

# On the Requirement from Practice for Meaningful Variability in Visual Notation

Dirk van der Linden<sup>(✉)</sup>, Irit Hadar, and Anna Zamansky

Department of Information Systems, University of Haifa, Haifa, Israel  
{djtlingen, hadari, annazam}@is.haifa.ac.il

**Abstract.** This research-in-progress paper proposes the need for a move towards more *meaningful variability* of visual notations. Evidence accumulated via an online survey on the requirements practitioners have for visual notations, indicate the need for variability of a modeling language's visual notation. Widely used modeling languages in practice such as UML and BPMN do not support redesign of the visual notation of core constructs without modifying or extending the underlying abstract syntax and semantics (e.g., UML stereotypes, BPMN extensions). The expressed need to vary commonly used visual notations depending on particular users or contexts, while not changing the underlying modeling language itself, poses a set of research challenges discussed here.

**Keywords:** Conceptual modeling · Requirements · Visual notations · Variability · Concrete syntax · Practitioner · Empirical study

## 1 Introduction

Visual notation, or concrete syntax, is the main way people interface with models. Whether models convey their meaning effectively and accurately depends on their visual design [14], as well as the cognitive makeup of the person looking at them [24]. Consequently, a model used among developers or technical experts would have different requirements towards its visual representation being effective than a model used to communicate with other stakeholders such as users, business experts and domain experts.

This paper will show that many practitioners who employ conceptual modeling techniques require the ability to vary the visual representation (i.e., the concrete syntax) of their models depending on the audience they target. This does not mean they are interested in modifying the language itself, whether by extending the meta-model or creating specialized extensions. In particular, when developers communicate with customers or business stakeholders they prefer using less technical or complicated displays for the same model, such as using rich pictures or clear iconography.

This implies the need for a certain *variability* in the way models are represented: changing the way the visual notation appears. However, at the same time practitioners use general-purpose languages which remain visually abstract, and

offer little to no support for on-the-fly variation of their visual notation. Short of creating specialized notations, e.g., by stereotyping in UML or extension creation in BPMN, there is no easy way to vary only the visual representation of the notation.

This paper is focused on understanding how we can give practitioners what they want. In particular, how the visual notation of a single modeling language can accommodate a diverse audience of model users, including non-experts with no experience with modeling. Based on an ongoing empirical study into the requirements practitioners have for visual notations, we describe the need for a move towards a *meaningful variability* of visual notation use. We discuss the challenges stemming from the requirement to change parameters of a visual notation on-the-fly to suit different audiences, in particular when these changes to accommodate non-experts are non-trivial and require rethinking existing models.

## 2 Defining Variability of Concrete Syntax

Variability in the context of modeling languages has received attention in literature, such as the need for systematic ways to create dialects of enterprise modeling languages [3]. However, such work remains primarily on the level of meta-models describing which entities exist, namely the abstract syntax. To define or describe a modeling language fully this is not sufficient, as both semantics (what things mean) and the concrete syntax, or visual notation (how things look), are important [8]. Meta-modeling approaches grounded in the OMG Meta-Object-Facility (MOF) [20] have been proposed to extend the degree to which visual notations are systematically captured and linked to their meta-models [6, 17]. Related, approaches to detect inconsistencies between such definitions of visual notations and meta-models have been proposed [1]. Some of this work has explicitly noted the option of having multiple visual notations for a single meta-model [8], concluding that multiple visual notations can be used as long as the underlying meta-model is well defined and serves as a common (abstract) representation of the actual information represented in the model.

However, these studies predate insights from more recent theory on visual notation design [14], which shows diagram-level aspects of design known to be important for ensuring that non-experts can parse models effectively. In particular, it is now understood that *meaningful* variations in concrete syntax to bridge the expert/non-expert gap amount to more than mere differences in symbols or color schemes used. Some examples of such variation are [14]:

- Targeted iconographic design to suggest meaning: non-experts are aided by the use of rich pictures that suggest their underlying concepts' meaning clearly.
- Use of visual complexity management mechanisms: non-experts may find it hard to parse models that do not incorporate any mechanisms to abstract and hide information, having to mentally 'chunk' elements into sub-diagrams.
- Variation in the number of visual variables used to discriminate between visual elements: non-experts may benefit from graphical symbols being distinguished on more than just shape or color.

- Variation in size of the visual vocabulary: non-experts are challenged by notations with a high number of distinct graphical symbols.

These aspects of visual notation design are not addressed by the earlier work mentioned, and pose additional challenges for the ease with which a visual notation may be selected and used on-the-fly, as will be further discussed in Sect. 5.3.

### 3 Empirical Study

The original objective of the empirical study was to elicit data on the requirements that practitioners hold for visual notations, and analyze how these compare to the Physics of Notations (PoN) [14], a widely used design theory for cognitively effective visual notations. We then aimed to determine whether some requirements are considered more important than others, depending on professional context.

While the assessment of the relative importance of the PoN requirements was done in a quantitative fashion, We took a qualitative approach [18] to explore practitioners' views on what is important in visual notations. Specifically, we used exploratory coding [22] to analyze the views elicited from practitioners.

#### 3.1 Research Questions

Our general research question is: *what requirements do practitioners have for visual notations?* For a satisfactory answer to this question, we need not only to understand what requirements practitioners might find important, but also to what degree these requirements are found important, which ones are perceived to be the least addressed, and so on. To this end, we concretely investigate:

1. What requirements do practitioners have for visual notations?
2. To what extent does the existing theory for visual notation design adequately cover practitioners' requirements?
3. Which requirements, if any, are considered more high priority than others? Can we correlate this to a domain or focus of modeling?
4. To what extent do existing modeling languages satisfy these requirements? How can this be improved?

#### 3.2 Research Design

**Participants.** We used LinkedIn to approach practitioners employing conceptual modeling techniques. In particular, we solicited participation in the study via relevant professional groups. We searched first for groups based on keywords including “conceptual modeling”, “requirements”, “business analyst”, “software architect/engineer”, “enterprise architect/engineer”. After joining a group, we posted a message asking group members for their input on requirements for visual notations.

**Materials.** To assess requirements for visual notations we used the summary of the PoN principles [14] given in [15]. For each principle we displayed the summary with a 5-point Likert scale ranging from “very important” to “not important at all.” Although these results are not discussed in this paper, we mention them as they are used to contextualize the qualitative questions on any additional requirements practitioners may have.

**Pilot.** Before setting out the survey on LinkedIn, we piloted an initial version among four professionals with expertise in conceptual modeling techniques. Their feedback was used to verify the estimated time needed to complete the survey (attempting to keep it short in order to stimulate participation), and remove any potential misunderstandings in the phrasing. Two participants in the pilot indicated that they would have different answers for how important each requirement was depending on whether they interacted with experts or non-experts. As a result of this feedback we adapted the survey to be of two parts, investigating the importance of requirements for visual notations used – first among experts, and second among non-experts such as business professionals. This version was piloted again with the same group, after which no more ambiguities were found.

**Procedure.** We invited people to participate in the survey voluntarily, with no incentive given. The questionnaire first elicited some general demographic data, as well as some information about their modeling experience asking:

- What country do you work in?
- How many people are employed in your organization? [Less than 100, Less than 1000, Less than 10.000, More than 10.000]
- How many years have you used modeling languages in a professional setting? [Less than 5 years, 5 to 10 years, More than 10 years]

This was followed by specific demographics, asking:

- What do you mostly model? [Processes, Goals/Motivations, Information/Data, Requirements, Architecture (Software), Architecture (Enterprise), Other: ...]
- What is the typical purpose of your models? [...]
- What modeling language(s) do you have significant experience with? [...]
- What domain do you currently work in? [Services, Manufacturing, Telecom, Financial, Health, Government, Academic, IT/Software, Other: ...]

Next, participants were presented the one-line summaries of each PoN principle and asked to rate them on a 5-Point Likert scale. This was done twice, first asking participants to consider their requirements for visual notations used among fellow modelers, followed by the question:

- Are there any requirements you feel are not covered by the ones you just saw, specific to the use of a visual notation among fellow modeling experts? And then, considering their requirements for non-modeler experts, followed by the question:

- Are there any requirements you feel are not covered by the ones you just saw, specific to the use of a visual notation among other stakeholders with no expertise in modeling?

### 3.3 Data Analysis

The data described in this paper is based on the first **85** responses received. One response was discarded on suspicion of non-serious data, with all Likert Scale data repeating the same value, and any open question containing only meaningless repeating characters. The data we analyze in this paper results from the open questions, yielding qualitative data. We used a qualitative approach for coding this data, using iterative coding in which all three authors coded the data independently. Coding of the data describing the purpose of modeling involved two iterations of coding, after which we agreed on a limited set of codes that arose similarly out of our analyses. For the data on missing requirements we separately encoded whether one of the PoN principles addressed the presented requirement and/or whether the requirement was instead related to a different factor, such as tool support or semantic quality instead of the visual notation. After discussion these coding sessions led to the results shown in Tables 1 and 2.

### 3.4 Threats to Validity

The primary threats to validity in this study are construct validity and participant fit. In the survey, requirements are presented as the one-sentence summary given by the PoN itself. It is possible that they would be interpreted differently than intended, however, given the ambiguous nature of the PoN itself [11], even if given full details of the principles as presented in [14] such differences in understanding could arise. Nonetheless, these high-level descriptions represent the summarized overall ‘spirit’ of the principle, and are widely used by different applications of the PoN. We therefore work under the assumption they serve as an adequate representation of the principles.

Whether participants actually know if, and to what degree, these requirements are important for them is another matter. We targeted participants with experience in conceptual modeling, specifically capturing their years of professional experience, in order to have a grounded assumption that they had been exposed to the use of modeling languages long enough to develop an internal set of high-priority requirements. In an analog to Henry Ford’s quote on the development of the first cars, “if I asked anyone at the time what they wanted, they would have said faster horses,” we do acknowledge that the importance of these requirements might not be understood or underwritten by all participants.

The fit of participants to the study was ensured as much as possible by limiting the participant recruitment to relevant groups of LinkedIn, our own professional network, and academic mailing lists, in order to target only those with experience in conceptual modeling. The profile built by the questions given above further helped to (de)select only those participants with relevant and significant experience. Furthermore, we specifically targeted those with primary

industrial experience, and specifically, ensured in the datasets that there were no participants whose primary experience was solely of an academic nature.

A final threat to validity is the potential self-selection bias, as we only elicit responses from those practitioners willing to respond. However, in our experience setting out these surveys on LinkedIn, we encountered several groups where one or more participants enthusiastically replied to the survey and stimulated others to join, emphasizing the potential benefit of the insight the study could also bring for their community.

## 4 Study Results

### 4.1 Demographic Data

While the dataset at this point in time cannot yet be considered to be significant enough to generalize (given  $n = 84$ ), we established some basic demographic data to ensure that the data represents a heterogeneous sample of participants (see Fig. 1).

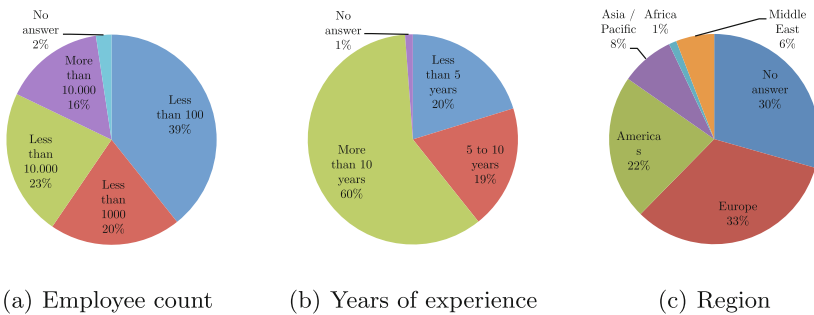


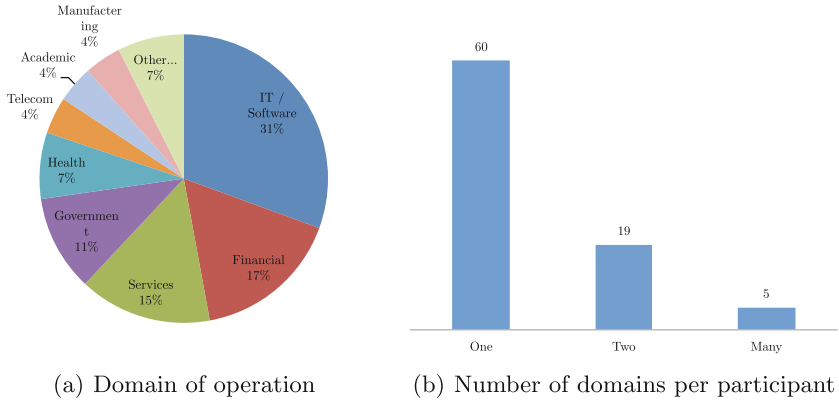
Fig. 1. General demographics.

For modeling-specific demographics we list the domains in which participants operated. Practitioners typically only operated in a **single** domain (see Fig. 2).

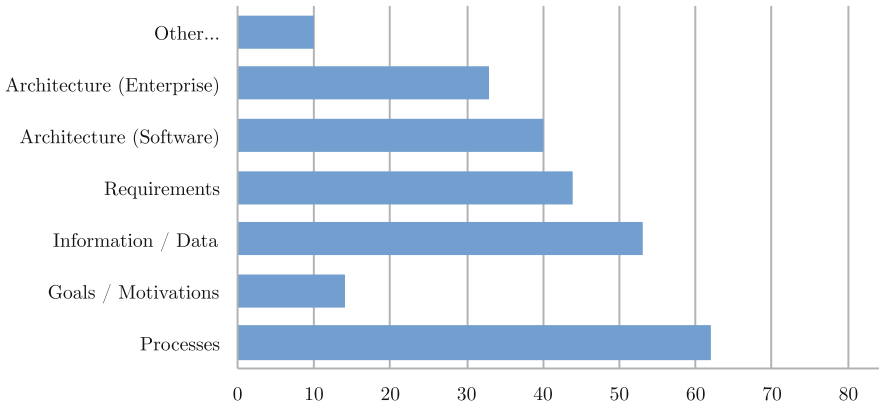
As for what is actually modeled: most participants were involved in modeling multiple foci, the median being **three**. Fig. 3 shows how many participants modeled each focus.

### 4.2 Used Visual Notations

A large number of modeling languages (**36**) was reported to be used by the respondents: BPMN, UML, xtUML, SysML, SimuLink, Stateflow, SDL, MARTE, ERD, ORM, BPEL, FSA, ArchiMate, IDEF0, IDEF1x, CMMN, Viso, RDF/OWL, GRAPES BPM, i\*, IE, DMN, MODAF, GSN, EPC, C4, and more.



**Fig. 2.** Domains in which participants operate.



**Fig. 3.** Different foci modeled by participants. The x-axis is number of participants, running up to the  $n$  of 84.

However, most of these notations were reported to be used by very few participants. By far the most used visual notations reported on were UML (**49** counts) and BPMN (**32** counts), followed at a big distance by ArchiMate (**12** counts) and SysML (**8** counts).

Of particular interest is that the two most frequently used visual notations are general-purpose languages, seemingly wielded and adapted to many domains and purposes. Interestingly, dataflow diagrams (DFDs) and entity-relationship diagrams (ERDs) were noted only **once** and **four** times, respectively. Compared to a study from 2006 which found DFDs and ERDs [4] among the most used notations in practice, this could point towards changing attitudes.

As noted, there are limitations to keep in mind when considering these data, most notably self-reporting bias and selection bias. However, given the wide spread of LinkedIn groups targeted and different domains reached, we believe

that even in these preliminary results a tendency of practice can be seen regarding the use of general-purpose languages and the eschewing, from their perspective, of more esoteric notations.

### 4.3 Purpose of Modeling Efforts

Of the **77** responses received for the question regarding the purpose of modeling effort, we discarded one for being irrelevant (stating solely “UML”). To code these responses, we first looked to use an existing set of codes for purposes of conceptual modeling, such as used in two widely-cited papers on the practice conceptual modeling [4, 5]. However, the list of given purposes did not include communication, commonly seen as the core purpose of conceptual modeling [21], nor was their origin (e.g., literature, resulting from coding) discussed.

Thus, we decided to have each author code all **76** responses independently. These responses encompassed **106** distinct purposes described by participants. Following comparison of the codes that arose in the initial coding process, the set of codes presented in Table 1 was agreed upon.

**Table 1.** Coded purpose of modeling efforts

Purpose	#	Exemplary quote(s)
Communicating	25	“Common point of reference for requirements discussions”
		“Make the modelled system clear and understandable to various stakeholders”
		“Bridging communication gaps across diverse groups of stakeholders”
Designing	23	“Designing new software”
		“Systems design”
Understanding	20	“Simplification of complex concepts/solutions”
		“High level understanding of the system and purpose”
Supporting development	13	“Guide me when actually writing the software”
		“Supporting decisioning, and instructing designers”
Visually representing	13	“Visualisation of architectural metadata”
		“Visualizing design”
Engineering requirements	12	“Represent knowledge at different levels of abstraction to look for missing, incorrect, and unnecessary requirements”



#### 4.4 Additional or Missing Requirements

We processed the data elicited on missing requirements with a coding schema with the three researchers independently coding the data marking whether, and if so, which PoN principle addressed the proposed requirements. A total of **49** remarks were coded: **six** remarks were irrelevant to the posed question; **five** remarks were excluded due to their ambiguous nature; and, **eleven** remarks were not related to the principles, reflecting requirements related to a tool rather than the visual notation itself. Of the remaining **27** remarks, only two dealt with something not clearly or directly addressed by a PoN principle: how to display visually overlapping relationships. The remaining **25** remarks were found to be addressed by one or more PoN principles, as summarized in Table 2 below.

**Table 2.** Requirements covered by each principle.

PoN principle	#	Exemplary quote(s)
Cognitive fit	8	“I cannot do the formal models without ‘artist impressions’ or rich pictures tailored to specific stakeholders or stakeholder groups, even fellow modeling insiders/experts”
		“My responses are coloured by my desire to use these diagrams to collaborate with non - experts, those most familiar with the problem domain”
		“Highlight how important is to have flexibility to communicate to several audiences perhaps incorporating a more complex visual design. The simplicity of the visual design of UML could be perfect for a software engineer but very cold for a Business User”
		“Flexibility in presentation”
Complexity management	6	“Visual representation capabilities like zooming in or out”
		“Visual simplification techniques”
		“Provide different views of complexity level”
		“Use of abstraction (e.g. a high-level overview)”
Semantic transparency	5	“I think [the] biggest detractor to existing [notations] are that they are conceptually abstract and have steep learning curves”
Semiotic clarity	3	“Precision and unambiguous”
Dual coding	1	“Visual notation needs to have a textual counterpart”
Perceptual discriminability	1	“The size and usability of the symbols”
All (PoN overall)	2	“The modeling notation should be empirically founded on cognitive theories of visualisation”

The responses in Table 2 show a link to the aspects in which non-experts are perceived by our participants as more cognitively challenged during model

usage, such as the notion of personalizing the notation for different audiences, and ensuring that the used visual representation be as simple as possible. In an earlier study the first author performed on model-aided decision making in Enterprise Architecture [10], numerous responses were found that corroborate this tendency to require simplicity when dealing with modeling non-experts. For example, one architect noted that PowerPoint, Excel and Visio were more suitable for non-technical audiences, and another architect noted that in dialogues with management stakeholders, they did not use any modeling languages or techniques.

## 5 Toward Meeting Practitioners' Requirement for Visual Variability

### 5.1 The Need

The presented results sketch a clear view on the research question *what requirements practitioners have towards visual notations*, as well as showing that *existing theory for visual notation design adequately covers practitioners' requirements*. What seems particularly salient in Table 2 is practitioners' requirements to support the purpose of models in communicating with non-experts, and bridging the cognitive gap between modeling experts and non-experts. This means, specifically, that a meaningful visual variability is indeed required: varying the (properties of the) visual notation depending on the audience. However, this need seems to not yet be accommodated by the main modeling languages that are used in practice: UML and BPMN.

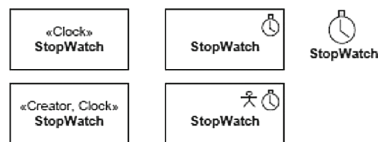


Fig. 4. Example of a stereotyped entity from the UML standard [19].

### 5.2 Why This Need Is Not Accommodated

To concretely answer the research question *To what extent does the existing theory for visual notation design adequately cover practitioners' requirements?*, let us look at the two most used modeling languages among our participants. UML allows a designer to adapt the notation to a specific context by using stereotyping, which enables both the use of *specific terminology*, and *[visual] notation* [19, Sect. 12.3.3.4]. The extent to which a new notation can be introduced is limited though, to primarily new symbols and coloring. Stereotyped entities can have symbols appended to them as markers, or be displayed as that symbol entirely, as shown in Fig. 4.

This allows at least for the use of rich pictures: the use of detailed iconographic representation for domain concepts. However, there is a significant limitation in that these visual modifications only seem to be allowed over stereotyped elements. This means that new elements in the abstract syntax have to be created, and semantics defined, instead of allowing for simple visual variability in the representation. The existence of numerous tool-specific extensions to allow for modification and coloring of core elements (e.g., in Visual Studio) seems to be a clear hint at people implementing this need themselves. Similar to UML, BPMN extensions' primary means of visual modification in practice seems to be coloring and the addition of markers to existing graphical elements [12]. There are concrete instructions in the standard for BPMN [20] when it comes to extending its notation. Particularly salient are:

- “A new shape representing a kind of Artifact MAY be added to a Diagram, but the new Artifact shape SHALL NOT conflict with the shape specified for any other BPMN element or marker.”
- “An extension SHALL NOT change the specified shape of a defined graphical element or marker (e.g., changing a square into a triangle, or changing rounded corners into squared corners, etc.).”

The same restriction as in the UML standard is found again: that existing elements may not be meaningfully changed. Shape, color, and line style of existing core constructs are all protected. This impacts the ability to create a meaningful variability in the visual notation, as properties of the core constructs would be modified to deal with practitioners' needs. An argument may be brought that allowing to make changes to core constructs' representation would impact the *mutual intelligibility* of created models. However, as practitioners clearly indicate such variability would be used to communicate from an expert audience (e.g., developer, technical analysts) to a non-expert audience (e.g., business stakeholders, management, end-users), there is no need for these two groups to read the same underlying model in the visual representation optimal for the other group. Therefore, the challenge of mutual intelligibility does not come into play.

The ability to create meaningful visual variability in UML and BPMN thus lags behind the ability to create meaningful semantic variability.

### 5.3 The Challenges

Finally, we reflect on *what can be done*. Some aspects required to implement variability in the concrete syntax to accommodate the expert/non-expert divide are seemingly trivial, although they would require incorporation into the relevant modeling language's standard to be truly effective. However, other aspects require more careful thought.

The aspects needed for *meaningful variability* to target non-experts as described in Sect. 2 lead to a number of challenges for the implementation and use of visual variability. In particular, the last three require thought on how to redraw models: when adding or removing complexity management mechanisms, when changing the number of visual variables used to discriminate between symbols, and when changing the total number of graphical symbols used.

**Challenge 1: Complexity Mechanisms.** It is known that non-experts are more challenged by visual complexity on models due to a lack of “chunking strategies.” [2]. This means that non-experts find it more challenging to mentally group together closely related entities and effectively perceive them as sub-diagrams. When changing the visual notation for a non-expert audience, one would thus have to incorporate such mechanisms. However, how do we decide which parts of the model to group together and hide without expert (modeler) oversight? The meta-model has to enforce encoding of meta-data that represents whether, and to which, sub-diagram an element may be collapsed. Determining the boundaries for this is challenging, as over-zealous encoding of sub-diagram potential may lead to diagrams with little actual information, with all meaningful semantic elements hidden away in sub-diagrams. Furthermore, the “chunks” of a model should ideally also be represented in a rich visual way. This poses an additional challenge for the iconographic design of these chunks, as the chunked sub-diagram not only requires complicated iconography (e.g., representing in a realistic way the chunked concept of “financial handling”, which is composed of say “payment request,” “payment receiving,” “payment registration,” “payment reminder”) but also has to be relatable to the *rich pictures* used for all the underlying elements when the sub-diagram is unfolded into all of its constituent parts.

**Challenge 2: Variation in Visual Variables.** To assist non-experts in clearly discriminating between different elements in a model, we can use visual variables. However, the more visual variables that are used, the more cognitively complicated it will be to assess which elements are distinct. If only shape is used to distinguish between different elements, most people would see so quickly. However, with the complexity of realistic models, more variables are needed to distinguish between all possible different elements, going up as high as using e.g., the unique combination shape, color, texture and size to determine an element’s uniqueness. How can we redraw a model for non-experts, requiring using fewer combinations of visual variables as a unique separation? This requires the incorporation of evidence-driven data, showing exactly which visual variables are most distinguishable, and then in particular which instantiations thereof, as recently proposed for e.g., color schemes [23]. However, for other visual variables such data is needed too, showing e.g., optimal schemes of most distinct textures, shapes, orientations.

**Challenge 3: Variation in Visual Vocabulary Size.** Visual notations use many different graphical symbols. While experts may be trained to deal with this complexity, for non-experts not trained or familiar with the use of such models, limiting the size of the visual vocabulary is important. Practically, this means not exceeding the established threshold of  $7 \pm 2$  distinct symbols [13]. However, the two most used notations exceed this by far, with UML going up as high as 60 distinct symbols for some diagrams [16], and BPMN as high as 171 [7]. When presenting a model to a non-expert, this requires one to thus either use a very limited subset of the notation, as in the case of BPMN’s “core”

constructs, or dynamically concatenating semantic elements to be represented by a similar visual element. When we have a model using, say, 35 distinct elements, and we want to apply a visual notation for non-experts that reduces that down to seven, how do we decide which semantic elements to represent by the same visual element? This, again, requires either extensive additional meta-data, grounded in e.g., ontological or psycholinguistic work establishing the similarity or “closeness” of these concepts [9] – or expert (modeler) oversight establishing clear rationale for the used grouping.

## 6 Concluding Outlook

In this research-in-progress paper we have described initial results from an ongoing study into the requirements practitioners have for visual notations, clearly showing the need from practice for meaningful variability in visual notations. In particular, variability in aspects that make diagrams easier to read for non-experts. Allowing for such variability poses some challenges to on-the-fly re-representing models, often demanding either expert (modeler) oversight and clear rationale, or grounding in additional meta-data of the modeled elements.

The contribution of this work so far is in its empirical findings providing insight into requirements from modeling practice, and how those may clash with prevailing research efforts. For example, the findings diverge from previous studies in the past decade [4], pointing towards changing attitudes in what modeling languages are most commonly used. Furthermore, it re-emphasizes the lack of widespread acceptance of specialized, niche notations for specific foci, showing that practitioners commonly use UML and/or BPMN instead of domain-specific notations. Perhaps as a consequence, participants stressed the yet unsatisfied requirement for visual notations to allow for variability in its visual notation. In particular, for design that allows for models to be effectively used with non-experts (i.e., end-users, business stakeholders).

Our further work in this area will center around implementing this kind of *meaningful variability* in the visual notation of general-purpose modeling languages. To do so, we will focus on (i) systematic formulation of visual notation dialects that account for design aspects important for non-expert understanding, (ii) mechanisms to allow OMG specifications to use multiple visual notations linked to one core meta-model (or abstract syntax), and finally (iii) an evidence-driven approach for systematically capturing (structures for) meta-data in the meta-model of a modeling language that can inform the on-the-fly rendering into varying visual notations.

## References

1. Baar, T.: Correctly defined concrete syntax for visual modeling languages. In: Nierstrasz, O., Whittle, J., Harel, D., Reggio, G. (eds.) MODELS 2006. LNCS, vol. 4199, pp. 111–125. Springer, Heidelberg (2006). doi:[10.1007/11880240\\_9](https://doi.org/10.1007/11880240_9)

2. Blankenship, J., Dansereau, D.F.: The effect of animated node-link displays on information recall. *J. Exp. Educ.* **68**(4), 293–308 (2000)
3. Braun, R., Esswein, W.: Designing dialects of enterprise modeling languages with the profiling technique. In: 2015 IEEE 19th International Conference Enterprise Distributed Object Computing (EDOC), pp. 60–67. IEEE (2015)
4. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? *Data Knowl. Eng.* **58**(3), 358–380 (2006)
5. Fettke, P.: How conceptual modeling is used. *Commun. Assoc. Inf. Syst.* **25**(1), 43 (2009)
6. Fondement, F., Baar, T.: Making metamodels aware of concrete syntax. In: Hartman, A., Kreische, D. (eds.) *ECMDA-FA 2005*. LNCS, vol. 3748, pp. 190–204. Springer, Heidelberg (2005). doi:[10.1007/11581741\\_15](https://doi.org/10.1007/11581741_15)
7. Genon, N., Heymans, P., Amyot, D.: Analysing the cognitive effectiveness of the BPMN 2.0 visual notation. In: Malloy, B., Staab, S., van den Brand, M. (eds.) *SLE 2010*. LNCS, vol. 6563, pp. 377–396. Springer, Heidelberg (2011)
8. Kleppe, A.: A language description is more than a metamodel. In: *Fourth International Workshop on Software Language Engineering (ATEM 2007)* (2007)
9. van der Linden, D., Proper, H.A.: Category structure of language types common to conceptual modeling languages. In: Bider, I., Gaaloul, K., Krogstie, J., Nurcan, S., Proper, H.A., Schmidt, R., Soffer, P. (eds.) *BPMDS/EMMSAD -2014*. LNBIP, vol. 175, pp. 317–331. Springer, Heidelberg (2014). doi:[10.1007/978-3-662-43745-2\\_22](https://doi.org/10.1007/978-3-662-43745-2_22)
10. van der Linden, D., Van Zee, M.: Insights from a study on decision making in enterprise architecture. In: *PoEM (Short Papers)*, vol. 1497, pp. 21–30 (2015)
11. van der Linden, D., Zamansky, A., Hadar, I.: How cognitively effective is a visual notation? On the inherent difficulty of operationalizing the physics of notations. In: Schmidt, R., Guédria, W., Bider, I., Guerreiro, S. (eds.) *BPMDS/EMMSAD -2016*. LNBIP, vol. 248, pp. 448–462. Springer, Cham (2016). doi:[10.1007/978-3-319-39429-9\\_28](https://doi.org/10.1007/978-3-319-39429-9_28)
12. van der Linden, D., Zamansky, A., Hadar, I.: On the need for more requirements towards visual notation design of BPMN extensions. In: *Caise Radar 2016*, University of Ljubljana (2016)
13. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychol. Rev.* **63**(2), 81 (1956)
14. Moody, D.L.: The “physics” of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Trans. Softw. Eng.* **35**(6), 756–779 (2009). <GotoISI>://WOS:000272172800003
15. Moody, D.L., Heymans, P., Matuleviius, R.: Visual syntax does matter: improving the cognitive effectiveness of the i\* visual notation. *Requirements Eng.* **15**(2), 141–175 (2010)
16. Moody, D., van Hilleegersberg, J.: Evaluating the visual syntax of UML: an analysis of the cognitive effectiveness of the UML family of diagrams. In: Gašević, D., Lämmel, R., Wyk, E. (eds.) *SLE 2008*. LNCS, vol. 5452, pp. 16–34. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-00434-6\\_3](https://doi.org/10.1007/978-3-642-00434-6_3)
17. Muller, P.-A., Fleurey, F., Fondement, F., Hassenforder, M., Schneckenburger, R., Gérard, S., Jézéquel, J.-M.: Model-driven analysis and synthesis of concrete syntax. In: Nierstrasz, O., Whittle, J., Harel, D., Reggio, G. (eds.) *MODELS 2006*. LNCS, vol. 4199, pp. 98–110. Springer, Heidelberg (2006). doi:[10.1007/11880240\\_8](https://doi.org/10.1007/11880240_8)
18. Myers, M.D.: Qualitative research in information systems. *Manage. Inf. Syst. Q.* **21**(2), 241–242 (1997)
19. (OMG) O.M.G.: Unified modeling language (UML), version 2.5 (2015). [www.omg.org/spec/UML/2.5/](http://www.omg.org/spec/UML/2.5/)

20. (OMG) O.M.G.: Meta object facility (MOF) version 2.5.1 (2016). <http://www.omg.org/spec/MOF/2.5.1/>
21. Robinson, S., Arbez, G., Birta, L.G., Tolk, A., Wagner, G.: Conceptual modeling: definition, purpose and benefits. In: Proceedings of the 2015 Winter Simulation Conference, pp. 2812–2826. IEEE Press (2015)
22. Saldana, J.: The Coding Manual for Qualitative Researchers. Sage, London (2015)
23. Stark, J., Braun, R., Esswein, W.: Systemizing colour for conceptual modeling (2017)
24. Wilmont, I., Hengeveld, S., Barendsen, E., Hoppenbrouwers, S.: Cognitive mechanisms of conceptual modelling. In: Ng, W., Storey, V.C., Trujillo, J.C. (eds.) ER 2013. LNCS, vol. 8217, pp. 74–87. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-41924-9\\_7](https://doi.org/10.1007/978-3-642-41924-9_7)