Improving Semantic Transparency of Committee-Designed Languages through Crowd-sourcing

Amine El Kouhen, Abdelouahed Gherbi, Cédric Dumoulin

Improving Semantic Transparency of Committee-Designed Languages through Crowd-sourcing

Amine El Kouhen Abdelouahed Gherbi
Department of Software and IT Engineering,
École de Technologie Supérieure, Montréal, QC, Canada
amine.el-kouhen.1@ens.etsmtl.ca /
abdelouahed.gherbi@etsmtl.ca

Cédric Dumoulin
University of Lille, LIFL CNRS UMR 8022,
Cite scientifique - Batiment M3, Villeneuve d'Ascq,
France
cedric.dumoulin@lifl.fr

Abstract

Committee-designed languages such as those of the OMG consortium are widely used in both industry and academia. These languages seem to be used increasingly by users with no technical background for the visualization, documentation and specification of workflows, data and software systems. However, according to several studies on these languages, the used visual notations do not seem to convey any particular semantics and the recognition of such notations is not perceptually immediate. This lack of semantic transparency increases the cognitive load to differentiate concepts from each other and slows down recognition and learning of the language constructs. This paper proposes a process, which leverages the crowd-sourcing to improve the semantic transparency of such languages. We believe that involving end-users in the design process of the languages visual notations should increase the expressiveness of these languages and then their acceptance for a wide range of novice-users.

Categories and Subject Descriptors I.3.6 [*Computing Methodologies*]: Methodology and Techniques
; I.6.5 [*Computing Methodologies*]: Model Development
; D.3.1 [*Software*]: Formal Definitions and Theory

Keywords Visual Languages, UML, Semantic Transparency, Crowdsourcing

1. Introduction

Recently, increasing attention has been paid to the development of visual modeling languages. Among them, we find Committee-designed languages such as those of the OMG consortium (e.g., UML, BPMN...), which are widely accepted as industry standard languages for several purposes.

The visual notations of Committee-designed languages were developed in a bottom-up approach, by reusing and synthesizing existing notations, with a selection of graphical conventions based on expert consensus [21]. We think that this is an inappropriate basis for making visual representation decisions, which they should be based on theory and empirical evidence about cognitive effectiveness [21, 23].

We have made several observations on these visual languages. First of all, the current process for the development of these modeling languages 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 domain experts with their modeling language and its semantic transparency plays a crucial role in its acceptance. Also, the current process is contradicted 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 [16, 28]. A major incentive for using visual notations is the widely-held belief that they convey information more effectively than text, especially to novices [3].

Another observation was done when novice-users manipulate the language for the first time. Committee-designed visual notations can sometimes be very distant from semantic concepts they represent. The lack of semantic transparency increases the cognitive load to differentiate concepts from each other and slows down both the recognition and the learning of the language constructs. That was among the results of several papers [8, 20, 21] where authors evaluate 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 [16].

These assertions and the results of all former evaluations motivate our research to improve this link of visual language effectiveness in practice, particularly for communicating with end-users. This paper proposes a process that can be used to improve these languages usability and effectiveness in practice, especially for communicating with new users. We believe that this kind of languages is an important innovation and our aim is to help remove potential barriers to its adoption and usage by a wide range of novice users.

We ask a rather obvious question: to design notations that are understandable by end-users, why not involve them in the notation design process? If this works in developing software systems (e.g. participatory design [24], user-centred design), why this should not also work in developing diagramming notations?

Thus, we propose in this paper an experiment involving crowd-sourcing in notations design process and we choose UML as a workbench language. For space limitations, we restrict this experimentation to 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 proposals into the design decisions made generally by experts.

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. We raise also the importance to involve end-users actively in the notations design process as *co-designers* rather than as passive *consumers*.

2. Previous Research

One of the main success factors behind the use of a modeling language is its ability to provide to its target users a set of concrete artifacts (visual notations) that can directly express relevant domain abstractions in a concise, complete and unambiguous way [10]. The form of visual representations is known to have an equal if not greater effect on understanding and problem solving performance than their content [21].

Current approaches to designing visual notations (e.g. as followed by OMG technical committees) 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 [27], to have an unambiguous modeling language, its symbols should provide cues to their meaning. Semantically direct representations reduce cognitive load through built-in mnemonics: their meaning can be either perceived directly or easily learned. Such representations speed up recognition and improve intelligibility, especially for novices [3, 19]. According to [4], current visual notation design practice is characterised by:

- An unselfconscious design approach [1]: there is a lack of explicit principles for designing visual notations (i.e, designers rely on instinct, imitation and tradition) [22].
- Lack of design rationale [17]: symbols are typically defined without any explanation or justification as to why they were chosen [12].
- Lack of forms variety: the same or similar symbols (mostly geometrical shapes) are used over and over again for different purposes [25]. Without explicit principles the range of symbols is limited by the imaginations of the design team: this explains why IT diagramming notations use only a fraction of the graphic design space [22].
- Lack of involvement by members of the target audience: visual notation design is conducted exclusively by technical experts, with little or no involvement of end users. For this reason, we propose our experimental process, which uses 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 [22] 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. These nine principles are:

1. Principle of *Visual Expressiveness*: use the full range and capacities of visual variables.
2. Principle of *Semiotic Clarity*: there should be a one-to-one correspondence between elements of the language and graphical symbols.
3. Principle of *Perceptual Discriminability*: different symbols should be clearly distinguishable from each other.
4. Principle of *Semantic Transparency*: use visual representations whose appearances suggest their meaning (evocative). 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 *Complexity Management*: include explicit mechanisms when dealing with complexity.
6. Principle of *Cognitive Integration*: include explicit mechanisms to support the integration of information from different diagrams.
7. Principle of *Dual Coding*: enrich diagrams with textual descriptions.
8. Principle of *Graphic Economy*: the number of different graphical symbols should be cognitively manageable.
9. Principle of *Cognitive Fit*: use different visual dialects when required.

Indeed, these principles have already been used in several works to evaluate and improve other visual comittee-designed languages such as i* [20], BPMN [8] and UML [21]. In the next section, we will reuse the alternative notation proposed by [21], which is based on these nine principles into our proposal.

3. Towards Semantic Transparency for Committee-Designed Languages

Our proposal consists of an experimental process composed of 4 related empirical studies (4 experiments), summarized in figure 1. This process is actually an adaptation of the experiment [4], changing, among other details, some steps and the language for which this experiment is being conducted. As we noted in the introduction, we choose UML as a workbench language and we restrict this experimentation to few elements of this language.The purpose is not to redefine the visual syntax of UML but to show the importance to involve end-users proposals into the design decisions made generally by experts. As shown in the diagram, the results of earlier studies provide inputs for later studies:

1. *Symbolization experiment*: naive participants (i.e, with no previous knowledge on UML) generated symbols for UML concepts.
2. *Stereotyping analysis*: we analysed the results of Experiment 1 and identified the most common symbols produced for each UML concept.
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.
4. *Semantic transparency experiment*: naive users were asked to infer the meaning of symbols from their appearance alone.
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.

The design process combines quantitative and qualitative research methods: studies 1, 2, 3 primarily use qualitative methods, while study 4 uses quantitative methods, although most use a combination of both. Unlike most qualitative studies, the data used is primarily in the form of pictures (drawings) rather than words. The quantitative studies use objective measures of performance (interpretation/recognition accuracy) in combination with psychometric scales (for rating cognitive difficulty of tasks). More detailled data

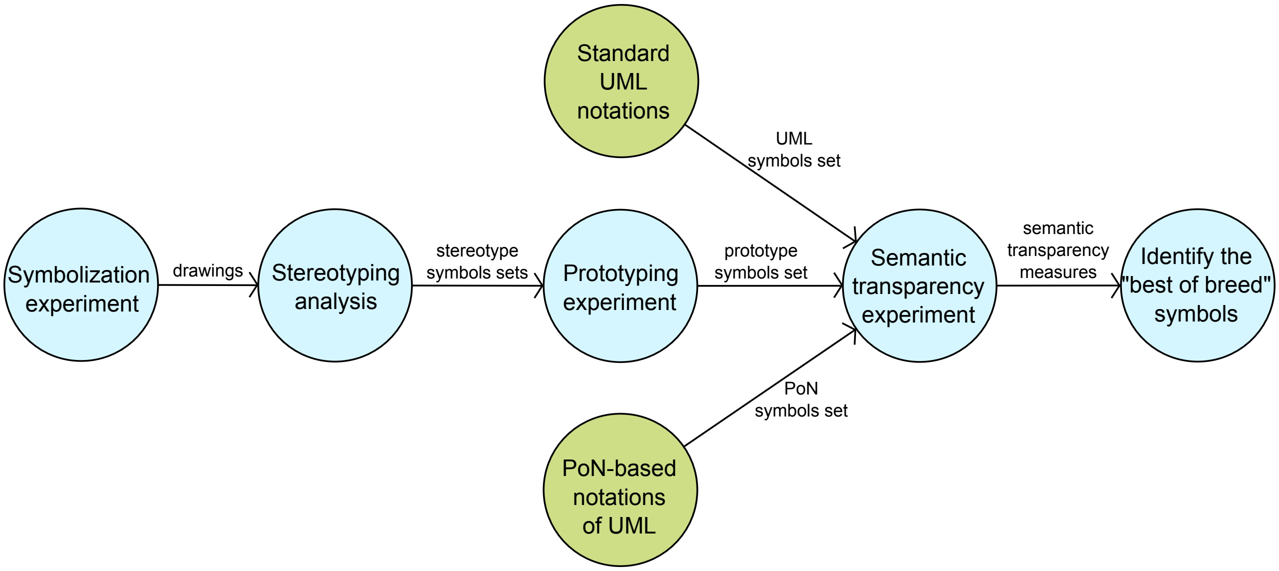


Figure 1. Our experimental process

on participants artifacts, samples, results as well as the coded data set and the statistical scripts are available in [6].

3.1 Symbolization experiment

In this experiment, "naive" participants generated symbols for UML concepts, a task normally reserved for experts.

There were 64 participants (29 females and 35 males) in this experiment, all undergraduate students in computer sciences. 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** [11]).

Each participant was provided with a two-page form that consists of a table of constructs (we chose twelve concepts, which are used frequently 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 in the form. They were instructed to produce drawings that they felt most effectively conveyed the meaning of the construct. It was emphasized that their drawings should be as simple as possible and that artistic ability or quality of drawings was not important: the most important thing was to represent the meaning of the construct as clearly and unambiguously as possible.

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*" [15] 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).

UML Construct	Non-Reponses	Reponse 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%**

Table 1. Response rates for symbolization task

3.2 Stereotyping analysis

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

The analysis was conducted by 2 UML experts (volunteers) plus one of the authors of this paper. "Naive" participants were not required for this study as stereotype identification can be done relatively objectively by looking at similarity of drawings: it is a perceptual (pattern-matching) task rather than a cognitive task so less subject to expertise bias.

The drawings produced in Experiment 1 were used as input for this experiment. Three copies were made of the drawings, so participants could conduct the task independently.

We used the **judges' ranking method** [15], which is a common approach for achieving convergence on a set of categories. In the first round, each judge independently categorized the drawings produced for each concept by sorting them into piles based on their visual and conceptual similarity, following the approach described in [14]. They then compared their categories for each concept, agreed on a common set of categories and how each drawing should be classified. Finally, they selected the most representative drawing from the category with the most representative drawings for each concept (*the stereotypical category*), resulting in 12 stereotypical drawings.

3.3 Prototyping experiment

In this level, naive participants analysed the drawings selected in Experiment 2 and identified the "best" representations for each UML concept. These defined the **prototype symbol set**.

The materials for this experiment were representative drawings for each category identified in the stereotyping analysis. There were 40 naive participants in this experiment, all undergraduate students in computer science from different universities around the world. We used a different sample population from Experiment 1 but drawn from the **same** underlying population. It would not have been appropriate for the authors to perform this analysis as, unlike stereotyping, it is not possible to do this objectively and it would be difficult for us to think like novices. It would also not have been appropriate 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 questionnaire, 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 drawing that most effectively conveyed each concept and to disregard the artistic quality of the drawings. Both the order of the concepts and the position of the drawings on each page were randomized to avoid sequence effects.

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%**

Table 2. Degree of prototypy

The primary outcome 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, a clear prototype emerged: there was a relatively high level of consensus among judgements of prototypicality (43.67% overall). 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 we explained in [21]) and those designed by novices (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 (unselfconscious design).
2. UML notations designed according *Physics of Notations* theory (selfconscious design) called **PoN** in table 3. The details of this notation are available in [21].
3. **Prototype** notations of UML: the best symbols produced by novices as judged by other novices.

We conducted this experiment using a multi-choice questionnaire. One symbol was displayed at the top of each page (representing the **stimulus**) and the complete set of UML constructs and definitions displayed in a table below (representing the possible responses). Participants were asked to indicate which construct they thought most likely corresponded to the symbol. In each page, there was one correct or target response and 11 incorrect or distractor responses. Both the order in which the stimuli (symbols) were presented (i.e., order of pages) and the order in which the responses (concepts) were listed on each page were randomized to avoid sequence effects.

Participants were randomly assigned to experimental groups. They were instructed to work alone and not discuss their answers with any other participants. They were asked to answer each question in order and not to review previous answers. They were told to choose one and only one concept for each symbol presented but that each choice was independent: they could choose the same concept in response to multiple symbols. The purpose of this was to reduce the difficulty to remember previous choices (cognitive difficulty) and to choose directly about what symbols meant.

Hypothesis : Given that sign production studies consistently show that symbols designed by novices are more accurately interpreted than those designed by experts, we predicted that the *prototype* symbol set would outperform both of the standard *UML* and *PoN* symbol sets.

3.5 Identify *best of breed* symbols

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

The traditional way of measuring comprehensibility of graphical symbols is by measuring *hit rates* (i.e., percentage of correct responses). The ISO standard for testing graphical symbols [26] defines 67% as the hit rate required for acceptance of public information and safety symbols. 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-reponse in experiment 1, a high level of consensus (degree of prototypy) in experiment 2 and exceed the ISO threshold (67%) in experiment 3. More detailed results as well as the coded data set and the statistical scripts are available in [6].

The differences between groups are visually confirmed by the box and whisker plot in Figure 2. 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.

We can observe that the prototype symbol set exceeds largely the ISO threshold for comprehensibility of symbols.

Using explicit design principles (selfconscious design) significantly improves semantic transparency (supported by our hypoth-

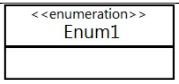
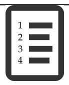
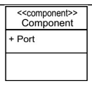
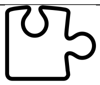
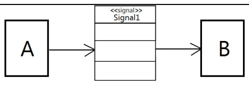
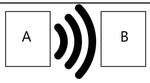
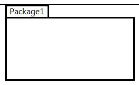
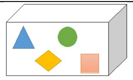
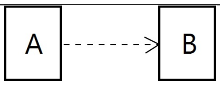
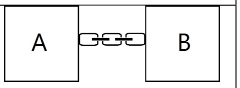
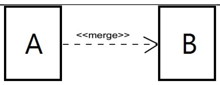
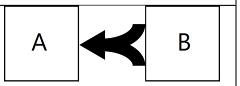
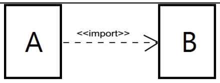
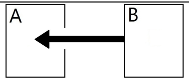
UML Concepts	UML Standard symbols	Experiment 3 : Semantic Transparency			*Best of breed* symbol
		Standard	PoN Notations	Prototype	
Enumeration		30%	45%	77.5%	
Component		22.5%	30%	70%	
Signal		5%	35%	87.5%	
Package		22.5%	20%	87.5%	
Dependency		12.5%	20%	75%	
Merge		27.5%	37.5%	72.5%	
Import		32.5%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

Table 3. "*Best of Breed*" symbols

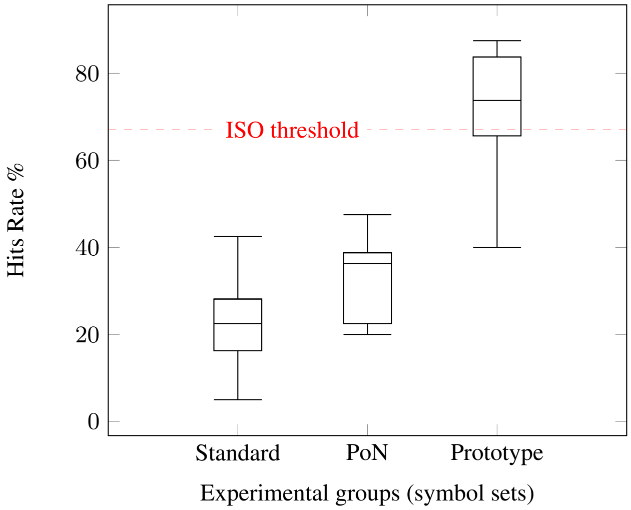


Figure 2. Differences in hite rate between experimental groups

esis), 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.

In this experiment, our *a priori* hypothesis was confirmed. We find that symbols proposed and chosen by novices 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.

4. Discussion and Conclusions

Several works have evaluated the cognitive effectiveness of the committee-designed languages (e.g., UML, BPMN...) using theory and empirical evidence from a wide range of fields. The conclusion is that radical improvement is required to these languages visual notations to make them cognitively effective. For that, we involved target audience as co-designers of these languages notations rather than as passive consumers as in current approaches. Thus, we choose UML as a workbench language. For space limitations, we restrict this experimentation to few elements of UML visual syntax. The purpose is not to redefine the visual syntax of UML but to show the importance to involve end-users proposals into the design decisions made generally by experts.

We think that the next release of UML represents a good opportunity to redesign the visual notations in a cognitively optimal manner. The existing UML notations have been developed in a bottom up manner, by reusing and synthesising existing notations. Ideally, visual representations should be designed in a top-down manner based on a thorough analysis of the information content to be conveyed: form should follow the content [13]. Also, the target audience should be actively involved in the visual notations design (participative design). We believe that to communicate effectively with an audience, we need to speak its language. Giving end-users the ability to design the visual language that we use to communicate with them, allows us to propose more cognitively effective languages.

The approach described in this paper represents an application of the **crowdsourcing** in UML visual notations design. This approach also called peer production or collective intelligence [5] 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 technical experts to design notations, groups like OMG could follow this approach and invoke the ideas of the target audience.

Symbols designed this way increased semantic transparency by almost 300% compared to the standard UML notation. Design errors are the source of more than half the errors in software development [7, 18], are the most common cause of failure of software development projects [7, 9] and are the most costly errors of all: it is more than 100 times more costly to correct a defect post-implementation than to correct it during the design phase [2]. Reducing interpretation errors by end users could therefore lead to significant cost savings and productivity improvements in software development.

Of course, such change will need to be handled carefully as practitioners familiar with the existing notation will resist radical change. However, they may be more open to changes if they have a clear design rationale grounded on theory and empirical evidence, something which is currently lacking from the Committee-designed languages.

Another goal of this paper is to draw attention to the importance of visual syntax in the design of Committee-designed languages. Visual syntax is an important factor of the cognitive effectiveness of software engineering modeling languages, and a determinant way to a better inference/recall of their semantics.

Acknowledgements

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

References

[1] Christopher Alexander. *Notes on the synthesis of form*. Harvard Press, 1964.

[2] Barry W. Boehm. *Software Engineering Economics*. Prentice hall edition, 1981.

[3] Carol Britton and Sara Jones. The untrained eye: How languages for software specification support understanding in untrained users. *Hum.-Comput. Interact.*, 14(1):191–244, March 1999.

[4] P. Caire, N. Genon, P. Heymans, and D.L. Moody. Visual notation design 2.0: Towards user comprehensible requirements engineering notations. In *21st IEEE International Requirements Engineering Conference (RE)*, pages 115–124, July 2013.

[5] Anhai Doan, Raghu Ramakrishnan, and Alon Y. Halevy. Crowdsourcing systems on the world-wide web. *Commun. ACM*, 54(4):86–96, April 2011.

[6] Amine El Kouhen and Abdelouahed Gherbi. Semantic transparency experiment artifacts, 2014. `http://www.lifl.fr/~elkouhen/SemanticTransparencyExperiment/artifacts.zip`.

[7] A. Endres and D. Rombach. *A Handbook of Soft. and Sys. Engineering: Empirical Observations, Laws and Theories*. Addison-Wesley, illustrated edition, 2003.

[8] Nicolas Genon, Patrick Heymans, and Daniel Amyot. Analysing the cognitive effectiveness of the bpmn 2.0 visual notation. In Brian Malloy, Steffen Staab, and Mark van den Brand, editors, *Software Language Engineering*, volume 6563 of *Lecture Notes in Computer Science*, pages 377–396. Springer Berlin Heidelberg, 2011.

[9] Standish Group. The chaos report., 1994. `http://www.standishgroup.com/sample_research/chaos_1994_1.php`.

[10] Giancarlo Guizzardi, LuisFerreira Pires, and Marten van Sinderen. Ontology-based evaluation and design of domain-specific visual modeling languages. In Anders G. Nilsson, Remigijus Gustas, Wita Wojtkowski, W. Gregory Wojtkowski, Stanislaw Wrycza, and Joze Zupancic, editors, *Advances in Information Systems Development*, pages 217–228. Springer US, 2006.

[11] C. Heath and D. Heath. *Made to Stick: Why Some Ideas Take Hold and Others Come Unstuck*. Arrow Books, London, England, 2008.

[12] Steve Hitchman. The details of conceptual modelling notations are important. *Communications of the Association for Information Systems*, 9(10), 2002.

[13] Jeff Howe. The rise of crowdsourcing. *Wired Magazine*, 14(6), 06 2006.

[14] William C. Howell and Alfred H. Fuchs. Population stereotypy in code design. *Organizational Behavior and Human Performance*, 3(3):310–339, 1968.

[15] Sheila Jones. Stereotypy in pictograms of abstract concepts. *Ergonomics*, 1983.

[16] Jill H. Larkin and Herbert A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65 – 100, 1987.

[17] Jintae Lee. Design rationale systems: understanding the issues. *IEEE Expert*, 97.

[18] J. Martin. *Information Engineering*. Prentice hall. edition, 1989.

[19] Kamal Masri, Drew Parker, and Andrew Gemino. Using iconic graphics in entity-relationship diagrams. *J. Database Manag.*, 2008.

[20] D. L. Moody, P. Heymans, and R. Matulevicius. An evaluation of i* visual syntax. In *17th IEEE Int. Conference on Requirements Engineering*, 2009.

[21] Daniel Moody and Jos van Hillegersberg. Evaluating the visual syntax of uml: An analysis of the cognitive effectiveness of the uml family of diagrams. In Dragan Gasevic, Ralf Lammel, and Eric Van Wyk, editors, *Software Language Engineering*, volume 5452 of *Lecture Notes in Computer Science*, pages 16–34. Springer Berlin Heidelberg, 2009.

[22] D.L. Moody. The “physics” of notations: Toward a scientific basis for constructing visual notations in software engineering. *Software Engineering, IEEE Transactions on*, 35(6):756–779, Nov 2009.

[23] S. Morris and G. Spanoudakis. Uml: An evaluation of the visual syntax of the language. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, Washington, DC, USA, 2001. IEEE Computer Society.

[24] Michael J. Muller and Sarah Kuhn. Participatory design. *ACM Com.*, June 1993.

[25] Laura R. Novick. The importance of both diagrammatic conventions and domain-specific knowledge for diagram literacy in science. In *Diagrammatic Representation and Inference*. Springer, 2006.

[26] International Standards Organisation(ISO). *Graphical symbols - Test methods - Methods for testing comprehensibility (ISO 9186-1)*. Geneva, Switzerland, 2007.

[27] Marian Petre. Why looking isn’t always seeing: Readership skills and graphical programming. *ACM Com.*, June 1995.

[28] K. Siau. Informational and computational equivalence in comparing information modelling methods. *Database Management*, 15(1):73–86, 2004.