# Intuitive Understanding of Domain-Specific Modeling Languages: Proposition and Application of an Evaluation Technique

Dominik Bork and Christine Schrüffer and Dimitris Karagiannis

Published in:

Conceptual Modeling, 38th International Conference, ER 2019, Salvador, Brazil, November 4–7, 2019, pp. 311-319

The final authenticated versionis available online at:
https://doi.org/10.1007/978-3-030-33223-5_26

# Intuitive Understanding of Domain-specific Modeling Languages: Proposition and Application of an Evaluation Technique

Dominik Bork, Christine Schrüffer, and Dimitris Karagiannis

University of Vienna, Faculty of Computer Science
Waehringer Street 29, 1090 Vienna, Austria
dominik.bork@univie.ac.at

**Abstract.** For correct utilization of a modeling language and comprehension of a conceptual model, the graphical representation, i.e., the notation, is of paramount importance. A graphical notation, especially for domain-specific languages, should be aligned to the knowledge, beliefs, and expectations of the intended model users. More concretely, the notation of a modeling language should support computational offloading for the human user by increasing perceptual processing (i.e., seeing) and reducing cognitive processing (i.e., thinking and understanding). Consequently, method engineers should design intuitively understandable notations. However, there is a lack of support in evaluating the intuitiveness of a notation. This paper proposes an empirical evaluation technique for bridging that research gap. The technique comprises three independent experiments: term association, notation association, and case study. Usefulness of the technique is shown by an exemplary evaluation of a business continuity management modeling language.

**Keywords:** Conceptual Modeling · Domain-specific Modeling · Modeling Language · Notation · Evaluation · Business Continuity Management.

## 1 Introduction

Due to their abstracting power, conceptual models are excellent in decreasing complexity of a system under study, thereby highlighting its relevant aspects for means of understanding and communication by human beings [21]. In order to achieve this ambitious goal, the demand for intuitively understandable graphical notations advances, consequently asking to fill a research gap of specialized design and evaluation techniques [9, 8]. This affects both, "standard" modeling languages (see [4, 5, 7]) and domain-specific modeling languages (DSMLs).

For efficient model-based communication, the notation plays an important role [6, 21] as it establishes the *"first contact of the users with the modeling language"* [7, p. 123] and a first precondition for its adoption and correct usage [4]. A notation should thus support the modeler in creating and the user in interpreting a model. An *intuitive* notation should moreover account for *computational offloading*, i.e., shifting some of the *cognitive tasks* to *perceptual tasks* [18] which ultimately leads to an intuitive understanding of a modeling language [17]. Intuitivity refers to *Semantic Transparency* as proposed in [18], i.e., the extent to which the graphical representation encodes the meaning of a modeling language concept. Intuitiveness is also referred to by *readability* - models are represented *"in a natural way and can be easily understood without the need for further explanations"* [2, p. 214], *pragmatic quality* - *"correspondence between the model and the audiences interpretation of the model"* [14, p. 94], or *understandability* - *"the ease with which the concepts and structures in the [..] model can be understood by the users of the model"* [20].

Evaluation of modeling languages is very subjective and difficult [16, 19, 11]. *"While the finished product (the software system) can be evaluated against the specification, a conceptual model can only be evaluated against people's (tacit) needs, desires and expectations"* [19, p. 245]. The difficulty further increases when focusing on intuitive understanding. We believe intuitive understanding can only be evaluated when the user's knowledge, beliefs, and aptitudes are known - a prerequisite for designing a DSML. Another open issue emerges when combining method chunks in situational method engineering [10] to select one or integrate existing notations. Consequently, our research question was: *"How to efficiently evaluate the intuitiveness of a domain-specific modeling language notation?"*

This paper builds upon the foundations of conceptual modeling and visualization (Section 2) and proceeds by proposing a new evaluation technique in Section 3. Section 4 then reports on an exemplary application of the technique. Eventually, Section 5 provides conclusions and directions for future research.

## 2 Foundations

**Domain-specific Conceptual Modeling.** A conceptual modeling method comprises [12]: A *modeling language*, a *modeling procedure*, and *mechanisms & algorithms*. The modeling language encompasses the language syntax, i.e., the grammar of the language; the language semantics, i.e., the meaning of the language concepts; and the language notation (also referred to as concrete syntax), i.e., the visual representation of the language. Based on the application, general-purpose modeling languages (GPMLs) like BPMN and UML can be differentiated from DSMLs as e.g., realized within the OMiLAB [3, 13]. Evaluating the intuitiveness of GPML notations is problematic because of the diverse stakeholders involved and their modeling purposes addressed with such languages. When designing a new DSML, on the other hand, evaluating intuitiveness becomes feasible because the potential users and their purposes of using the DSML are part of the design process [8]. Thus, DSML method engineers should respect domain-specificity not only in the syntax but also in an intuitive notation.

**Visual Aspects in Conceptual Modeling.** [22] developed a decoding theory considering humans as information processing entities. Information processing can be divided into: *Perceptual Processing (seeing)* which is fast and automatic, and *Cognitive Processing (understanding)* which is slow and resource-intensive. Diagrams aim for *computational offloading* by replacing some cognitive tasks by perceptual ones. The objective of designing cognitive effective notations thus needs to be to reduce cognitive processing. Similarly, [18, p. 761] states *"Designing cognitively effective visual notations can [..] be seen as a problem of optimizing them for processing by the human mind"*.

In conceptual modeling, an intuitive visual representation is vital for acceptance and adoption of the modeling method [7, p. 123]. *"The extent to which diagrams exploit perceptual processing largely explains differences in their effectiveness"* [18, p. 761] (see also [15, 23]). A comprehensive foundation for empirical research on conceptual modeling notations was proposed by Daniel Moody's impactful Physics of Notation [18]. Moody developed nine design principles for designing cognitive effective notations. The motivation for his research was that *"cognitive effectiveness of visual notations is one of the most widely held (and infrequently challenged) assumptions in the IT field. However, cognitive effectiveness is not an intrinsic property of visual representations but something that must be designed into them"* [18, p. 757].

**Semantic Transparency.** The semantic transparency design principle is defined as *"the extent to which the meaning of a symbol can be inferred from its appearance"* [18, p. 765]. In literature, semantic transparency is often considered synonymous to an intuitive understanding, i.e., novice users having no training on a modeling language are capable of intuitively deriving the meaning of the language elements from looking at their notation [18]. A notation with a high semantic transparency moves cognitive processing toward perceptual processing as users can infer the meaning of a symbol/model from their working and/or long term memory. Consequently, method engineers should design semantically transparent (mnemonic) visual notations.

## 3 An Evaluation Technique for Notation Intuitiveness

A new evaluation technique assessing the intuitiveness of modeling language notations is proposed in the following. The technique builds upon participatory design [24] while aiming to be efficiently customized and utilized by method engineers. The evaluation technique's core consists of three sequential phases, each of which conducting a specific experiment with participants. The core phases are preceded by an *initiation* and concluded by a *conclusion* phase (see Fig. 1).

**Initiation Phase.** Participants are briefly introduced to the domain and the building blocks of the modeling method to be evaluated. This primarily concerns the definition of the relevant domain terms and an introduction to the individual model types of the modeling method (if more than one model type is given). This introduction needs to be textually or orally, i.e., without showing any visual aspects like language concepts or sample models. Moreover, useful information for analyzing the results of the experiments like demographics and previous experience in modeling and the domain to be addressed is collected.

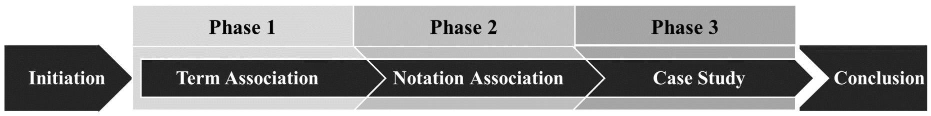


**Fig. 1.** Procedure of the evaluation technique

**Phase 1 – Term Association Experiment.** Participants are provided terms that refer to names of modeling language concepts. Each participant then individually drafts one or more graphical representations for each term he/she deems most intuitive. For this task, participants are provided blank papers that only list the terms and coloured pencils for the sketches. As a conductor of this experiment, one needs to classify the returned notation drafts into groups of similar graphics with respect to the *most frequent shapes and colors*. Comparing the gained drafts with the current modeling language notation might identify inadequacies and point to potential improvements.

**Phase 2 – Notation Association Experiment.** Participants are presented notations of the current modeling language. They are then asked to record their up to three intuitive associations that pop out when looking at the notations. It is important to note, that participants are only presented the notation without any hint of e.g., the name or the semantics of this concept. As a conductor of this experiment, one needs to classify all responses to measure the percentage of participants that intuitively associated the correct semantics to a provided notation. If one of the named terms of one participant matches with the true name or semantics of the concept, the notation is classified *identified*. For instance, if one of the named terms for a class 'Recovery activity' of one participant is 'recovery activity' or 'rollback activity', the notation is correctly identified. In the case that one of the named terms nearly fits the semantics, it is categorized as *partially identified*. In the example of a class 'Recovery activity', the terms 'task' or 'recovery measure' nearly fit to the true semantics. If none of the provided terms expresses the semantics, the notation is classified as *not identified*.

**Phase 3 – Case Study Experiment.** The case study should be as focused and short as necessary to test whether participants are able to intuitively combine the modeling language concepts in order to solve the presented case. It should be textually introduced and participants shall be provided a modeling tool if applicable. As a conductor of this experiment, one needs to classify the provided models according to their semantic and syntactic correctness. Three error categories are distinguished: *application error*, considers a wrong application of a concept or a wrong definition of a concept property; *procedural error*, covers a wrong sequence of concepts and a wrong/missing application of a relation; and *incomplete model*, covers missing concepts or missing properties of a concept.

**Concluding Phase.** The conductor presents the solution of the case study before the participants are asked to fill out a feedback survey. The survey covers the *Intuitivity of the notations* and optionally also the *Usability of the modeling tool* (not in scope of this paper). Eventually, participants are asked to provide positive and negative feedback, and improvement suggestions e.g., using post-its.

## 4 Application of the Evaluation Technique

This section describes an application of the technique to a DSML for business continuity management (BCM) which is under development in the scope of an international research project. BCM is defined as a *"holistic management process that is used to ensure that operations continue and that products and services are delivered at predefined levels"* [1]. It includes the identification of possible risks of regular business processes and of processes which handle the consequences of an occurred risk. The evaluation aimed to assess the intuitiveness of the graphical notation of the first version of the BCM modeling language.

In total, 15 information science Master students participated in the evaluation. Most participants are male (87%), between 25 and 29 years old, and are in the second semester of their Masters. The initiating survey showed, that the participants have solid experience with modeling and meta-modeling, fundamental experience with business process modeling, and no experience in risk management and business continuity management.

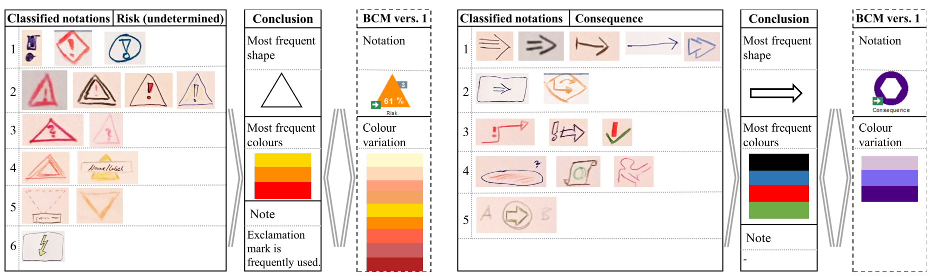


**Fig. 2.** Term Association experiment results for Risk (undetermined) and Consequence

**Results of Phase 1: Term Association Experiment** Participants were provided ten concepts of the BCM modeling language. Within ten minutes, they were asked to draft a notation for each term that they deemed most intuitive. Fig. 2 summarizes the classification of the results for the term *Risk (undetermined)* on the left side. The most frequent shape is a triangle and the most frequently used colours are red shades. Furthermore, it can be stated that exclamation marks are frequently used. By comparing these results with the notation realized in version 1 of BCM, it can be concluded that the notation is already intuitive.

Fig. 2 (right) summarizes the term association experiment results for the concept *Consequence*. It can be derived, that in most classes an arrow is used whereas the colours vary. By comparing these results it can be concluded, that shape and colours are different. Therefore, the notation in BCM version 1 is categorized as not intuitive, requiring a major revision for this concept.

Fig. 3 classifies excerpts of the results of the term association experiment using a traffic light system. The green light (left circle) indicates that the association is correct, the yellow light (middle circle) that the association is partially correct, and the red light (right circle) that the association is not correct. From the ten concepts tested in total, five associations were correct, four were partially correct, and one was not correct (the concept Consequence).

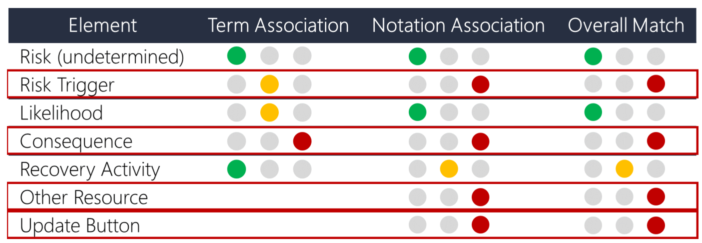


**Fig. 3.** Excerpt of term association and notation association experiments results

**Results of Phase 2 – Notation Association Experiment** Participants were given 15 notation samples of the first BCM modeling language version. They had ten minutes to write up to three most intuitive meanings they associate to a given notation. Fig. 3 (second column) classifies the gained insights again using the traffic light system. In total, eight concepts were correctly identified, four concepts partially identified, and four concepts were not identified, including the Consequence concept that already failed passing the term association experiment.

**Results of Phase 3 – Case Study Experiment** Participants were asked to create five BCM models. For ensuring the test is focusing intuitiveness, a time limit was set. Based on a pre-test with a novice modeler, we decided to give participants 30 minutes to create all five models. The analysis of the models resulted in the following observations: Most errors are *application errors* that are twice as many as *procedural errors* or issues of *incomplete models*. Twenty-two of the thirty-one errors are due to a wrong application of a concept which can be explained by the misunderstood notation of the Risk Trigger or the misunderstood relation between a Risk Trigger and a Risk. Interestingly, while the Consequence notation was not identified in the first two evaluation experiments, it was used correctly in every created model of all participants.

## 5 Lessons Learned, Implications and Conclusions

The concluding feedback session included a survey and a focus group discussion. Participants proposed to develop new gateways especially for the risk model of BCM which differ from BPMN gateways. Furthermore, it was mentioned that the allocation of the likelihood was not intuitive. Fig. 3 (right column) summarizes an excerpt of the results of the term and notation association experiments. If both experiments categorized a concept in the same colour of the traffic lights, the concept is overall also categorized with this colour. The risk trigger is categorized red since it was not identified ten times in the notation association experiment. The participants applied different colours by drawing the notation of a likelihood, but nevertheless, twelve of fifteen participants correctly identified the likelihood notation. Fig. 4 exemplifies how the experiments' led to more intuitive notations.

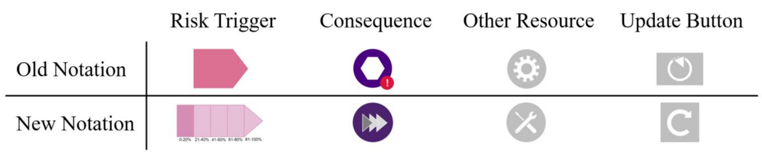


**Fig. 4.** Revised notations for four BCM modeling language concepts

By involving the participants in co-creating and evaluating the notation it was possible to improve the first version of the BCM modeling language with respect to its intuitive understanding. A limitation of this research is related to the generalizability of the findings. First, the participants were Master students and not the actual users in the domain. It can be assumed however, that domain experts would produce even better suggestions for improvement. A further limitation targets the limited number of participants (15) and the single application with one modeling language (BCM). However, even under these conditions, the technique proved utility and produced notation improvements.

The technique proposed in this paper targets the empirical evaluation of the intuitiveness of a modeling language notation. Strengths of the technique are its technology-independence and language-customizability enabling efficient adoption. In our future research we plan to apply the technique to further modeling languages and to develop a web-based evaluation environment which enables method engineers to efficiently set-up the experiments for their languages.

## Acknowledgment

This research has been partly funded through the Federal Ministry of Education, Science and Research (BMBWF) funded France / Austria Joint Scientific and Technological Cooperation program with the project number FR 01/2019.

## References

1. Plain english iso 22301 2012 business continuity definitions, `http://www.praxiom.com/iso-22301-definitions.htm`, last accessed: 2019-06-27
2. Batini, C., Ceri, S., Navathe, S.B., et al.: Conceptual database design: an Entity-relationship approach, vol. 116. Benjamin/Cummings Redwood City, CA (1992)
3. Bork, D., Buchmann, R.A., Karagiannis, D., Lee, M., Miron, E.T.: An Open Platform for Modeling Method Conceptualization: The OMiLAB Digital Ecosystem. Communications of the Association for Information Systems **44**, pp. 673–697 (2019)

4. Bork, D., Karagiannis, D., Pittl, B.: Systematic Analysis and Evaluation of Visual Conceptual Modeling Language Notations. In: 2018 12th International Conference on Research Challenges in Information Science (RCIS). pp. 1–11. IEEE (2018)
5. Bork, D., Karagiannis, D., Pittl, B.: A Survey of Modeling Language Specification Techniques. Information Systems p. in press (2019)
6. Caire, P., Genon, N., Heymans, P., Moody, D.L.: Visual notation design 2.0: Towards user comprehensible requirements engineering notations. In: 21st IEEE International Requirements Engineering Conference (RE). pp. 115–124. IEEE (2013)
7. El Kouhen, A., Gherbi, A., Dumoulin, C., Khendek, F.: On the semantic transparency of visual notations: experiments with uml. In: International SDL Forum. pp. 122–137. Springer (2015)
8. Frank, U.: Domain-specific modeling languages: requirements analysis and design guidelines. In: Domain Engineering, pp. 133–157. Springer (2013)
9. Gulden, J., van der Linden, D., Aysolmaz, B.: A research agenda on visualizations in information systems engineering. In: 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering. pp. 234–240 (2016)
10. Henderson-Sellers, B., Ralyté, J.: Situational method engineering: state-of-the-art review. Journal of Universal Computer Science (2010)
11. Izquierdo, J.L.C., Cabot, J.: Collaboro: a collaborative (meta) modeling tool. PeerJ Computer Science **2**, e84 (2016)
12. Karagiannis, D., Kühn, H.: Metamodelling Platforms. In: E-Commerce and Web Technologies, Third International Conference, EC-Web 2002. p. 182 (2002)
13. Karagiannis, D., Mayr, H.C., Mylopoulos, J. (eds.): Domain-Specific Conceptual Modeling, Concepts, Methods and Tools. Springer (2016)
14. Krogstie, J., Sindre, G., Jørgensen, H.: Process models representing knowledge for action: a revised quality framework. European Journal of Information Systems **15**(1), 91–102 (2006)
15. Larkin, J.H., Simon, H.A.: Why a diagram is (sometimes) worth ten thousand words. Cognitive science **11**(1), 65–100 (1987)
16. Lindland, O.I., Sindre, G., Solvberg, A.: Understanding quality in conceptual modeling. IEEE software **11**(2), 42–49 (1994)
17. Michael, J., Mayr, H.C.: Intuitive understanding of a modeling language. In: Australasian Computer Science Week Multiconference. p. 35. ACM (2017)
18. Moody, D.: The 'physics' of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Transactions on Software Engineering **35**(6), 756–779 (2009)
19. Moody, D.L.: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. Data & Knowledge Engineering **55**(3), 243–276 (2005)
20. Moody, D.L., Shanks, G.G.: What makes a good data model? evaluating the quality of entity relationship models. In: International Conference on Conceptual Modeling. pp. 94–111. Springer (1994)
21. Mylopoulos, J.: Conceptual modelling and Telos. Conceptual Modelling, Databases, and CASE: an Integrated View of Information System Development, New York: John Wiley & Sons pp. 49–68 (1992)
22. Newell, A., Simon, H.A.: Human problem solving, vol. 104. Prentice-Hall Englewood Cliffs, NJ (1972)
23. Petre, M.: Why looking isn't always seeing: readership skills and graphical programming. Communications of the ACM **38**(6), 33–44 (1995)
24. Recker, J.: Opportunities and constraints: the current struggle with BPMN. Business Process Management Journal **16**(1), 181–201 (2010)