How Cognitively Effective is a Visual Notation? On the Inherent Difficulty of Operationalizing the Physics of Notations

Dirk van der Linden, Anna Zamansky, and Irit Hadar

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

Abstract. The Physics of Notations [9] (PoN) is a design theory presenting nine principles that can be used to evaluate and improve the cognitive effectiveness of a visual notation. The PoN has been used to analyze existing standard visual notations (such as BPMN, UML, etc.), and is commonly used for evaluating newly introduced visual notations and their extensions. However, due to the rather vague and abstract formulation of the PoN's principles, they have received different interpretations in their operationalization. To address this problem, there have been attempts to formalize the principles, however only a very limited number of principles was covered. This research-in-progress paper aims to better understand the difficulties inherent in operationalizing the PoN, and better separate aspects of PoN, which can potentially be formulated in mathematical terms from those grounded in user-specific considerations.

Keywords: visual notations, cognitive effectiveness, physics of notations, operationalization

1 Introduction

Conceptual modeling is a widely used technique in software engineering and information systems development to capture and reason about a particular domain of interest. Visual notations used in such modeling tasks have often been designed without eliciting and considering empirical evidence for what fits best the potential users and the task at hand. Some of the most widespread visual notations used in practice, such as UML, are affected by this limitation (cf. [8]). Some work has attempted to alleviate this by more explicitly tracing design to its rationale (cf. [14,20]), but such work remains on the level of the domain, not the notation itself.

The main issue with visual notations developed in this way is a lack of focused attention on ensuring their cognitive effectiveness, namely the ease with which people can read and understand diagrams written in the notation. Given that visual languages are often used for their convenience over textual languages, they should be designed and analyzed "from the perspective of languages that are cognitively usable and useful." [12]

Over the years, several frameworks have been proposed (e.g., Cognitive Dimensions [3], SEQUEL [7], GoM [15]) that, at least partially, paid attention to this aspect and provided notation designers with guidelines on how to improve the quality of visual notations. Recently, one such framework focusing exclusively on the cognitive effectiveness of visual notations, the Physics of Notations (PoN) [9], has become relatively widespread. Its adoption by researchers is evident by the ever growing number of analyses using it [18], including having been applied to e.g., BPMN, UML, i*, WebML, as well as the increase in the number of works citing it over other frameworks [2].

Moody positions the PoN's nine principles as constituting a type V prescriptive theory in terms of Gregor's [4] taxonomy of theory in IS [9, p. 775]. He states that these principles "can be used to evaluate, compare, and improve existing visual notations as well as to construct new ones". This effectively means that instead of considering endless possibilities when coming up with a new visual notation, one may opt for those possibilities which best comply with PoN. We refer to the activity of checking the compliance of a visual notation with a PoN principle as an *operationalization* of that principle.

Unfortunately, to the best of our knowledge no concrete guidelines on practical operationalization of PoN principles have been proposed thus far. Moreover, there has been criticism aimed towards their formulation as informal, though well-described and thorough, guidelines. In particular, the feasibility of verifying whether they can be verified in a replicable and systematic way has been criticized (cf. [5,19,17,1,16]) The latter authors have further argued that the PoN's principles in their current state are "neither precise nor comprehensive enough to be applied in an objective way to analyze practical visual software engineering notations".

One natural direction toward operationalization of PoN principles, proposed by [16], is their formalization (or formulation in mathematical terms). However, they encountered a number of challenges while attempting to formalize the first two (out of nine) principles of the PoN. Information needed to formalize the principles was posited, while acknowledging that "[we] do not yet have empirical evidence to support our assumption" [16, p. 116]. The authors similarly acknowledged that the application or formalization of a number of principles requires a base in other existing theories [16, p. 118].

In this paper we aim to better understand the inherent difficulties behind operationalization, and in particular of formalization of PoN principles. Clearly, we cannot expect to have an algorithm for computing compliance to PoN of every newly introduced visual notation. It is not only due to the fact that visual notations usually do not have fully formalized representations, but also that some PoN principles rely on information that can only be obtained from cognitive theories and/or empirical data from users of the particular new notation. This leads to the question to what extent aspects of the PoN can be formalized. As a starting point, we define the notion of visual notation in set-theoretical terms, which provide a formal ground for our analysis. We then use these terms to answer the following research questions:

RQ1. What elements are involved in operationalizing each PoN principle with respect to a given visual notation?

RQ2. What effect do these elements have on the feasibility of operationalizing each principle into a well-defined mathematical question?

RQ1 will be addressed by analyzing the different PoN principles, analyzing them for the basic elements required for their employment. These findings will be used for investigating RQ2, where we will discuss the way in which the identified elements can be used to address the operationalization of the principles in a structured mathematical way. Finally, we further reflect on what the identified challenges mean in terms of needed research efforts.

By addressing the above questions, this paper takes a first step towards grounding the PoN in more formal and operational foundations.

2 PoN Principles Overview

This section provides a brief overview of the principles of the PoN. Table 1 presents the nine principles of the PoN together with their high-level descriptions.

Table 1: Overview of the PoN's nine principles.

Principle	**Explanation**
Semiotic clarity	There should be a one-to-one correspondence between elements of the language and graphical symbols
Perceptual discriminability	Different symbols should be clearly distinguishable from each other
Semantic transparency	The use of the visual representations whose appearances suggest their meaning.
Complexity management	Notation includes explicit mechanisms for dealing with complexity
Cognitive integration	Notation include explicit mechanisms to support the integration of information from different diagrams
Visual expressiveness	The use of the full range and capacities of visual variables
Dual coding	Use of text to complement graphics
Graphic economy	The number of different graphical symbols should be cognitively manageable
Cognitive fit	Use of different visual dialects for different tasks and audiences

From the above descriptions it becomes clear that the principles involve different types of elements, which have a direct impact on their operationalization.

The first principle of semiotic clarity, e.g., mentions *language elements* and *graphical symbols*. Given the language elements, graphical symbols and a mapping between them, it is an easy mathematical question to determine whether the mapping is 1:1.

The second principle, perceptual discriminability, again speaks of graphical symbols, but this time requires their *distinguishability*. Note, however, that given two graphical symbols, establishing their distinguishability is not a mathematical question. Symbols distinguishable for a typical human, may not be distinguishable for a color-blind one. And even after determining the target user, we need to know the values of the parameters of the representation medium in which the notation is used (such as number of pixels of the presented UI, texture, and color difference, in computer aided environment) which may affect distinguishability. But if, for instance, we know that a difference of more than 10 pixels is distinguishable, then given two shapes establishing their distinguishability becomes a mathematical question.

The semantic transparency principle, however, seems not to fall even in the latter category, as it speaks in terms of appearances (symbols) suggesting their meaning. How do we know, given a symbol, that it suggests its meaning? Suggests to whom? In what sense? How sensitive is it to, for example, cultural differences? And can this be verified?

In what follows we propose some notions which provide a formal ground for making the above distinctions in a more systematic way.

3 An Analysis of PoN Operationalization

3.1 A Set-Theoretical Framework for PoN

The basic element of our framework is a *graphical symbol*. Each graphical symbol g has an *appearance*, which can be represented using appearance *variables* (such as size, shape, texture, etc) which may assume different values from associated ranges. We shall identify appearance of a given graphical symbol $\mathsf{Ap}(g)$ with some assignment of values to appearance variables.
Note that $\mathsf{Ap}(g)$ is an abstraction of the actual symbol g; thus, for example the following three symbols have the same appearance in terms of variables shape, color and size, although they can be distinguished by texture and line style:

Fig. 1: Symbols equivalent in terms of shape, color and size

In addition to appearance, each graphical symbol g also has an associated meaning $\mathbf{I}(g)$ which takes the form of a semantic construct.
The above can be formalized as follows:

Definition 1. *A visual notation is a triple* $\mathbf{V} = (\mathcal{G}(\mathbf{V}), \mathcal{L}(\mathbf{V}), \mathcal{R}(\mathbf{V}))$ *where* $\mathcal{G}(\mathbf{V})$ *is a set of graphical symbols,* $\mathcal{L}(\mathbf{V})$ *is a set of textual symbols (letters) and* $\mathcal{R}(\mathbf{V})$ *is a set of rules for the composition of elements from* $\mathcal{G} \cup \mathcal{L}$ *into models in* $\mathbf{V}$*. The closure of* $\mathcal{G} \cup \mathcal{L}$ *under* R *provides the set of possible models that can be constructed over* $\mathbf{V}$*, denoted by* $\mathsf{Models}(\mathbf{V})$*.*
The set of appearance variables of $\mathcal{G}$*, together with their associated ranges, is denoted by* $\mathsf{ApVars}(\mathbf{V}) = \{\mathsf{Ap}(g) \mid g \in \mathcal{G}(\mathbf{V})\}$*.*
An interpretation for $\mathbf{V}$ *is a mapping* $\mathbf{I} : \mathsf{Models}(\mathbf{V}) \rightarrow \mathbf{C}$*, where* $\mathbf{C}$ *is a set of semantic constructs.*

For example, consider the following excerpt from the BPMN 2.0 OMG standard [13] introducing the concept of web task. The graphical symbols discussed here are a rectangle with rounded corners and a rectangle with rounded corners that has a marker in its left corner. These two symbols are mapped to the semantic constructs Task and Web Task respectively:

Service Task

A **Service Task** is a **Task** that uses some sort of service, which could be a Web service or an automated application.

A **Service Task** object shares the same shape as the **Task**, which is a rectangle that has rounded corners. However, there is a graphical marker in the upper left corner of the shape that indicates that the **Task** is a **Service Task** (see Figure 10.11).

A **Service Task** is a rounded corner rectangle that MUST be drawn with a single thin line and includes a marker that distinguishes the shape from other **Task** types (as shown in Figure 10.11).

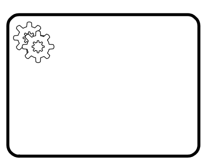

Figure 10.11 - A Service Task Object

Fig. 2: Excerpt from the BPMN 2.0 OMG standard [13, p. 158]

The above set-theoretical terms will be useful in the sequel to make the meaning of PoN more precise and make a clearer distinction between the principles. In particular, we will distinguish between the following levels of notation:

- Level 1: principles considering only symbols from $\mathcal{G}(\mathbf{V})$.
- Level 2: principles considering symbols from $\mathcal{G}(\mathbf{V})$ together with the mapping $\mathbf{I}$ to semantic constructs.
- Level 3: principles considering elements from $\mathsf{Models}(\mathbf{V})$ as a whole (which consist of symbols from $\mathcal{G}(\mathbf{V})$, as well as from $\mathcal{L}(\mathbf{V})$).

3.2 Operationalization Analysis

In what follows we analyze each of the PoN principles in terms of their operationalization. In other words, given a visual notation, we ask what it takes to check whether a certain principle applies to it. In addition to the levels of notation specified above, we consider also an additional dimension: the extra information (e.g., particular thresholds, measures, definitions or evaluation) that is needed for operationalization.

Semiotic clarity: requires a visual notation to have a 1:1 correspondence between semantic constructs and graphical symbols. This principle implies that when there is a graphical symbol in the notation (e.g., a stickman), it is used for representing solely one meaningful semantic construct or thing from the universe of discourse (e.g., a person). The PoN provides a number of exact instructions to ensure this, based on ontological literature. Concretely, the following situations should be avoided: one construct represented by multiple graphical symbols, multiple constructs represented by the same graphical symbol, graphical symbols that do not correspond to any construct, and constructs that do not have any graphical symbols. While ontological theory has been used to ground the instructions given for this principle, the given simple rules require no acquaintance with other theoretical frameworks. An example of a notation that does not satisfy the criteria is i*, which has 27 semantically distinct relationships, but only five graphically distinct graphical symbols for relationships [10].
Set-theoretical formulation: Let $\mathbf{V}$ be a visual notation and $\mathbf{I}$ an interpretation for $\mathbf{V}$. We say that $\mathbf{V}$ enjoys semiotic clarity if the restriction of $\mathbf{I}$ to $\mathcal{G}(\mathbf{V})$ is 1:1.
Classification: The operationalization of this principle requires both $\mathcal{G}(\mathbf{V})$ and the semantic mapping $\mathbf{I}$ (level 2 of notation). Once the sets of graphical symbols, semantic constructs and the mapping between them are established, checking whether the mapping is 1:1 does not require any extra information. The main challenge here remains the required explicit specification of all needed constructs.

Dual coding: requires a visual notation to use text to complement graphics. For example, using commonly understood and agreed upon words to complement graphical symbols to further ensure they are interpreted unambiguously. The PoN suggests using both annotations (i.e., including textual explanations in analog to comments in source code) and hybrid symbols (i.e., textual reinforcement of visual symbol meaning). Further requirements placed upon such text are not fully clarified. For example, it is not clear whether the use of free form natural language is preferred over, e.g., a controlled or structured natural language (e.g., SBVR[1]), or whether there should be limits to the length of text (i.e., concrete string limits). Many modeling languages satisfy the core criteria of dual coding by letting users place textual annotations. ORM 2.0 [6] could be

[1] http://www.omg.org/spec/SBVR/

a good example of potential further operationalization, providing its textual annotation of ternary fact types being written in a way that follows the structure and layout of the related visual elements.
Set-theoretical formulation: Let $\mathbf{V}$ be a visual notation. We say that $\mathbf{V}$ enjoys dual coding if there are models in $\mathsf{Models}(\mathbf{V})$ which include elements from both $\mathcal{G}(\mathbf{V})$ and $\mathcal{L}(\mathbf{V})$.
Classification: This principle involves $\mathsf{Models}(\mathbf{V})$ (level 3 of notation). Interpreting the question of dual coding in the Boolean sense, it requires no extra information. However, it seems that the intended meaning here is more than just Boolean (a yes/no question); additional external information could give more valuable insights into further constraints placed on the text like e.g., cognitive limits on the amount of text that is efficiently parsed. The vague formulation of this principle leaves room for a variety of interpretations, and the extent to which text should be combined with symbols should be clarified before operationalization can be made possible.

Graphic economy: requires the visual notation to make economical use of graphical symbols. The size of the notation's visual vocabulary should not exceed the cognitive limit of how many distinct visual symbols can be effectively recognized. The PoN references existing and widely known work, re-iterating that people can discriminate between around six different visual graphical symbols, and therefore proposes to not exceed this number. Regardless of *how* this is achieved, for which the PoN gives a number of different strategies and instructions, operationalizing this criteria and verifying whether it holds is simple, requiring only the visual notation itself to check how many distinct graphical symbols it has. An example of a visual notation that likely satisfies most operationalizations would be petri nets and ER-diagrams, both consisting of very few visually distinct elements. Petri net models indeed appear out of only three elements (four, if one includes tokens): places, transitions and arcs. Of course, the more specialized a visual notation becomes, the harder it typically is to keep the total number of graphical symbols down; for example, the total number of graphical symbols in BPMN has grown to be over 50 [11].
Set-theoretical formulation: Let $\mathbf{V}$ be a visual notation. We say that $\mathbf{V}$ is graphically economic with respect to a threshold n if $|\mathcal{G}(\mathbf{V})| < n$.
Classification: This principle involves only $\mathcal{G}(\mathbf{V})$ (level 1 of notation), and given the threshold n requires no extra information for operationalization.

Complexity management: This is similar to graphic economy, except that the formulation here is on a diagram (or model) level. Visual complexity of entire diagrams often becomes high due to a large number of elements in a diagram. The PoN grounds itself in literature showing that the number of diagram elements that a person can comprehend at a time is limited by working-memory capacity, and should this limit be crossed, the degree of comprehension decreases significantly. To be cognitively effective, a visual notation should thus avoid such situations from occurring. While the PoN clearly states that complexity manage-

ment is about preventing a particular threshold of comprehension being crossed, it does not offer values for such a threshold.
Set-theoretical formulation: Given such threshold n and a way to establish the size of a model in $\mathbf{V}$, this principle can be taken to mean that for every $m \in \mathsf{Models}(\mathbf{V})$, $|m| < n$.
Classification: This principle involves $\mathsf{Models}(\mathbf{V})$ (level 3 of notation). If the question of complexity management is understood in a Boolean way, no extra information is required. However, it seems that checking a Boolean assertion that this threshold is never crossed is not useful, and one needs to check that the notation offers good enough mechanisms to ensure it can be dealt with, such as having semantic constructs for subsystems, decomposable constructs, and relevant syntactical diagrammatic conventions for decomposing diagrams. Thus also in this case the abstract formulation of the principle leaves room for many interpretations and should be further clarified. Therefore, the extra information required here is what is what exactly is understood by "complexity management mechanisms".

Cognitive integration: requires a visual notation to incorporate explicit mechanisms to support the integration of information from different diagrams. For example, in ArchiMate where an enterprise is described by the three layers of business, application, and technology, models can exist for each separate layer, but the information therein has to be able to be directly related to each other. Short of its extensive description of potential implementations, the concrete features that the PoN argues a visual notation needs to have are: "Mechanisms to help the reader assemble information from separate diagrams into a coherent mental representation of the system", and "Perceptual cues to simplify navigation and transitions between diagrams." However, the problem is that while ostensibly only the visual notation is needed in order to check whether such mechanisms exist, the PoN describes what can be done to implement these requirements in a visual notation only as suggestions, not as hard requirements. For example, to implement contextualization, the PoN reasons that one can "include all directly related elements from other diagrams (its "immediate neighborhood") as foreign elements."
Set-theoretical formulation: this principle can be taken to mean that R has integration mechanisms.
Classification: This principle is formulated in terms of $\mathsf{Models}(\mathbf{V})$ (level 3 of notation). As in the previous principle, although a Boolean condition could be formulated here, it seems to be not useful enough, and the vague formulation of the principle should be further elaborated, providing as extra information a working definition of "integration mechanisms".

Perceptual discriminability: requires a visual notation to have clearly distinguishable symbols. This means that the main visual elements used are not strongly similar, or difficult of being discriminated. The PoN operationalizes this as having to investigate the visual distance between symbols, basing it on ex-

isting discriminability thresholds. The primary suggestions given are to use the shape of symbols as their primary discriminant, to introduce redundant coding in the sense of employing multiple visual variables to distinguish between graphical symbols (e.g., shape and color), ensuring a perceptual pop out by having each visual element have at least one unique visual variable (e.g., a particular concept is always, and uniquely visualized as a square), as well as using textual differentiation. In order to verify this principle, the visual notation and its specification are needed, complemented with suitable additional information grounding the choice for discriminability thresholds.
Set-theoretical reformulation: Let $Disc$ be a discriminability relation on $\mathcal{G}(\mathbf{V})$. We say that a visual notation $\mathbf{V}$ enjoys perceptual discriminability if for every $g_1, g_2 \in \mathcal{G}$, $Disc(g_1, g_2)$ holds.
Classification: This principle uses only $\mathcal{G}(\mathbf{V})$ (level 1 of notation). The extra information required here is the measure $Disc$. As discriminability thresholds are published and referenced explicitly by the PoN, defining such measures in a natural way seems feasible. Complications here might stem from a need to validate that the used additional information accounts for potentially expected complications in discriminability thresholds, such as for instance colorblind users of a modeling language who cannot distinguish between some used colors, thereby potentially reducing the overall discriminability (e.g., if red and green are used to distinguish elements, for a colorblind user the discriminability would not be achieved).

Visual expressiveness: concerns the number of visual variables used in the notation, such as color, shape and texture. The PoN recommends that notation designers: use color (though only for redundant coding); ensure that form follows content, meaning that the choice of visual variables should not be arbitrary but rather match the properties of the visual variables to the properties of the information to be represented. This is operationalized in more detail by explaining that (1) the power of the visual variable (nominal, ordinal, interval) should be greater than or equal to the measurement level of the information; and, (2) the capacity defined as the number of perceptible steps ranging from two to infinity should be greater than or equal to the number of values required.
Set-theoretical reformulation: Let $WellUsed$ be an expressiveness predicate defined on the set $\mathsf{ApVar}(\mathbf{V})$. We say that a visual notation $\mathbf{V}$ enjoys visual expressiveness if for every $v \in \mathsf{ApVar}(\mathbf{V})$, $WellUsed(v)$ holds.
Classification: This principle uses only $\mathcal{G}(\mathbf{V})$ and their visual variables (level 1 of notation). The extra information required for operationalization of this principle is the availability of the expressiveness $WellUsed$ predicate. This is not trivial, as the PoN provides many examples for the range of visual expressiveness, including what elements contribute and detract (e.g., use of color, positioning, size, brightness), but does not detail hard values for minimum or maximum thresholds. The PoN provides data on the total capacity of different visual variables in terms of how distinctive they are for human observers (e.g., orientation yielding four distinct variables), but does not explicitly say to what degree to use it. Thus, determining the parametric values for the expressiveness predicate, which itself

is to be built on measuring the different visual variables requires interpretation of relevant literature to determine suitable values.

Semantic transparency: deals with ensuring that visual representations suggest their meaning via their appearance. The PoN describes it as a continuum of meaning, arguing that it "formalizes informal notions of "naturalness" or "intuitiveness" that are often used when discussing visual notations" [9].
Set-theoretical reformulation: We say that a visual notation $\mathbf{V}$ enjoys semantic transparency if for every $g \in \mathcal{G}(\mathbf{V})$, $\mathbf{I}(g)$ is "suggested".
Classification: This principle uses both $\mathcal{G}(\mathbf{V})$ and $\mathbf{I}$ (level 2 of notation). The crucial extra information we need here is a more precise characterization of what it means for a semantic meaning to be "suggested" by a graphical symbol. This of course cannot be determined *a priori* and needs empirical evaluation. The PoN describes a range of how suggestive visual symbols can be characterized, from fully transparent (i.e., conveying its intended meaning) to perverse (i.e., conveying a different, incorrect meaning). Empirical work directly involving the user is needed to determine how well a particular symbol suggests its intended meaning.

However, instead of providing a formal notion, the PoN suggests avoiding situations where novice readers would likely infer a different meaning from appearance, and further advocates the use of icons as symbols that perceptually resemble the concepts they represent. This principle seemingly can only be performed by directly involving users. Furthermore, cultural and temporal ("zeitgeist") dependency of such suggested meaning would make it more challenging to generalize findings from users. While some icons and symbols might have meaning for a group of people, few of them are universal. Furthermore, the meaning of icons or symbolism changes over time, making operationalizations also temporally bound. A practical example of how suggested meaning is clearly culture bound can be found in an application of the PoN to i* [10], a goal modeling notation. In this notation, it is proposed to distinguish different kinds of acting entities, where agents are proposed to be depicted with "black sunglasses and a pistol", arguing that users would make "an association of the 007 kind." This presupposes a shared cultural knowledge between the designer and user of the notation that needs empirical grounding.

Cognitive fit: concerns personalizing the visual notation to the target audience and ensuring that it "fits" with the cognitive background and skills of different users and tasks it is used for. For example, when people with different backgrounds and skill sets use the notation, it is important they can all use it at a minimum level of proficiency. The PoN recommends focusing on taking into account at least (1) expert-novice differences, and (2) the representational medium. While particular instructions are given for how to optimize a notation for either expert or novice, the principle itself centers on ensuring that the visual notation does not exhibit visual monolinguism. In a way, only the visual notation is needed to verify whether this principle holds: one can check whether different dialects

for particular users or tasks exist. However, the core difficulty of the principle is that for a given notation these differences need to be identified first. Thus, users have to be directly involved, leading to the same challenges described for other principles, such as semantic transparency, requiring direct user involvement. For example, say that the visual notation of some process modeling language uses realistic pictograms in order to clearly visualize what things are needed for a particular task. Specifically, a realistic pictogram of a wrench is used for a task of 'screwing down bolts'. If this notation has the requirement that it can be drawn on paper, how do we actually verify whether needing to draw a wrench is difficult or not? Without knowing the users, one cannot postulate their artistic skill, or their inclination to spend time drawing realistic depictions. Regardless of whether it was intended, BPMN is an example of a language, which, in practical use seems to satisfy what cognitive fit aims to achieve. It has been viewed as consisting of a number of 'sets' of functionality, a common core, extended core, specialist set, overhead in use by people of varying levels of expertise and focus. [11]
Set-theoretical reformulation: this principle seems to us to be the most vague of all, and no set-theoretical reformulation in the terms defined in this paper can be suggested.
Classification: this principle uses $\mathsf{Models}(\mathbf{V})$ (level 3 of notation). The starting point for extra information required here is providing more concise characterization of the elements involved in the formulation of this principle.

4 Summary & Identified Concerns

The above discussion provides a number of new insights into the inherent difficulty of operationalizing PoN principles. First of all, two dimensions emerge from our analysis, which may provide indications on the feasibility of operationalization of the principles. The first is the distinction between the different layers of visual notation addressed by each principle. Some principles are targeted at the level of an individual symbol and its structure, others at the interplay of the symbols with their semantic constructs, and some target the interplay of many symbols (i.e., a model). These different levels as referenced in Section 3.1 increase the challenge of clearly operationalizing, as the increase in elements that have to be considered make clear and precise verification more challenging.
The second is the distinction between the different types of extra information needed for operationalization of the principle. Sometimes additional information is needed that is both simple to gather and interpret, such as widely published accounts of how many distinct graphical symbols the human mind can perceive at a time. However, when more information has to be distilled from more complicated literature (e.g., scientific theory), an additional challenge arises of ensuring the correct selection and interpretation of that information. Finally, when information specific to users is needed (e.g., to determine what meaning is 'suggested' by a symbol), a whole new challenge appears with the need to design empirical work, argue for the validity of elicited information, and reason how it either generalizes or applies to the intended users of the visual notation.

Table 2 provides an overview of our findings. For each principle it presents the notation level, a set-theoretical formulation of the principle, and the extra information that is needed to achieve operationalization.

Table 2: Summary of PoN Operationalization Analysis

Principle	Set-theoretical Desc.	Elements used	Extra info required
SemCl	$\mathbf{I}\|\mathcal{G}$ is 1-1	$\mathcal{G}(\mathbf{V}) + \mathbf{I}$ (level 2)	-
PerDisc	$\forall g_1, g_2 \in \mathcal{G}(\mathbf{V}) : Disc(g_1, g_2)$	$\mathcal{G}(\mathbf{V})$ (level 1)	**measure** $Disc$ on $\mathcal{G}(\mathbf{V})$
SemTr	$\forall g \in \mathcal{G}(\mathbf{V})$: g **"suggests"** $M(g)$	$\mathcal{G}(\mathbf{V}) + \mathbf{I}$ (level 2)	**evaluation** of "suggestiveness"
CmpMng	R has **"compl. management"**	$\mathsf{Models}(\mathbf{V})$(level 3)	**defn.** of **"compl. management"**
CogInt	R allows **"integration"**	$\mathsf{Models}(\mathbf{V})$(level 3)	**defn.** of **"integration"**
VisExp	$\forall v \in \mathsf{ApVar}(\mathbf{V})$, $\mathrm{WellUsed}(v)$	$\mathcal{G}(\mathbf{V})$ (level 1)	**measure** WellUsed on $\mathsf{ApVar}(\mathbf{V})$
DualC	Some $m \in \mathsf{Models}(\mathbf{V})$ combine symbols & text	$\mathsf{Models}(\mathbf{V})$(level 3)	-
GrE	$\mathcal{G}(\mathbf{V}) < n$	$\mathcal{G}(\mathbf{V})$	**threshold** n
CogFit	?	$\mathsf{Models}(\mathbf{V})$(level 3)	**evaluation** of "cog. fit"

To the extent of our knowledge, dedicated operationalization efforts so far address only two principles out of nine, focusing on semiotic clarity and perceptual discriminability [16]. These two principles are arguably among the best candidates for operationalization as they provide clear, quantitative judgement criteria, and involve the lowest degree of subjective interpretation[2]. Indeed, our classification of the principles supports this view. Another good candidate for formalization, according to our classification, seems to be visual expressiveness. The most challenging principle, according to Table 2, seems to be cognitive fit. The most vague principles, requiring a reformulation in precise terms, are complexity management and cognitive integration.

Below we summarize a number of further concerns that should be addressed in the context of PoN operationalization:

Vague satisfaction criteria. A significant problem in operationalization of the PoN is the vague satisfaction criteria of many principles. While it is clearly stated what a principle should do, or achieve, the exact details on how to achieve that are left up to the theory's wielder. For example, for cognitive integration we can check a Boolean assertion that structures exist to support e.g., modularization or clustering. However, this says little about how successfully such structures will be used, as their design in itself is also subject to cognitive factors. Thus a degree-based approach is more appropriate here.

Relative impact of satisfying a principle is unclear. Given that some principles are defined in such a way that their satisfaction is almost trivial (e.g., dual coding not saying anything about the *kind* or *structure* of complementary text), how much each individual principle contributes to the overall cognitive effectiveness of a visual notation is unclear. This also makes it harder to know what principles to focus, or spend most time on should they prove challenging for a particular notation.

[2] Nonetheless, existing work [16] seems to take debatable choices, such as seemingly arbitrary weights for distinguishing visual distance variables, whose objective nature can also be discussed.

Operationalization interrelations. An additional complication arises from the relationships that exist between the different principles. Given that multiple principles have been documented to have positive or negative influence on each other (for example, increasing graphic economy can decrease semiotic clarity), operationalization of one principle may involve having to operationalize multiple principles concurrently. For example, when considering semiotic clarity, one should also take into account graphic economy, which requires taking visual expressiveness into account, which in its turn requires additional external information. Gaining a better understanding of the interrelations between the principles is thus crucial for their operationalization.

5 Concluding Outlook

This paper presented a preliminary analysis of PoN principles with respect to difficulties of their operationalization. The main contribution of this work is establishing a formal ground for distinction of different aspects that pose difficulties for operationalization of PoN principles. Using this distinction, different types of efforts can be directed at different principles, e.g., reducing vagueness of formulations, providing concrete mathematical metrics and/or methods for empirical evaluation.

Our most immediate direction for future research is using empirical methods to establish the relative importance of each principle for users of particular modeling domains (e.g., software architecture, business processes). Such empirically grounded data can be used to more clearly operationalize domain-specific 'instantiations' of the PoN, and also show where principles that are mathematical in their nature, but afford for more complex evaluation given the involvement of additional elements, can and should be raised to a higher level of evaluation.

References

1. Giraldo, F.D., España, S., Pineda, M.A., Giraldo, W.J., Pastor, O.: Conciliating model-driven engineering with technical debt using a quality framework. In: Information Systems Engineering in Complex Environments, pp. 199–214. Springer (2014)
2. Granada, D., Vara, J.M., Brambilla, M., Bollati, V., Marcos, E.: Analysing the cognitive effectiveness of the webml visual notation. Software & Systems Modeling pp. 1–33 (2013)
3. Green, T.R.G., Petre, M.: Usability analysis of visual programming environments: a cognitive dimensions framework. Journal of Visual Languages & Computing 7(2), 131–174 (1996)
4. Gregor, S.: The nature of theory in information systems. MIS quarterly pp. 611–642 (2006)
5. Gulden, J., Reijers, H.A.: Toward advanced visualization techniques for conceptual modeling. In: Proceedings of the CAiSE Forum 2015 Stockholm, Sweden, June 8-12 (2015)
6. Halpin, T.: Orm 2. In: On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops. pp. 676–687. Springer (2005)

7. 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)
8. Moody, D., van Hillegersberg, J.: Evaluating the visual syntax of uml: An analysis of the cognitive effectiveness of the uml family of diagrams. In: Software Language Engineering, pp. 16–34. Springer (2008)
9. Moody, D.L.: The physics of notations: toward a scientific basis for constructing visual notations in software engineering. Software Engineering, IEEE Transactions on 35(6), 756–779 (2009)
10. Moody, D.L., Heymans, P., Matulevičius, R.: Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation. Requirements Engineering 15(2), 141–175 (2010)
11. zur Muehlen, M., Recker, J.: How much bpmn do you need. Posted at http://www.bpm-research. com/2008/03/03/how-much-bpmn-do-you-need (2008)
12. Narayanan, N.H., Hübscher, R.: Visual language theory: Towards a human-computer interaction perspective. In: Visual language theory, pp. 87–128. Springer (1998)
13. (OMG), O.M.G.: Business process model and notation (BPMN) version 2.0. Tech. rep. (jan 2011), `http://taval.de/publications/{BPMN}20`
14. Plataniotis, G., de Kinderen, S., Proper, H.A.: Ea anamnesis: An approach for decision making analysis in enterprise architecture. International Journal of Information System Modeling and Design (IJISMD) 5(3), 75–95 (July 2014), `http://dx.doi.org/10.4018/ijismd.2014070104`
15. Schuette, R., Rotthowe, T.: The guidelines of modeling–an approach to enhance the quality in information models. In: Conceptual Modeling–ER98, pp. 240–254. Springer (1998)
16. Störrle, H., Fish, A.: Towards an operationalization of the physics of notations for the analysis of visual languages. In: Model-Driven Engineering Languages and Systems, pp. 104–120. Springer (2013)
17. van der Linden, D., Hadar, I.: Cognitive effectiveness of conceptual modeling languages: Examining professional modelers. In: Proceedings of the 5th IEEE International Workshop on Empirical Requirements Engineering (EmpiRE). IEEE (2015)
18. van der Linden, D., Hadar, I.: Evaluating the evaluators – an analysis of cognitive effectiveness improvement efforts for visual notations. In: Proceedings of the 11th International Conference on Evaluation of Novel Approaches to Software Engineering. INSTICC (2016)
19. van der Linden, D., Hadar, I.: User Involvement in Applications of the PoN. In: Proceedings of the 4th International Workshop on Cognitive Aspects of Information Systems Engineering. Springer (2016)
20. Van Zee, M., Plataniotis, G., van der Linden, D., Marosin, D.: Formalizing enterprise architecture decision models using integrity constraints. In: Business Informatics (CBI), 2014 IEEE 16th Conference on. vol. 1, pp. 143–150. IEEE (2014)