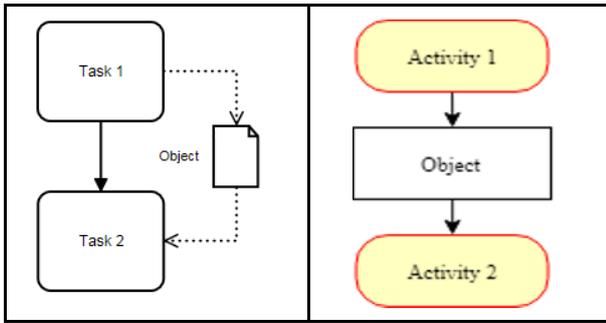


Table 5. Mapping of DMN and UML elements.

| DMN | UML |
|-----|-----|
| | |
| | |
| | |
| | |

However, in the case of DMN decision logic, the solution is not as simple. Decision logic in DMN is represented using decision tables, as presented in Fig. 1.

| Determine rented car category | | | | View DRD |
|-------------------------------|---------|---------|----------------------|------------|
| rentedcarcategory | | | | |
| U | Input + | | Output + | |
| | Age | Period | Category | Annotation |
| | integer | integer | set | |
| 1 | < 18 | - | "none" | - |
| 2 | < 21 | < 2 | "A" | - |
| 3 | < 21 | < 2 | {"A", "B"} | - |
| 4 | >= 21 | >= 2 | {"A", "B", "C"} | - |
| 5 | >= 21 | >= 2 | {"A", "B", "C", "D"} | - |
| + | - | - | - | - |

Fig. 1. DMN decision table.

The translation algorithm is the adaptation of the algorithm presented in [28]:

- 1) Every input/output attribute becomes an input or output parameter respectively,
- 2) For each attribute, if there is more than one unique cell value, a decision node is added and the control flows have guard conditions with these values. If the value occurs more than once, the flow is finished with a fork node.
- 3) For each DMN rule, a join node with the number of inputs equal to the number of input parameters is added. For each join node:
 - a) inputs are connected using the appropriate flow control (according to the values of attributes in the rule), and
 - b) outputs are connected using a flow control with the action having a value corresponding to the output attribute in the rule (directly or through a merge node, depending how many times the value occurs).
- 4) Outputs of all actions are merged in a merge node and a control flow leads to the corresponding activity output parameter.

6 Case study

To illustrate the transformation process, an example of the BPMN model was developed which presents the process of determining the category of a car, which based on the age and period of possession of a driving license can be borrowed by a given driver (Fig. 2).

Next, using the transformation rules contained in Tables 1-4, the BPMN model was transformed into a

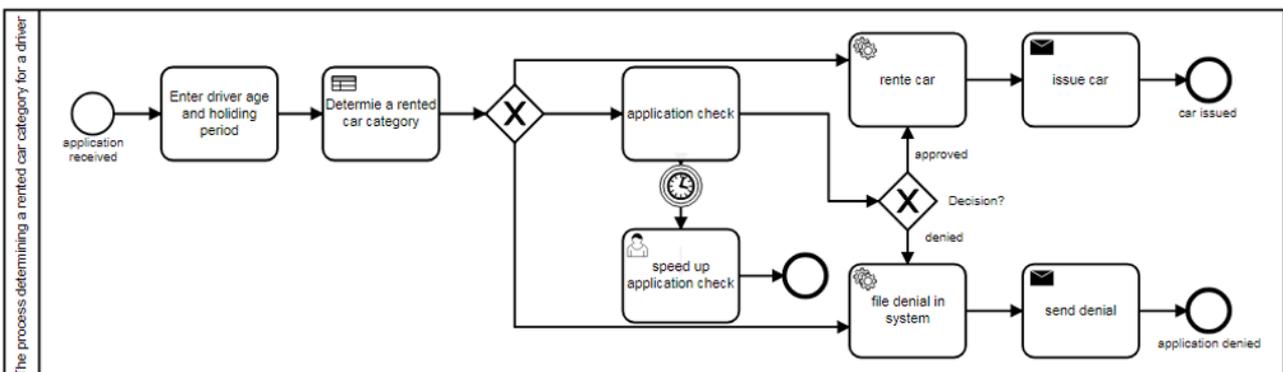


Fig. 2. BPMN model for car rental.

UML activity diagram (Fig. 3). For the presented example, a DMN model providing the decision for the process was presented in Fig. 4. The corresponding UML model is presented in Fig. 5 and 6. For the DMN decision logic specified in the decision table presented in Fig. 1, its UML representation was shown in Fig. 7. Thus, the entire integrated BPMN and DMN model can be represented in the pure UML notation.

7 Conclusion

Designing business systems requires process and decision models. We developed a mapping from BPMN and DMN models to UML models. BPMN and its application are limited only to processes, while DMN reflects the rules and the decision process. However, UML constitutes a general notation, thus the mapping can be especially useful when modelling the entire system in UML. The limitation of our method is that for business users, UML is too technical and complex in the preparation of models.

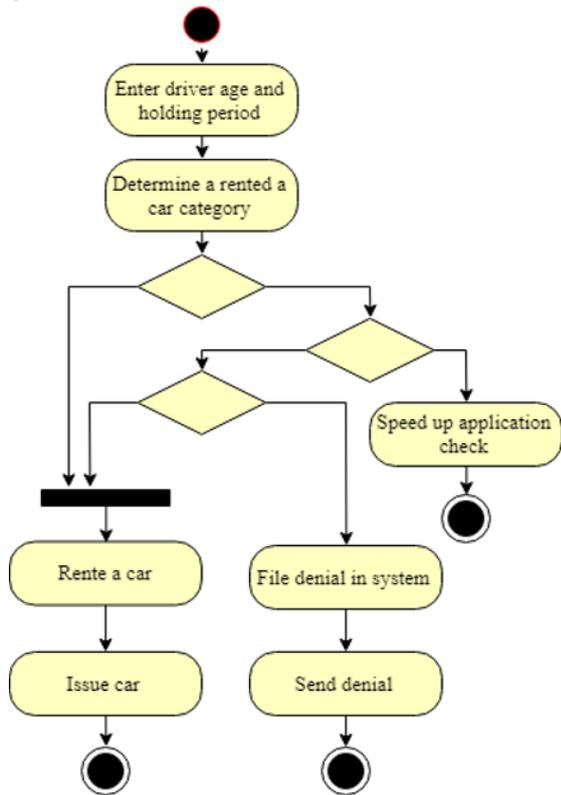


Fig. 3. Activity diagram for the car rental.

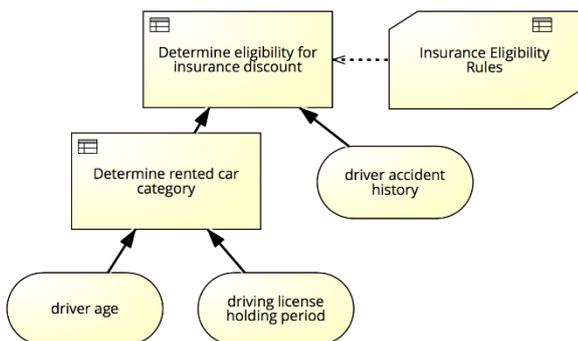


Fig. 4. DMN decision diagram.

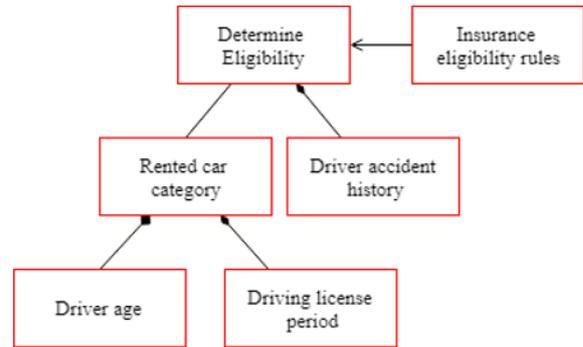


Fig. 5. UML representation of DMN decision diagram.

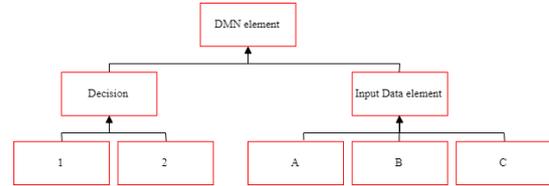


Fig. 6. Details of the DMN data attributes represented in the UML diagram (1 – Rented car category, 2 – Determine Eligibility for insurance discount, A – driver age, B – driver license holding period, C – driver accident history).

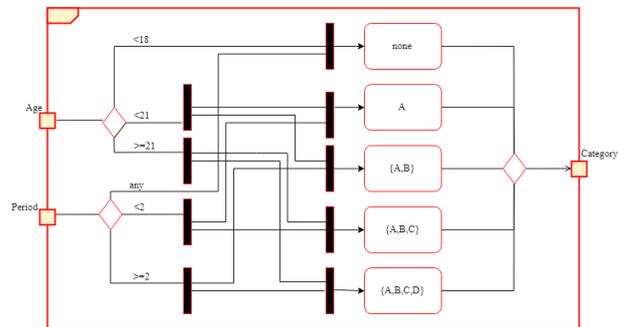


Fig. 7. DMN decision table represented as UML diagram.

References

1. K. Kluza, G. J. Nalepa, Information and Software Technology **91**, 123 (2017)
2. F. Hasić, J. De Smedt, J. Vanthienen, Decision Support Systems, **107**, 1 (2018).
3. G. J. Nalepa, K. Kluza, K. Kaczor, Proposal of an inference engine architecture for business rules and processes, In: *International Conference on Artificial Intelligence and Soft Computing*, 453 (Springer, 2013)
4. K. Kluza K., P. Wiśniewski, K. Jobczyk, A. Ligęza, A. Suchenia, Comparison of selected modeling notations for process, decision and system modeling, In: *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 1095 (IEEE, 2017)
5. Object Management Group. *UML*. (OMG, 1996)
6. R. Klimek, Towards deductive-based support for software development processes, In: *2013 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 1389 (IEEE, 2013)

7. R. Eshuis, R. Wieringa, A Comparison of Petri Net and Activity Diagram Variants, In: *Proc. of 2nd Int. Coll. on Petri Net Technologies for Modelling Communication Based Systems*, 93 (2001)
8. Y.L. Chang, S. Chen, CC. Chen, I. Chen, Workflow process definition and their applications in e-commerce, In: *2000 Proceedings of International Symposium on Multimedia Software Engineering*, 193 (IEEE, 2000)
9. T. Schattkowsky, UML 2.0 – Overview and Perspectives in SoC Design, In: *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'05)* (2005)
10. G. Łabiak, M. Adamski, Zastosowanie języka UML w modelowaniu sterownika dyskretnego, In: *KNWS '06*, 50 (2006)
11. K. Hinkelmann, *UML Class Diagrams* (2014)
12. S.W. Ambler, *UML 2 Class Diagrams: An Agile Introduction*, <http://www.agilemodeling.com/artifacts/classDiagram.htm> (accessed: 2018-06-15)
13. M. Chinosi, A. Trombetta, *Computer Standards & Interfaces* **24**, 124 (2012)
14. K. Kluza, P. Wiśniewski, Spreadsheet-Based Business Process Modeling, In: *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 1355 (IEEE, 2016)
15. OMG. *Business Process Model and Notation (BPMN): Version 2.0 Specification*. (OMG, 2012)
16. Object Management Group. *Decision model and notation. Beta1* (OMG, 2014)
17. J. Taylor, A. Fish, J. Vanthienen, P. Vincent, *iBPMS: Intelligent BPM Systems: Intelligent BPM Systems: Impact and Opportunity, ser. BPM and Workflow Handbook Series. Future Strategies*. (WfMC, 2013)
18. T. Debevoise, J. Taylor, J. Sinur, 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)
19. A. Suchenia, K. Kluza, K. Jobczyk, P. Wiśniewski, M. Wypych, A. Ligęza, Supporting BPMN Process Models with UML Sequence Diagrams for Representing Time Issues and Testing Models, In: *Lecture Notes in Computer Science* **10246**, 589 (2017)
20. D. Peixoto, V. Batista, A. Atayde, E. Borges, R. Resende, C. I. P. S. Pádua, A comparison of BPMN and UML 2.0 activity diagrams, In: *VII Simposio Brasileiro de Qualidade de Software* **56**, 012010 (2008)
21. C. V. Geambasu, *Accounting and Management Information Systems* **11**, 637 (2012)
22. L. Eloranta, E. Kallio, I. Terho, A notation evaluation of BPMN and UML activity diagrams, In: *Special course in information systems* (SoberIT, 2006)
23. J. C. Recker, M. zur Muehlen, K. Siau, J. Erickson, M. Indulska, Measuring method complexity: UML versus BPMN, In: *Association for Information Systems* (AIS, 2009)
24. D. Birkmeier, S. Kloeckner, S. Overhage, An Empirical Comparison of the Usability of BPMN and UML Activity Diagrams for Business Users, In: *ECIS 2010*, 2 (2010)
25. O. Macek, K. Richta, The BPM to UML activity diagram transformation using XSLT, In: *Dateso 2009*, 119 (2009)
26. N. Q. Báo, A proposal for a method to translate BPMN model into UML activity diagram, In: *13th International Conference on Business Information Systems* (2010)
27. M.A. Cibrán, Translating BPMN Models into UML Activities, In: *Lecture Notes in Business Information Processing* **17**, 236 (Springer, 2009)
28. M. Januszka, W. Moczulski, Acquisition and knowledge representation in the product development process with the use of augmented reality. In *Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment*, In: *Proceedings of the 19th ISPE International Conference on Concurrent Engineering*, 315 (Springer, 2013)
29. G. J. Nalepa, K. Kluza, *International Journal of Software Engineering and Knowledge Engineering* **22**, 485 (2012)