

On the Semantic Transparency of Visual Notations: Experiments with UML

Amine El Kouhen¹, Abdelouahed Gherbi², Cédric Dumoulin³, and Ferhat Khendek¹

¹ Faculty of Engineering and Computer Sciences, Concordia university, Montréal, QC, Canada

{[elkouhen](mailto:elkouhen@encs.concordia.ca),[khendek](mailto:khendek@encs.concordia.ca)}@encs.concordia.ca

² Department of Software and IT Engineering, École de Technologie Supérieure, Montréal, QC, Canada

abdelouahed.gherbi@etsmtl.ca

³ University of Lille, CRISTAL CNRS UMR 9189, Cite scientifique - Batiment M3, Villeneuve d'Ascq, France
cedric.dumoulin@univ-lille1.fr

Abstract. Graphical notations designed by committees in the context of standardization bodies, like Object Management Group (OMG), are widely used in the industry and academia. Naive users of these notations have limited background on visualization, documentation and specification of workflows, data or software systems. Several studies have pointed out the fact that these notations do not convey any particular semantics and their understanding is not perceptually immediate. As reported in these studies, this lack of semantic transparency increases the cognitive load to differentiate between concepts, slows down the learning and comprehension of the language constructs. This paper reports on a set of experiments that confirm the lack of semantic transparency of the Unified Modeling Language (UML) as designed by OMG and compares this standard to alternative solutions where naive users are involved in the design of the notations to speed-up the learning of these languages to new users.

Keywords: Visual Languages · UML · Semantic Transparency · Crowdsourcing

1 Introduction

Over the past three decades, visual languages have been gaining in popularity. Several of these languages, such as the Unified Modeling Language (UML) have been designed by standardization bodies involving different contributors, such as language experts, tool vendors, (expert) users, etc. As reported in [28] these languages, we refer to as committee-designed languages, have been developed in a bottom-up approach by reusing existing notations and through consensus among the experts. As in [28, 31] we believe that this is not the best approach, especially when the target audience includes naive users.

Several observations on visual languages have been made over time. The current development process strongly emphasizes the domain conceptualization (i.e. building the abstract syntaxes) and often relegates the visual notations (concrete syntaxes) and their semantic transparency as secondary products (byproducts). However, the visual notation is the first contact of the users with the modeling language and its semantic transparency plays a crucial role in its acceptance. The current development process is criticized by research in diagrammatic reasoning, which shows that the form of representations has an equal, if not greater, influence on cognitive effectiveness as their content [23, 39]. A major incentive for using visual notations is the widely-held belief that they convey information more effectively than text, especially to novices [4]. Committee-designed visual notations can sometimes be very distant from semantic concepts they represent. As reported in [28, 13, 29] this lack of semantic transparency increases the cognitive load to differentiate between concepts and slows down both the recognition and the learning of the language constructs. In [9], the authors suggest that the difficulty of understanding many of UML notations “*supports the argument that the UML may be too complex*”. The authors in [28, 13, 29] evaluated the visual syntax of several committee-designed languages using a set of evidence-based principles for designing cognitively effective visual notations. The analysis reveals some serious flaws in the notation in terms of cognitive effectiveness of these languages, which is defined as the speed, ease and accuracy with which a representation can be processed by the human mind [23].

In this paper, we ask a rather simple question: to design notations that are understandable by new users, why not involve them in the notation development process? If involving end-users in the development of software systems (e.g. participatory design [32], user-centred design) is working well, why this should not also be the case for graphical notations? The purpose is not to make modeling languages understandable without learning, but to speed-up the learning and comprehension of the language constructs and reducing misinterpretations due to the notations complexity.

For this purpose we conduct a set of experiments on UML involving end-users. Some of these experiments have been proposed and applied to another language in [5]. The main difference is our comparison between the outcome of these experiments and the standard UML notations. We have also used as input for this comparison a set of notations created by experts in cognitive sciences according to the Physics of Notations theory, which was not the case in [5]. On the other hand, [5] has used the recognition experiment which was not necessary in our process.

Because of space limitations, we selected only a set of visual elements of UML. We believe this set represents a good sample of UML diagrams widely used and easily understandable without deep technical background. For instance, class, statechart and use-case diagrams are the most widely used UML diagrams by non-experts users [38] and are included in this study. The other diagrams, which are created and used only by system designer experts in their domains and without direct impact to other stakeholders are out of the scope of this study.

The goal of this paper is not to redefine the visual syntax of UML but to show the importance of involving end-users in the design of visual notations. We show the importance of involving end-users actively in the notations design process as *co-designers* rather than as passive *consumers*. The broader goal of this paper is to raise awareness about the importance of the semantic transparency in the acceptance of a modeling language, which has historically received little attention.

2 Background and Related Research

One of the main advantages behind the use of modeling languages is their ability to provide to their target users a set of concrete artifacts (visual notations) that can directly express related domain abstractions (concepts) in a concise, complete and unambiguous way [16]. According to [28], visual representations have greater effect on understanding and performance than their content.

Existing approaches for designing visual notations consist of proposing symbols and voting on them (i.e. expert consensus). For example, in UML diagrams, symbols are conventional shapes on which iconic markers are added. However, symbol shapes seem not to convey any particular semantics: there is no explicit rationale to represent a Class as a rectangle, an action as a rounded rectangle and a use case as an ellipse. The differentiation of UML notations is not perceptually immediate, it is purely conventional. According to [37], to have an unambiguous modeling language its symbols should provide cues to their meaning. Semantically direct representations reduce cognitive load: their meaning can be perceived directly or easily. This kind of representations speeds-up concepts recognition, especially for novices [4, 27]. According to [5], current visual notation design practice is characterised by:

- An unselfconscious design approach [1]: there are no explicit principles for designing visual notations [30].
- Lack of design rationale [24]: symbols are chosen without any evidence-based decisions or rational justification [19].
- Small forms variations: similar geometrical shapes are used for different purposes [34]. Without self-conscious design principles, the range of symbols is limited by the imagination of the design team [30].
- Lack of involvement of the target audience: notations design decisions are made exclusively by experts, without the involvement of the target audience. For this reason, we propose our experimental process, which uses end-users (target audience) suggestions as inputs into the language design process made by experts.

2.1 Physics of Notations

On the whole, the most complete and referenced work on the assessment of visual notations is probably the Physics of Notations theory [30] of Moody, which is

exclusively devoted to the design, evaluation, comparison and improvement of visual notations. In this work, Moody establishes a set of nine principles defined from theory and empirical evidence and obtained from different disciplines such as: cognitive and perceptual psychology, graphic design, communication theory, cartography, etc.

Each of the principles of the Physics of Notations contains: design strategies, which may contribute towards improving visual notations regarding this principle; a different evaluation procedure or metric that can be used to compare different notations, and examples of notations that satisfy or violate the principle. These nine principles are:

1. *Visual Expressiveness*: use the full capacities of visual variables. The seminal work in the graphical communication field is Jacques Bertin’s *Semiology of Graphics* [21]. Bertin identified eight elementary visual variables, which can be used to graphically encode information. These are categorized into planar variables (the two spatial dimensions x, y) and retinal variables (features of the retinal image).

The set of visual variables define a *vocabulary* for graphical communication: a set of atomic building blocks that can be used to construct any graphical representation. Different visual variables are suitable for encoding different types of information (Fig. 1). The choice of visual variables has a major impact on cognitive effectiveness as it affects both speed and accuracy of interpretation [7, 25, 40].

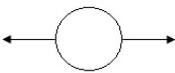
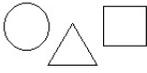
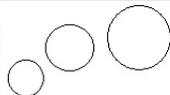
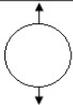
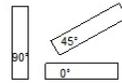
PLANAR VARIABLES	RETINAL VARIABLES		
Horizontal Position (x)	Shape	Color	Size
			
Vertical Position (y)	Brightness	Orientation	Texture
			

Fig. 1. Visual variables [21]

2. *Semiotic Clarity*: According to [14], there must be a one-to-one correspondence between elements of the language and graphical symbols. The Semiotic Clarity’s principle, maximises expressiveness by eliminating the deficit when a domain concept is not represented by any representation and reduces ambiguity by eliminating symbol overload (multiple domain concepts are represented by one representation).

UML visual notation violates the semiotic clarity principle. Many of the diagrams that need to be understandable by everyone (e.g. Structural diagrams) contain high levels of symbol redundancy and symbol overload. For example, of the 33 symbols commonly used on class diagram there are 5 synographs (15%), 20 homographs (61%) and 2 symbol excesses (6%).

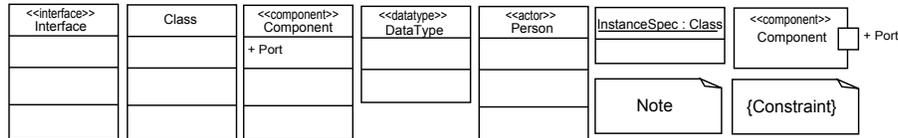


Fig. 2. Zero visual distance between UML notations (homographs)

This assessment was confirmed when we proposed in [10] a framework to specify and reuse visual languages. It turned out that we can reuse up to 71% of UML notations, which represents a high level of redundancy in this language and thus ambiguity.

Symbols are called homographs if they have zero visual distance (i.e. they have identical values for all visual variables) but represent different semantic constructs (Fig. 2). Thus, in UML the majority of graphical conventions used to mean different things. For example, in Class Diagrams, the same graphical symbol can be used to represent objects, classes, interface and attributes. Different types of relationships can also be represented using the same graphical convention e.g. package merges, package imports and dependencies are all represented using dashed arrows. These notations are differentiated only by textual stereotypes. However, text relies on sequential cognitive processes, which is an inefficient way to differentiate symbols.

3. Principle of *Perceptual Discriminability*: different symbols should be distinguishable from each other. Discriminability is a mental process that consists of symbols segregation [40, 36] from the background. Then symbols discrimination from each other. This differentiation relies on variations of visual variables between symbols [13]:
 - Shape plays the main role in this process since it represents the first concern on which we classify objects in the real world [2]. In UML, the notations differ on only a single visual variable (shape) and the values chosen are very close together: all shapes are either rectangles or rectangle variants. Given that experimental studies show that rectangles and diamonds are often confused by naive users in Entity/Relation diagrams [33].
 - Some of the symbols have zero visual distance (homographs) and are differentiated by labels or typographical characteristics. It is the case of most of UML links, which have identical shapes but differentiated only by textual stereotypes (Fig. 3). However, according to [28]: text is

an inefficient way to differentiate between symbols as it relies on slower cognitive processes.



Fig. 3. Symbol overload: UML homographs

- Colour is the most performant and cognitively expressive visual variable. The human visual system can quickly and accurately distinguish between them [40]. However, colours are prohibited in UML: The use of colours is up to tool vendors.
- 4. Principle of *Semantic Transparency*: using visual representations whose appearances suggest their meaning. We will focus on this principle when we propose our experimental process. We believe that involving target audience into design process may improve this criterion.
- 5. Principle of *Dual Coding*: enhance diagrams with textual information.
- 6. Principle of *Complexity Management*: suggest some mechanisms when dealing with notations complexity.
- 7. Principle of *Cognitive Integration*: suggest explicit mechanisms to support the integration of information from different diagrams.
- 8. Principle of *Graphic Economy*.
- 9. Principle of *Cognitive Fit*.

Indeed, these principles have already been used in several works to evaluate and improve other visual committee-designed languages such as *i** [29], Business Process Modeling Notation (BPMN) [13] and UML [28]. In the next section, we reuse the alternative notation proposed by [28], which is based on these nine principles.

3 Experiments on UML Semantic Transparency

This section summarizes a set of related experiments applied to UML Visual Syntax. The experiments and the workflow are shown in Fig. 4. A similar experimental process have been proposed and applied to another language in [5]. In this paper, we reused the Experiments 1, 2, 3 and 4 from [5]. As mentioned earlier, the main difference is our comparison between the outcome of these experiments and the standard UML notations. We also used as input for this comparison, a set of notations created by experts in cognitive sciences according to the Physics of Notations theory which is not the case in [5]. In [5], authors have used the recognition experiment, which is not the case in our process. As mentioned in the introduction, we limit these experiments to a few elements of UML visual syntax. The purpose is not to redefine the visual syntax of UML but to show the importance of involving end-users into the design decisions made generally by experts.

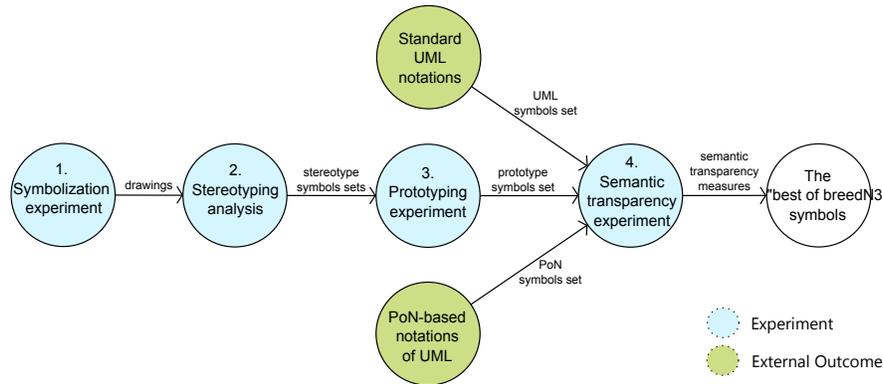


Fig. 4. Experiments Workflow

1. *Symbolization experiment*: naive participants (i.e. with background on Object-Oriented paradigm but without previous knowledge on UML) generated symbols for UML concepts (drawings).
2. *Stereotyping analysis*: we analysed the results of Experiment 1 and identified the most common symbols produced for each UML concept (stereotype symbols sets).
3. *Prototyping experiment*: other group of naive participants (different from the first one) analysed the drawings produced in Experiment 1 and identified the “best” representations for each UML concept (prototype symbols set).
4. *Semantic transparency experiment*: another group of naive users were asked to infer the meaning of 3 sets of symbols from their appearance alone : Prototypes from Experiment 3 and two external inputs, which are the Standard UML notation and the UML notation based on Physics of Notation Theory (PoN) .
5. *Identify “best of breed” symbols*: based on the results of Experiment 4, we identified the most cognitively effective symbols for each UML concept across all symbol sets.

This experimental workflow combines quantitative and qualitative research methods: Experiments 1, 2, 3 primarily use qualitative methods, while study 4 uses quantitative methods. The used data is primarily in the form of pictures (drawings). The quantitative studies measure the interpretation/recognition accuracy in combination with psychometric scales (for rating the cognitive difficulty of tasks). More detailed data on participants artifacts, samples, results as well as the coded data set and the statistical scripts are available in [11].

3.1 Symbolization Experiment

In this experiment, “naive” participants imagined and drew symbols for UML concepts, a task normally reserved for experts. There were 64 participants in

this experiment, all have background in Object-Oriented concepts. They had no previous knowledge of modeling languages in general or UML in particular: this was a requirement for participation in the study (**inclusion criterion**), to ensure participants were truly naive. UML regular users would not have been suitable participants, due to their technical orientation and knowledge (i.e. the **curse of knowledge** [18]).

Each participant was provided with a two-page questionnaire that consists of a table of constructs (we chose twelve concepts, which are frequently used in software engineering : *Class, Interface, Enumeration, Instance Specification, Component, Signal, Model, Package, Dependency, Merge, Import* and *Substitution*), their definitions, and an empty cell in which participants were instructed to draw the construct.

Participants were asked to draw the constructs in the order in which they appeared. They were instructed to produce drawings that they felt most expressive (conveyed better the meaning of the construct). They were instructed to draw as simple as possible and that the quality of drawings wasn't important: the most important thing was to represent clearly and unambiguously the meaning of the construct.

Table 1. Response rates for symbolization task

UML Construct	Non-Responses	Response Rate
Class	2	96.87%
Interface	3	95.3%
Instance Spec.	4	93.75%
Enumeration	0	100%
Component	1	98.44%
Signal	0	100%
Model	2	96.87%
Package	1	98.44%
Dependency	2	96.87%
Merge	1	98.44%
Import	1	98.44%
Substitution	2	96.87%
Average	1.58	97.5%

The participants produced a total of 749 drawings with a response rate of 97.5% for a set of twelve UML concepts, which was a high response rate given the known difficulty in “concretizing” [22] UML abstract concepts. Instance Specification (6.25%), Interface (4.68%), Class (3.13%) and Model (3.13%) received the highest number of non-responses, which is more likely to be the case for such abstract concepts. Enumeration, Signal and Package receiving less than 1% (only 1 non-response out of 64). Table 1 summarizes the response rates.

3.2 Stereotyping analysis

In this step, we analysed the results of Experiment 1 and identified the most common symbols produced for each UML concept. These defined the **stereotype symbol sets**.

The analysis was conducted by three volunteers. It was done by looking at similarity of drawings. The drawings produced in Experiment 1 were used as input for this experiment. Three copies were made of the drawings to conduct this task independently.

We used the **judges' ranking method** [22], which is an approach for reaching convergence on a set of categories. In the first step, each judge categorized the drawings produced for each concept by sorting them into categories based on their similarity (pattern-matching), following the approach described in [17].

Then, they compared their choices (categories) for each concept and agreed on a common set of categories. Finally, they selected for each concept, a set that consists of a drawing from each category (*the stereotypical category*), resulting in a stereotypical set for each concept (twelve in total).

3.3 Prototyping Experiment

For each evaluated concept, the participants studied the stereotypical sets selected in Experiment 2 and identified the “best” representation. These defined the **prototype symbol set**.

Table 2. Degree of prototypy

Concept	Degree of Convergence
Class	24.13%
Interface	31%
Instance Spec.	38%
Enumeration	72.4%
Component	44.8%
Signal	34.5%
Model	27.6%
Package	48.3%
Dependency	51.7%
Merge	31%
Import	62%
Substitution	58.6%
Average	43.67%

40 naive users participated in this experiment, all undergraduate students in computer science from multiple cultural backgrounds (different universities in Canada, France, Morocco, Algeria, Indonesia and Jordan). We used a different sample population from Experiment 1 but drawn from the **same** underlying

population. It would not have been accurate to use the same participants as in Experiment 1, as their judgements may have been biased by their own drawings.

We conducted this experiment using a form, which consists of a table showing the name and the definition of each concept with the candidate drawings (representatives from each category identified in the stereotyping study). Participants were asked to select the most expressive drawing for each concept (regardless on the aesthetic the drawing). The order of the concepts and the position of the drawings were randomized in the forms to avoid sequence effects.

The output of this experiment was a set of 12 prototypical drawings (one for each evaluated UML concept).

Table 2 shows the Degree of prototypy i.e. percentage of participants who rated the prototype drawing as the best. For all concepts, there was a high level of consensus among judgements of prototyping (43.67% in average). The highest score was for Enumeration, which achieved more than 72% agreement, and lowest for Class and Model, which achieved less than 30% agreement.

3.4 Semantic Transparency experiment

For this experiment, naive users were asked to infer the meaning of symbols from their appearance alone. The symbols were from one of 3 symbol sets, two designed by experts (the standard UML notation and the notation designed following *Physics of Notations* principles as explained in [28]) and those designed by naive end-users (the prototype symbols set from Experiments 3).

There were 120 participants, all undergraduate students in computer sciences from several universities. As in studies 1 and 3, the participants had no prior knowledge of software modeling languages or UML, so were truly naive.

There were three experimental groups (composed by 40 participants for each of them), corresponding to different levels of input:

1. **Standard** UML notations: official symbols from UML specification (unself-conscious design).
2. UML notations designed according *Physics of Notations* theory (self-conscious design) called **PoN** in table 3. The details of this notation are available in [28].
3. **Prototype** notations of UML: the best symbols produced by naive users as judged by other naive users.

We conducted this experiment using a multiple-choices questionnaire. One symbol was displayed at the top of each page (representing the **stimulus**[29]) and the complete set of UML constructs and definitions displayed in a table below (representing the possible responses). Participants were asked to choose the construct they thought most likely corresponds to the symbol. In each page, there was one correct response and 11 incorrect responses (distractors). The order in which the stimuli (symbols) were presented (i.e. order of pages) and the order in which the responses (concepts) were listed were randomized to avoid sequence effects.

Participants were instructed to work alone and not share their answers with each other. They were asked to answer each question in order and told to choose one and only one concept for each symbol presented. They were told that they could choose the same concept in response to multiple symbols.

3.5 Identify *best of breed* symbols

Table 3. “*Best of Breed*” symbols

UML Concepts	Experiment 3 : Semantic Transparency			<i>Best of breed</i> symbol
	Standard	PoN Notations	Prototype	
Enumeration	30%	45%	77.5%	
Component	22.5%	30%	70%	
Signal	5%	35%	87.5%	
Package	22.5%	20%	87.5%	
Dependency	12.5%	20%	75%	
Merge	27.5%	37.5%	72.5%	
Import	32.5%	42.5%	77.5%	
Group size	40	40	40	
Hit Rate Mean	22%	33%	71.5%	

Based on the results of steps 4, we identified the most cognitively effective symbols for each UML construct across all symbol sets.

To measure graphical symbols comprehensibility, we used the *hit rates* (i.e. percentage of correct responses). The ISO standard for testing graphical symbols [35] defines 67% as the hit rate required for acceptance of public information and safety symbols [20]. Only 9 out of 12 symbols across the 3 symbols sets met the ISO required limit for comprehensibility.

Table 3 shows the best symbols across all symbol sets in terms of hit rates. The *best of breed* symbol set includes 12 symbols from the prototype symbol set and none from the standard UML and PoN symbol sets. The mean hit rate is 71.5%, which exceeds the ISO threshold for comprehensibility of symbols. For

space limitation, we choose to show only symbols, which have met a low level of non-response in Experiment 1, a high degree of prototypy in Experiment 2 and exceed the threshold of 67% in Experiment 3.

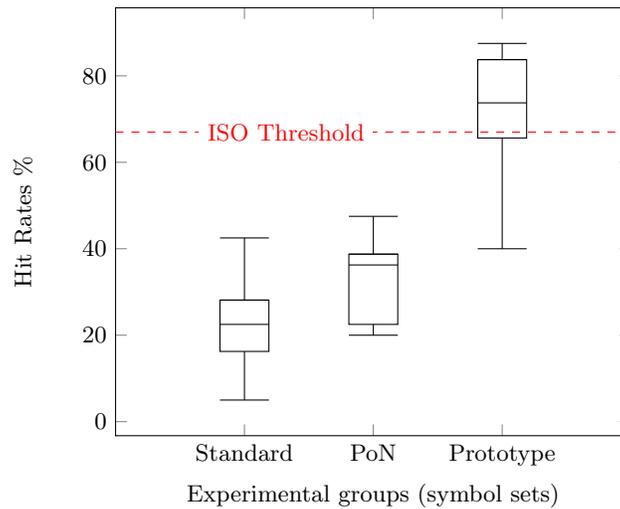


Fig. 5. Differences in hit rate between experimental groups

The differences between groups are visually confirmed by the box and whisker plot in Fig. 5. The *boxes* show confidence intervals for each group mean, while the *whiskers* show minimum and maximum values. The line through the middle of each box represents the median. Figure 5 shows a comparison of semantic transparency results for unselfconscious and self-conscious notation design. We can observe that the prototype symbol set exceeds largely the ISO threshold for comprehensibility of symbols.

Using explicit design principles (self-conscious design) significantly improves semantic transparency (supported by our hypothesis), showing that conscious efforts to improve semantic transparency are likely to be successful. The average hit rate for the *PoN* symbol set was more than 1.5 times that of the standard UML notation, meaning that *PoN* symbols were more than 1.5 times as likely to be correctly interpreted without prior explanation. Moreover, the average hit rate for the *Prototype* symbol set was more than three times that of the standard UML notation and more than twice that of the *PoN* notation, meaning that *Prototype* symbols were by far more expressive and more often interpreted correctly. More detailed results as well as the coded data set and the statistical scripts are available in [11].

In this experiment, our *a priori* hypothesis was confirmed. We find that symbols proposed and chosen by naive users are naturally based on good rules that can be found in theory and empirical approaches. We can thus observe that

user-comprehensible notations (Prototype set) have absolutely a better cognitive expressiveness and semantic transparency than the other two symbol sets combined.

3.6 Threats to validity

The validity of our experiments are categorized into internal and external validity. The internal one refers to whether an experimental condition could be sufficient to support the claim of the experiment or not. External validity refers to the generalization of experiments outcomes [6].

1. *Internal validity.* the following factors may jeopardize internal validity :
 - *Selection of subjects* : all participants in this study were randomly assigned to experimental groups to avoid the *selection bias*. As semantic transparency is culture-specific, we choose participants from different cultural and ethnical background.
 - *Instrumentation* : the same measurements were used in all experimental groups to avoid the *measurement bias*. As we mentioned before, the experiments materials were presented randomly to avoid the *sequence effect*.
 - *Statistical regression* : the results may be different if we applied this study on another subset of UML notations. However, in this study we get 300% of improvement, which is the same as the study [5] for another modeling language (i*).
2. *External validity.* the following factors may threaten external validity :
 - To avoid the *expertise bias*, we used naive users in all experiments. We considered computer science students as adequate participants as they must know Object-Oriented concepts to draw them, but we considered only students with no previous knowledge of modeling languages in general or UML specifically to avoid the *knowledge bias* (the curse of knowledge [18]).
 - We used different population samples in the last experiment. It would not have been accurate to use the same participants across the experiments, as their judgements may have been biased by the drawings they produced.
 - Our study evaluated the comprehension of symbols as visual unit rather than complete diagrams, which can represents a threat to this study. Thus the results and their interpretation cannot be generalizable to complete diagrams.
 - We did not consider other principles such as the Principle of Cognitive Fit (ease of drawing of symbols) or the Principle of Graphical Economy (usage of the space), which are important in modeling languages design. For example, the signal symbol may be hard to use when the sender and the receiver are not close in the diagram.
 - In this study, we used hit rates (positive values) as measurements for the semantic transparency. However, Semantic transparency is not a binary

state but a sliding scale (Fig. 6) defined from -1 to +1 : -1 for symbols whose appearance implies an incorrect meaning (semantically perverse) and +1 for symbols whose appearance implies the correct meaning (semantically transparent) [30]. Such kind of measurements need further investigations.

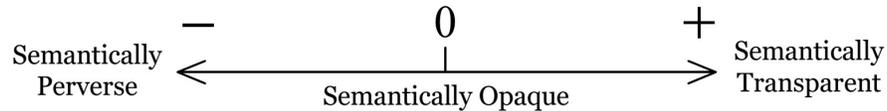


Fig. 6. The semantic transparency and the semantic perversity

4 Discussion and Conclusions

Several works have evaluated the cognitive effectiveness of committee-designed languages, such as UML, BPMN, etc., using theory and empirical evidence from a wide range of fields. The conclusion is that radical improvement is required to improve their cognitive effectiveness. One solution is to involve target end-users as co-designers of these languages rather than as passive consumers as it has been so far. In this paper, we have conducted experiments, for a subset of UML, that have confirmed the importance of involving end-users.

Symbols designed this way increased semantic transparency by almost 300% compared to the standard UML notation. Reducing misinterpretations by end users could therefore lead to significant cost savings: According to [12, 26], design errors are the source of more than half the errors and failure in software development projects [12, 15] and are the most costly errors of all, as their post-implementation correction costs 100 times more than correcting them during the design phase [3].

Our experimental approach is an application of the **crowd-sourcing** for UML visual syntax design. This approach also called peer production or collective intelligence [8] enlists a multitude of humans to help solve a problem. One of the advantages of this approach is that it enlarges/expands the range of notations ideas (i.e. beyond the imagination of the language design team), rather than relying exclusively on experts to design notations, one could follow this approach and take into the ideas of the target audience.

Acknowledgements

This work was supported by the internationalization fund of ETS Montreal and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. [Alexander, C.: Notes on the Synthesis of Form. Harvard Press \(1964\)](#)
2. [Biederman, I.: Recognition-by-components: A Theory of Human Image Understanding. Psychological Review 94, pp. 115–147 \(1987\)](#)
3. [Boehm, B.W.: Software Engineering Economics. Prentice hall edn. \(1981\)](#)
4. [Britton, C., Jones, S.: The Untrained Eye: How Languages for Software Specification Support Understanding in Untrained Users. Hum.-Comput. Interact. 14\(1\), pp. 191–244 \(Mar 1999\)](#)
5. [Caire, P., Genon, N., Heymans, P., Moody, D.: Visual Notation Design 2.0: Towards User Comprehensible Requirements Engineering Notations. In: 21st IEEE International Requirements Engineering Conference \(RE\), pp. 115–124 \(July 2013\)](#)
6. [Campbell, D.T., Stanley, J.C.: Experimental and Quasi-Experimental Designs for Research. Rand McNally College Publishing \(1963\)](#)
7. [Cleveland, W.S., McGill, R.: Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. Journal of the American Statistical Association 79\(387\), pp. 531–554 \(1984\)](#)
8. [Doan, A., Ramakrishnan, R., Halevy, A.Y.: Crowdsourcing Systems on the World-wide Web. Commun. ACM 54\(4\), pp. 86–96 \(Apr 2011\), <http://doi.acm.org/10.1145/1924421.1924442>](#)
9. [Dobing, B., Parsons, J.: How UML is Used. Commun. ACM 49\(5\) \(May 2006\)](#)
10. [El Kouhen, A.: Spécification d'un Métamodèle pour l'Adaptation des Outils UML. Ph.D. thesis, Université de Lille 1 \(2013\)](#)
11. [El Kouhen, A.: Semantic Transparency Experiment Artifacts. \(2014\), <http://www.lifl.fr/~elkouhen/SemanticTransparencyExperiment/artifacts.zip>](#)
12. [Endres, A., Rombach, D.: A Handbook of Soft. and Sys. Engineering: Empirical Observations, Laws and Theories. Addison-Wesley \(2003\)](#)
13. [Genon, N., Heymans, P., Amyot, D.: Analysing the Cognitive Effectiveness of the BPMN 2.0 Visual Notation. In: Software Language Engineering, LNCS, vol. 6563, pp. 377–396. Springer Berlin Heidelberg \(2011\), \[http://dx.doi.org/10.1007/978-3-642-19440-5_25\]\(http://dx.doi.org/10.1007/978-3-642-19440-5_25\)](#)
14. [Goodman, N.: Languages of Art: An Approach to a Theory of Symbols. Hackett, Indianapolis \(June 1976\)](#)
15. [Group, S.: The Chaos Report. \(1994\), \[https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf\]\(https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf\)](#)
16. [Guizzardi, G., Pires, L., van Sinderen, M.: Ontology-based Evaluation and Design of Domain-specific Visual Modeling Languages. In: Advances in Information Systems Development, pp. 217–228. Springer US \(2006\), \[http://dx.doi.org/10.1007/978-0-387-36402-5_19\]\(http://dx.doi.org/10.1007/978-0-387-36402-5_19\)](#)
17. [H., W.C., F., A.H.: Population Stereotypy in Code Design. Organizational Behavior and Human Performance 3\(3\), pp. 310–339 \(1968\)](#)
18. [Heath, C., Heath, D.: Made to Stick: Why Some Ideas Take Hold and Others Come Unstuck. Arrow Books, London, England \(2008\)](#)
19. [Hitchman, S.: The Details of Conceptual Modelling Notations are Important. Communications of the Association for Information Systems 9\(10\) \(2002\)](#)
20. [ISO/IEC: 24744: Metamodel for Development Methodologies. \(2007\)](#)
21. [Jacques, B.: Semiology of Graphics : Diagrams, Networks, Maps. University of Wisconsin Press, Madison, Wisconsin \(1983\)](#)
22. [Jones, S.: Stereotypy in Pictograms of Abstract Concepts. Ergonomics \(1983\)](#)

23. [Larkin, J.H., Simon, H.A.: Why a Diagram Is \(Sometimes\) Worth Ten Thousand Words. *Cognitive Science* 11\(1\), pp. 65–100 \(1987\)](#)
24. [Lee, J.: Design Rationale Systems: Understanding the Issues. *IEEE Expert* \(1997\)](#)
25. [Lohse, G.L.: A Cognitive Model for Understanding Graphical Perception. *Hum.-Comput. Interact.* 8\(4\), pp. 353–388 \(Dec 1993\)](#)
26. [Martin, J.: Information Engineering. Prentice hall. edn. \(1989\)](#)
27. [Masri, K., Parker, D., Gemino, A.: Using Iconic Graphics in Entity-relationship Diagrams. *J. Database Manag.* \(2008\)](#)
28. [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, LNCS*, vol. 5452, pp. 16–34. Springer Berlin Heidelberg \(2009\), \[http://dx.doi.org/10.1007/978-3-642-00434-6_3\]\(http://dx.doi.org/10.1007/978-3-642-00434-6_3\)](#)
29. [Moody, D.L., Heymans, P., Matulevicius, R.: An Evaluation of i* Visual Syntax. In: 17th IEEE Int. Conference on Requirements Engineering \(2009\)](#)
30. [Moody, D.: The “physics” of notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *Software Engineering, IEEE Transactions on* 35\(6\), pp. 756–779 \(Nov 2009\)](#)
31. [Morris, S., Spanoudakis, G.: Uml: An Evaluation of the Visual Syntax of the Language. In: *Proceedings of the 34th Annual Hawaii International Conference on System Sciences. IEEE Computer Society, Washington, DC, USA* \(2001\)](#)
32. [Muller, M.J., Kuhn, S.: Participatory Design. *ACM Com.* \(Jun 1993\)](#)
33. [Nordbotten, J.C., Crosby, M.E.: The Effect of Graphic Style on Data Model Interpretation. *Information Systems Journal* 9\(2\), pp. 139–155 \(1999\)](#)
34. [Novick, L.R.: The Importance of Both Diagrammatic Conventions and Domain-specific Knowledge for Diagram Literacy in Science. In: *Diagrammatic Representation and Inference. Springer* \(2006\)](#)
35. [Organisation\(ISO\), I.S.: Graphical symbols - Test methods - Methods for Testing Comprehensibility \(ISO 9186-1\). Geneva, Switzerland \(2007\)](#)
36. [Palmer, S., Rock, I.: Rethinking Perceptual Organization: The Role of Uniform Connectedness. *Psychonomic Bulletin & Review* 1\(1\), pp. 29–55, <http://dx.doi.org/10.3758/BF03200760>](#)
37. [Petre, M.: Why Looking Isn’t Always Seeing: Readership Skills and Graphical Programming. *ACM Com.* \(Jun 1995\)](#)
38. [Reggio, G., Leotta, M., Ricca, F.: Who Knows/Uses What of the UML: A Personal Opinion Survey. In: *Model-Driven Engineering Languages and Systems, LNCS*, vol. 8767, pp. 149–165. Springer International Publishing](#)
39. [Siau, K.: Informational and Computational Equivalence in Comparing Information Modelling Methods. *Database Management* 15\(1\), pp. 73–86 \(2004\)](#)
40. [Winn, W.: An Account of How Readers Search for Information in Diagrams. *Contemporary Educational Psychology* 18\(2\), pp. 162–185 \(1993\)](#)