



# Towards Assessing the Multi-view Modeling Capability of Enterprise Modeling Methods

Afef Awadid<sup>1</sup>(✉), Dominik Bork<sup>2</sup>, and Selmin Nurcan<sup>1</sup>

<sup>1</sup> University of Paris 1 Pantheon-Sorbonne, Paris, France  
afef.awadid@malix.univ-parisl.fr,  
nurcan@univ-parisl.fr

<sup>2</sup> University of Vienna, Vienna, Austria  
dominik.bork@univie.ac.at

**Abstract.** Today's enterprises and their underlying information systems ask for Multi-view Enterprise Modeling Methods (MVMMs) toward a comprehensive model representation. MVMMs capture the required aspects of complex systems using multiple views – the Multi-view Modeling (MVM) capability. However, not all modeling methods are endowed with a MVM capability. Means for assessing and improving such capability are therefore needed. Based on a comparative analysis of three MVMMs, we define the notion of MVM capability. Drawing on these criteria, an EBNF-based description is proposed, serving as a basis for MVM capability assessment. The strengths of the approach go beyond offering a common understanding of the MVM capability notion by (i) assessing the MVM capability, and (ii) identifying requirements to achieve this capability. Consequently, this approach primarily addresses method engineers aiming to employ MVM capability to a modeling method.

**Keywords:** Multi-view modeling · Capability · Assessment  
Business process modeling · Enterprise modeling · Comparative analysis  
EBNF rules

## 1 Introduction

In light of the complexity of today's enterprise and information systems owing to globalization and fierce competition amongst businesses, the need for Multi-view Modeling (MVM) to cope with such complexity in enterprise modeling is undisputed. MVM captures different aspects of the modelled system (e.g. its structure and its behavior) by different views (models) [1]. Each view (i) sheds light on certain aspects of the system, and (ii) is specified by a viewpoint which depicts the concepts considered by the view and the valid combinations (e.g. specified by a metamodel) [2].

MVM capability refers to how well an enterprise modeling method supports MVM [3]. This capability is embraced by what is commonly known as multi-view modeling methods (MVMMs). Enterprise Knowledge Development (EKD) [4], Function, Information, Dynamics, and Organization (FIDO) [5], and the Semantic Object Model (SOM) [6] are sample MVMMs from the enterprise modeling domain. All of them represent an enterprise by several interrelated views. However, when focusing the business process, not all are endowed with MVM capability.

Extant approaches focus on improving the MVM capability of a given method, usually eluding the question of assessment. This observation spawned the research that culminated in this paper. The overall question addressed in the present work is: *How to assess the MVM capability of a modeling method?* As a step towards an answer, we perceive the MVM capability as a quality criterion and decompose it into more fine-grained criteria. This idea has emerged from our previous work on the quality of business process modeling methods [7]. To identify the criteria, a comparative analysis of three MVMMs has been performed. Based on the analysis results, a formalized description for the MVM capability is presented that enables assessing the MVM capability of any modeling method.

Our approach can (i) assess the MVM capability of a modeling method, and (ii) identify requirements for adopting it in the context of method engineering. Method engineers aiming to introduce MVM capability benefit from this research in two ways: First, the formalized description enables the assessment of the MVM capability of a method. Second, our analysis reveals different ways of realizing MVM.

This paper is structured as follows: Sect. 2 provides the terminology used in the paper and briefly presents related works. Section 3 puts emphasis on the analysis of three MVMMs. In Sect. 4, a formalized description for assessing the MVM capability of modeling methods is defined and applied. Finally, we conclude the paper with an outlook on future work.

## 2 Terminological Foundations and Related Works

### 2.1 Multi-view Modeling: Key Terms

The viewpoint refers to the modelling language used to specify a view. A modeling language is syntactically defined by a meta model. Each view is represented by a conceptual model, and is specified by a viewpoint. The relationship between view and viewpoint is thus analogous to that between model and meta model [8]. A view allows capturing perspective(s). A perspective refers to certain aspect(s) from which the system under study can be viewed [9] (e.g., behavior and structure). These definitions given to the terms “viewpoint”, “view” and “perspective” are respectively in line with the terms “viewpoint”, “view” and “concern” as provided by ISO/IEC/IEEE 42010 [10]. The latter defines these three terms as follows: A viewpoint: is a work product establishing the conventions for the construction, interpretation, and use of architecture views to frame specific system concerns. A view is a work product expressing the architecture of a system with respect to specific system concerns. A concern is “any interest in the system”. The term “concern” is usually associated with the notion of “separation of concerns”. Separation of concerns means dealing with different aspects of a system individually. A separation of concerns is of type “horizontal” if the considered concerns belong to the same level of abstraction/phase of development, or otherwise of type “vertical” [11].

The notion of “capability” gains quite a lot of attraction in conceptual modeling (e.g., “capability-oriented information systems”, “capability-driven development”). A comprehensive paper on the variety of interpretations of this notion is [12]. In this

paper, we introduce the notion of “MVM capability”. The latter is an instance of a more generic notion: “modeling method capability”. In its broad sense, “modeling method capability” refers to the degree to which a method is able to achieve the goal(s) of modeling. A “modeling method capability” is therefore a quality criterion that is related to “Goals of modeling G”: one of the main parts of the generic quality framework defined by [13]. When the modeling goal is to reduce the complexity of the system under study and hence to foster a better understanding of it, a specific notion is employed: “MVM capability”. Based on this, MVM capability is defined *as the ability of a method to support MVM in order to curtail the complexity of the system under study*.

In MVM, viewpoints and hence views are not independent from each other since they all depict the same system under study. Consequently, inter-viewpoint relationships should be identified in order to manage consistency. Six types of inter-viewpoint relationships have been stressed in [14]. These types are useful in the management of inter-view consistency. By inter-view consistency, we refer to the extent to which information contained in multiple views is not contradicting [14].

## 2.2 Related Work

Most emphasis of the literature dealing with MVM in enterprise and business process modeling, is put on how to improve the MVM capability of a given modeling method. Broadly, each work uses either a unified or an hybrid approach to improve such capability. The former extends an existing unified overarching meta model to cover additional aspects. By contrast, the latter combines distinct modeling languages with separate meta models.

[15, 16] are example works adopting the unified approach. In [15], the emphasis was set on extending the UML Statechart language with security aspects. The focus in [16] was on extending BPMN to cover resource management and planning aspects. Additionally, literature yields works pursuing the hybrid approach. In [17], an integrative approach combines distinct modelling languages like BPMN and ER. The work in [18] illustrates the combination of I-STAR and BPMN.

In recent years, several works also aimed at the comprehensive analysis and comparison of enterprise modeling methods. The work presented in [19] analyses six enterprise modeling methods based on the formality of their specifications and enabled capabilities thereof. The authors in [9] performed a comparative assessment of three modeling methods with regard to criteria like completeness and simplicity. [20] proposed an analysis framework for assessing the explanatory capabilities of enterprise modeling methods. The authors in [21] analyzed four enterprise modeling methods with respect to background and goals of the methods in order to contribute to the elucidation of their overlaps, conceptual differences, and focal points. The work presented in [22] systematically evaluates enterprise modeling methods according to their capability of automatically generating ERP software.

In this paper, we build on these existing works and perform a comparative analysis that targets the MVM capability of enterprise modeling methods in order to contribute to a more comprehensive understanding on the design principles of such methods.

### 3 Analysis of Enterprise Modeling Methods

In this section, we start by defining a MVM analysis framework based on which a comparative analysis of enterprise modeling methods is performed

#### 3.1 A Multi-view Modeling Analysis Framework

Guided by the terminological foundations presented in Sect. 2, an analysis framework is proposed (see Table 1) that accounts for the main particularities of MVM. This framework will steer the comparative analysis that follows. Table 1 points out the analysis criteria along with questions clarifying their intended scopes.

**Table 1.** Analysis framework.

Criterion	Explanation
System subject to multi-view modeling	What is/are the system(s) subject to multi-view modeling?
	What views are being specified by the method?
Viewpoints	What are the different modeling languages employed by the views?
Perspectives	What are the perspectives [9, 23] covered by the viewpoints?
	Which kind of separation of concerns [11] is employed?
Inter-viewpoint relationships	What kinds of relationships [14] exist between the viewpoints?
Inter-view consistency	Is inter-view consistency specified by the modeling method [14]?
	By which mechanisms is it realized [8]?

#### 3.2 Comparative Analysis

In this subsection, the analysis framework is applied to comparatively analyze the EKD [4], FIDO [5] and SOM [6] enterprise modeling methods. These methods are selected as representative of MVMMs. Note that in our analysis, we are referring to the first version of the EKD method, which is significantly different from its successors. Moreover, for the sake of brevity and due to limited space, the analysis concentrates on the process-related aspects of these methods. In other words, considering a representation of an enterprise structure in terms of layers, our analysis targets only the layer “business processes”. The comparative analysis is presented in Table 2.

The results show the heterogeneity of the methods regarding the MVM (e.g. heterogeneity in terms of the perspectives supported and the type of separation of concerns adopted). Hence, different ways of realizing MVM and therefore of managing complexity are possible. At the same time, common characteristics of the investigated methods with respect to MVM can be identified:

- All methods hold at least two modeling languages (viewpoints).
- Different perspectives are covered by the methods.

- Viewpoints and hence views are overlapping.
- A given perspective can be covered by more than one modeling language.
- Each modeling language can capture one or more perspectives.
- Horizontal and/or vertical separation of concerns is employed.
- Inter-view consistency management is a recurring aspect, but treated differently.

**Table 2.** Comparative analysis.

Criterion	EKD [4]	FIDO [5]	SOM [6]
System subject to multi-view modeling	Enterprise	Enterprise	Enterprise
	Intra-enterprise business processes	Inter-enterprise business processes	Enterprise business processes
Viewpoints	Actor-role Role-activity Business objects	FIDO 1 FIDO 4	Interaction schema Task-event schema Transaction decomposition Object decomposition
Perspectives	Behavioral Functional Informational Intentional Organizational	Behavioral Functional Informational Organizational Operational	Informational Behavioral Structural
	Horizontal separation of concerns	Vertical separation of concerns	Horizontal and vertical separation of concerns
Inter-viewpoint relationships	Semantic overlaps Syntactic overlaps	Semantic overlaps Syntactic overlaps Refinement/abstraction	Semantic overlaps Syntactic overlaps Refinement/abstraction
Inter-view consistency	Consistency rules on overarching metamodel	Shared concepts in distinct metamodels	Consistency rules on overarching metamodel
	Modeling guidelines	Conversion mechanisms	Automatic consistency mechanisms

In light of these results, the comparative analysis highlighted in Table 2 can be beneficial for any stakeholder whose purpose is to reduce the complexity of a given system. It indeed helps him in choosing between different ways of managing complexity the one that better meets his specific requirements.

#### 4 An EBNF-Based Description to Assess the MVM Capability

This section introduces an Extended Backus Naur Form (EBNF)-based description of the MVM capability. It then shows how this description can be used to assess the MVM capability of modeling methods. Finally, a sample application evaluates the feasibility and shows its application with the EKD modeling method.

#### 4.1 A Formalized Description of the Multi-view Modeling Capability

Compared to informal assessment, a formal assessment promotes an unambiguous understanding of the notion of MVM capability. To derive such description, we proceeded in two steps: First, we delineate the notion of MVM capability by separating it into its constitutive parts. Second, we used this notion to create a formalized description for assessing the MVM capability based on EBNF.

To delineate the notion of MVM capability (step 1), we chiefly relied on two aspects: On the one hand, the common characteristics of MVMMs drawn from the comparative analysis. On the other hand, we relied on our previous work [7] in which we performed a systematic literature review on the quality of business process modeling methods. Apart from perceiving the MVM capability as a quality criterion, the findings of such review showed that the expressiveness of a modeling method with respect to each required perspective is a basic quality criterion. Based on these two aspects, we define the MVM capability criterion in terms of three complementary sub criteria viz., *the support of separation of concerns principle*, *the support of multi-perspective modeling*, and *the expressiveness with respect to each required perspective*. The support of separation of concerns principle is defined as the ability of a modeling method to separate the covered perspectives into multiple views. The support of multi-perspective modeling refers to the ability of a modeling method to capture all required perspectives. Lastly, the expressiveness criterion refers to the extent to which a modeling method provides all required modeling constructs for each perspective.

Bearing in mind the output of step 1, assessing the MVM capability (step 2) amounts to combining means for assessing all the three sub criteria. As to ‘separation of concerns principle’, the assessment of this criterion can be based on the analysis results, and particularly on one of the identified characteristics: “A MVMM holds at least two modeling languages (viewpoints)”. To assess the criterion “support of multi-perspective modeling”, we will use the Giaglis’ framework [24], as it matches the breadth (the modelling goals) with the depth (the required modeling perspectives). For the assessment of ‘expressiveness with respect to each perspective’, one can refer to one of the several frameworks in the literature. We choose the framework defined in [25] since it is generic, i.e., applicable independently of the modeling goal and because it covers multiple perspectives.

The EBNF description presented in Fig. 1 is a way of coupling the aforementioned means for assessing the three sub criteria. EBNF has the advantages of being simple, formal, and rule-based. The proposed description takes then the form of six EBNF rules that are mainly derived from our comparative analysis. Each rule comprises a Left Hand Side (LHS) and a Right Hand Side (RHS). The LHS refers to the name of the symbol which has to be non-terminal (i.e. non-atomic) and can be replaced by the RHS. The RHS represents the definition of the symbol. It can include terminal symbols, non-terminal symbols, or a combination of both.

<b>Functional_Pers   Behavioral_Pers   Organizational_Pers</b> ::= ({Modeling_Construct}+)	(6)	
<b>Informational_Pers   Operational_Pers   Intentional_Pers</b>		
<b>Perspective</b> ::= Functional_Pers   Behavioral_Pers   Organizational_Pers   Informational_Pers   Operational_Pers   Intentional_Pers	(5)	
<b>Modeling_language</b> ::= Modeling_language_provided_by_the_method	(4)	
<b>Modeling_language</b> ::= (Modeling_language "captures" {Perspective}+)	(3)	
<b>Viewpoint</b> ::= Modeling_language	(2)	
<b>Multi-view_modeling_description</b> ::= (Viewpoint "overlaps with" {Viewpoint}+)	(1)	
<b>Legend of the used notation (Extended Backus Naur Form - EBNF)</b>		
::= Is defined as	{ } A set	() Grouping of a sequence
Alternative	{ }+ Repetition (1,*)	A Non-terminal is capitalized
" " Terminal symbol		

**Fig. 1.** A formalized description of MVM capability using EBNF.

## 4.2 Assessing the Multi-view Modeling Capability

Based on a construction of a proof tree (graphical demonstration), an assessment determines not only whether or not a modeling method is endowed with a MVM capability, but also graphically reveals the criterion (criteria) that establish(es) requirements for improving MVM capability. Considering the proposed EBNF description (Fig. 1), the tree is built in depth-first left-first order. At the root appears the LHS of the rule (1) (i.e. ‘Multi-view\_modeling\_description’). By referring to the root level as level 0, each new tree level is derived from the previous one by following the procedure below:

- If the current node is ‘Modeling\_language\_provided\_by\_the\_method’, replace the node by the actual modeling language provided by the assessed method.
- If the current node is a terminal, keep the node and move to the adjacent node.
- If the current node is a repetition (marked with the ‘ + ’ sign) of a non-terminal, replace the node  $n$  times by this non-terminal. As for {Perspective} +,  $n$  and the required perspectives are determined using the Giaglis framework [24]. “ $\emptyset$ ” is assigned to each not supported perspective. Regarding {Modeling\_Construct} +,  $n$  and the required modeling constructs are determined using the metamodel defined in [25]. With respect to {Viewpoint} +, ( $n = \text{total number of viewpoints provided by the method} - 1$ ). In case  $n$  is equal to zero, replace the node by “ $\emptyset$ ”.
- If the current node is ‘Modeling\_Construct’,  $x$  is the level of this node. Using the aforementioned metamodel, replace the node by the actual required modeling construct at the tree level  $x + 1$ . Then, replace each required modeling construct by its corresponding modeling construct provided by the assessed method at the level  $x + 2$ . If no corresponding construct exists, replace the required construct by “ $\emptyset$ ”.
- If the current node is any other non-terminal, replace the node by its RHS using the EBNF rule number: (level of the current node + 1) if its root node at the level 1 is ‘Viewpoint’ and using the rule number: (level of the current node) if its root node is ‘{Viewpoint} +’. Repeat until the current node becomes a terminal.
- The construction terminates when all leaves are terminals.

Albeit the three sub criteria are all important for evaluating the MVM capability, they are not equally important. This is based on our assumption that: (i) there is no MVM without separation of concerns, (ii) capturing multiple perspectives is relevant as it facilitates understanding complex systems, and (iii) the expressiveness of each perspective is equally important to both customary (single view) modeling and MVM. Thus, ‘support of separation of concerns principle’ is the most discriminating criterion. Whereas, ‘expressiveness’ is the least discriminating one.

Drawing on this, we define an assessment scale as follows: (a) A modeling method has *no MVM capability* if it does not support the separation of concerns principle; (b) A modeling method has a *total MVM capability* if it (1) supports the separation of concerns principle, (2) supports the multi-perspective modeling, and (3) is sufficient expressive; (c) A modeling method has a *strong partial MVM capability* if it supports the separation of concerns principle and the multi-perspective modeling, but has only limited expressiveness; and (d) Otherwise, a modeling method has a *weak partial MVM capability*.

An assessed method requires improvement in terms of (i) ‘Support of separation of concerns principle’, if the symbol “∅” is assigned at least to one ‘Viewpoint’ node; (ii) ‘Support of multi-perspective modeling’, in case the symbol “∅” is assigned to all the occurrences of at least one ‘Perspective’ node, and (iii) ‘Expressiveness, if the symbol “∅” is assigned to all the occurrences of at least one ‘Modeling\_Construct’ node.

### 4.3 Assessing the Multi-view Modeling Capability of EKD

To evaluate the MVM capability of EKD, we opted for an advanced level of assessment by constructing a proof tree. Since the tree is too large to show in the paper, only the part that reveals deficiencies of the method is visualized (Fig. 2). By referring to the Giaglis framework, four perspectives are required to grasp the wider system picture: functional, organizational, behavioral, and informational.

As highlighted in blue in Fig. 2, the Role Activity Language does not support all the required modeling constructs for the functional and the behavioral perspectives. As to the organizational and the informational perspectives, their required modeling constructs are fully supported by respectively the Actor Role Language and the Business Objects Language (the two other parts of the tree). Hence, following the

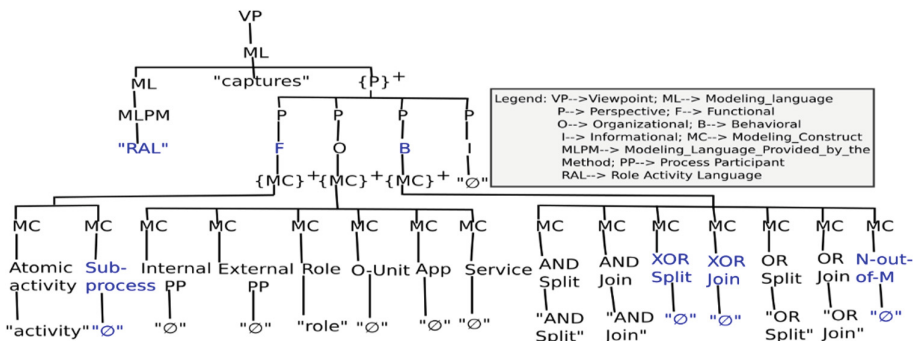


Fig. 2. A part of the constructed tree for assessing the MVM capability of EKD.



assessment scale defined in the previous section, EKD possesses a *strong partial MVM capability*, as only the criterion ‘expressiveness’ needs improvement. The deficiency in expressiveness has also appeared in the part of the tree pertaining to the Actor Role Language, where only one modeling construct (Actor) is provided to represent all types of process participants (internal, external and organizational unit).

## 5 Concluding Remarks

Based on a systematic analysis of three enterprise modeling methods with respect to how they realize multi-view modeling, this paper (1) describes the main characteristics of realizations of multi-view enterprise modeling; and (2) defines the notion of MVM capability in terms of support of separation of concerns principle, support of multi-perspective modeling, and expressiveness. These findings have been used to develop a set of EBNF rules that enable the formalized description of the MVM capability. An application of the rules to the EKD method showed, how they enable the assessment of the MVM capability of modeling methods.

This research establishes a first step towards a common understanding of the MVM capability notion. The results of this research are of primary interest for the assessment of existing methods and method engineers, designing a new multi-view modeling method. Future research will apply the rule base to evaluate more modeling methods in order to improve the EBNF description and to cover also individual aspects that have, for now, been neglected by aiming at general applicability.

## References

1. Reineke, J., Tripakis, S.: Basic problems in multi-view modeling. In: Ábrahám, E., Havelund, K. (eds.) TACAS 2014. LNCS, vol. 8413, pp. 217–232. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-642-54862-8\\_15](https://doi.org/10.1007/978-3-642-54862-8_15)
2. Bork, D., Karagiannis, D.: Model-driven development of multi-view modelling tools the MUVIEMOT approach. In: 9th International Conference on Software Paradigm Trends, p. IS-11. IEEE (2014)
3. Awadid, A.: Supporting the consistency in multi-perspective business process modeling: a mapping approach. In: 11th International Conference on Research Challenges in Information Science, pp. 414–419. IEEE (2017)
4. Loucopoulos, P., Kavakli, V., Prekas, N., Rolland, C., Grosz, G., Nurcan, S.: Using the EKD approach: the modelling component, Paris (1997). <https://hal.archives-ouvertes.fr/hal-00707997>. Accessed 19 May 2018
5. Shunk, D.L., Kim, J.I., Nam, H.Y.: The application of an integrated enterprise modeling methodology—FIDO—to supply chain integration modeling. *Comput. Ind. Eng.* **45**(1), 167–193 (2003)
6. Ferstl, O.K., Sinz, E.J.: *Grundlagen der Wirtschaftsinformatik*, 7th edn. Oldenbourg, München (2013)
7. Awadid, A., Nurcan, S., Ghannouchi, S.A.: Towards a decision-support system for selecting the appropriate business process modeling formalism: a context-aware roadmap. In: *Enterprise, Business-Process, and Information Systems Modeling*, pp. 239–256. Springer (2017)

8. Bork, D., Buchmann, R., Karagiannis, D.: Preserving multi-view consistency in diagrammatic knowledge representation. In: Zhang, S., Wirsing, M., Zhang, Z. (eds.) KSEM 2015. LNCS (LNAI), vol. 9403, pp. 177–182. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-25159-2\\_16](https://doi.org/10.1007/978-3-319-25159-2_16)
9. Daoudi, F., Nurcan, S.: A benchmarking framework for methods to design flexible business processes. *Softw. Process. Improv. Pract.* **12**(1), 51–63 (2007)
10. Systems and software engineering—architecture description: ISO/IEC/IEEE 42010. <http://www.iso-architecture.org/ieee-1471/cm/>. Accessed 19 May 2018
11. Solberg, A., Simmonds, D., Reddy, R., Ghosh, S., France, R.: Using aspect oriented techniques to support separation of concerns in model driven development. In: 29th International Conference on Computer Software and Applications, vol. 1, pp. 121–126. IEEE (2005)
12. Zdravkovic, J., Stirna, J., Grabis, J.: A comparative analysis of using the capability notion for congruent business and information systems engineering. *Complex Syst. Inform. Model. Q.* **10**, 1–20 (2017)
13. Nysetvold, A.G., Krogstie, J.: Assessing business process modeling languages using a generic quality framework. In: *Advanced Topics in Database Research*, pp. 79–93. IGI Global (2006)
14. Persson, M., et al.: A characterization of integrated multi-view modeling in the context of embedded and cyber-physical systems. In: *Proceedings of the Eleventh ACM International Conference on Em-bedded Software*, p. 10. IEEE Press (2013)
15. El-Attar, M., Luqman, H., Karpati, P., Sindre, G., Opdahl, A.L.: Extending the UML statecharts notation to model security aspects. *IEEE Trans. Software Eng.* **41**(7), 661–690 (2015)
16. Meyer, A.: Resource perspective in BPMN: extending BPMN to support resource management and planning. Master’s thesis, Hasso Plattner Institute, University of Potsdam (2009)
17. Letsholo, K.J., Chioasca, E.V., Zhao, L.: An integrative approach to support multi-perspective business process modeling. *Int. J. Serv. Comput.* **2**(1), 11–24 (2014)
18. Koliadis, G., Vranesevic, A., Bhuiyan, M., Krishna, A., Ghose, A.: Combining *i\** and BPMN for business process model lifecycle management. In: Eder, J., Dustdar, S. (eds.) BPM 2006. LNCS, vol. 4103, pp. 416–427. Springer, Heidelberg (2006). [https://doi.org/10.1007/11837862\\_39](https://doi.org/10.1007/11837862_39)
19. Bork, D., Fill, H.G.: Formal aspects of enterprise modeling methods: a comparison framework. In: 47th Hawaii International Conference on System Sciences, pp. 3400–3409. IEEE (2014)
20. Kaczmarek, M., Bock, A., Heß, M.: On the explanatory capabilities of enterprise modeling approaches. In: Aveiro, D., Pergl, R., Valenta, M. (eds.) EEWK 2015. LNBIP, vol. 211, pp. 128–143. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-19297-0\\_9](https://doi.org/10.1007/978-3-319-19297-0_9)
21. Bock, A., Kaczmarek, M., Overbeek, S., Heß, M.: A comparative analysis of selected enterprise modeling approaches. In: Frank, U., Loucopoulos, P., Pastor, Ó., Petrounias, I. (eds.) PoEM 2014. LNBIP, vol. 197, pp. 148–163. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-662-45501-2\\_11](https://doi.org/10.1007/978-3-662-45501-2_11)
22. Schunselaar, D.M., Gulden, J., van der Schuur, H., Reijers, H.A.: A Systematic evaluation of enterprise modelling approaches on their applicability to automatically generate ERP software. In: 18th Conference on Business Informatics, pp. 290–299. IEEE (2016)
23. Awadid, A., Nurcan, S.: *Softw. Syst. Model* (2017). <https://doi.org/10.1007/s10270-017-0629-2>

24. Giaglis, G.M.: A taxonomy of business process modeling and information systems modeling techniques. *J. Flex. Manuf. Syst.* **13**(2), 209–228 (2001)
25. List, B., Korherr, B.: An evaluation of conceptual business process modelling languages. In: *Proceedings of the symposium on applied computing*, pp. 1532–1539. ACM (2006)