Requirements Eng (2009) 14:15–26
DOI 10.1007/s00766-008-0071-7

ORIGINAL ARTICLE

A semiotic analysis of unified modeling language graphical notations

Keng Siau · Yuhong Tian

Received: 25 May 2006 / Accepted: 2 December 2008 / Published online: 8 January 2009

Abstract Unified modeling language (UML) is the standard modeling language for object-oriented system development. Despite its status as a standard, UML has a fuzzy formal specification and a weak theoretical foundation. Semiotics, the study of signs, provides a good theoretical foundation for UML research because graphical notations (or visual signs) of UML are subjected to the principles of signs. In our research, we use semiotics to study the effectiveness of graphical notations in UML. We hypothesized that the use of iconic signs as UML graphical notations leads to representation that is more accurately interpreted and that arouses fewer connotations than the use of symbolic signs. An open-ended survey was used to test these hypotheses. The results support our propositions that iconic UML graphical notations are more accurately interpreted by subjects and that the number of connotations is lower for iconic UML graphical notations than for symbolic UML graphical notations. The results have both theoretical and practical significance. This study illustrates the usefulness of using semiotics as a theoretical underpinning in analyzing, evaluating, and comparing graphical notations for modeling constructs. The results of this research also suggest ways and means of enhancing the graphical notations of UML modeling constructs.

Keywords Unified modeling language · Semiotics · Modeling methods · Graphical notations · Systems analysis and design

K. Siau (✉) · Y. Tian
Department of Management,
209 College of Business Administration,
University of Nebraska-Lincoln,
Lincoln, NE 68588-0491, USA
e-mail: ksiau@unl.edu

1 Introduction

Object-oriented modeling language is converging into a single standard—the unified modeling language (UML). This unification has prevented system analysts from getting lost in the jungle of modeling methods. It has also enabled the modeling community to focus its effort on improving and enhancing one language. However, there is still much work to be done in UML development. UML, together with its foundation, object-oriented design methodology, is constantly evolving. As we gain more knowledge of the modeling process and understand more about human cognition, we can continuously improve the modeling language to enhance its clarity and usefulness [37]. In this paper, UML is also used as an example to demonstrate how the knowledge we gain from semiotics can help in the evolution of modeling language.

Some prior research on UML focused on adding extended features to make UML more flexible and applicable (e.g., see [1, 13, 17, 27, 34]). However, one of the criticisms of UML is that it is large and complex, and this can be daunting to novice users [12, 19, 41, 42, 57]. Adding more features and constructs to UML may not be the best way to improve the modeling language. Empirical analysis and evaluation of modeling methods and languages will help us identify ways to make the modeling process easier and more end-user friendly [15, 25, 38, 39, 47–50, 52].

Semiotics is the study of signs. It is rooted in linguistics and has been applied to a wide range of communication forms such as advertising, television, cinema, and politics. Information system modeling language, which extensively uses visual signs (i.e., UML graphical notations) as basic language constructs, is subjected to the principles of semiotics—the study of signs. In this research, we use semiotics to help us analyze and evaluate UML graphical notations, and to suggest ways to enhance the clarity of UML graphical notations.

The rest of the paper is organized as follows: We introduced prior research on modeling language and UML complexity in the next section. Then we reviewed models and principles in semiotics, which form the conceptual and theoretical basis of our research. The remaining parts of the paper are the research model, research method and procedure, research results, and discussions. The final section concludes the paper.

2 Literature review

Compared to other modeling methods and languages, including entity-relationship modeling, Business process engineering (BPR), flow charting, and state-driven languages, UML provides improved expressiveness and holistic integrity [31], according to UML supporters. Proponents of UML (e.g., [7]) argue that it is more expressive yet cleaner and more uniform than Booch, OMT, OOSE, and other methods because it removes unnecessary differences in notations and terminologies that obscure the underlying similarities of most of these methods. On the other hand, studies have shown that UML is ambiguous, inadequate, and cognitively misdirected [40, 43, 51].

As a communication tool for the stakeholders in software development, including end-users, system designers, and developers, a modeling language should be easy to use and comprehend. A modeling language should be easy to model (construction) and easy to understand (interpretation) [37, 46, 48]. The complexity of a modeling language has significant practical implications [14].

One way of comparing modeling languages is to create meta-models of the modeling languages [1, 53]. Metamodeling makes different constructs in modeling languages comparable to each other. Another approach is to use structure matrices [35, 40]. Structure metrics provide an objective measure of modeling language complexity [44]. Another stream of research argues that modeling and model interpretation are mental processes. The explanation of these processes may require the use of psychology and cognitive theories (e.g., see [6, 37, 45]). GOMS is a cognitive theory that describes the procedures required for accomplishing a general set of tasks in standard statements. GOMS was used to analyze the complexity of UML modeling and interpreting tasks [48]. Further, the level of complexity of a modeling language is based on the perception of the stakeholders. Empirical researches on UML complexity include users' views of UML notational elements [37] and users' views of practical complexity versus theoretical complexity of UML [14]. Table 1 summarizes some of the existing work in the area.

Wand and Weber [58] introduced the ontology defined by Bunge into information modeling. The resulting framework is referred to as BWW ontology. According to the BWW ontology, information modeling method construct must have a counterpart in the real world. Construct excess (when the construct of modeling method does not have a real world counterpart) or construct deficiency (when no modeling construct exists for a corresponding real world object) are two situations that a good modeling method must avoid. However, the real world is so complex that a modeling method may not be able to model it exactly while still being reasonably wieldy. Further, a model should be a simplification of reality. Thus, an important design issue of a modeling method is to determine the appropriate level of real world complexity to capture. In other words, the modeling constructs of a modeling method determine the level of abstraction that can be captured by the modeling method. For a given modeling method, the modeler can further determine the level of abstraction she/he wishes to capture of the real world.

In most cases, the reason a modeling method becomes very complicated is it tries to incorporate as many constructs as possible so that it can model the domain complexity well. In doing so, the complexity of the real world will make a modeling method complex. Modeling method developers face a dilemma between expressive power and simplicity because simplified modeling methods are at risk of failing to model the complex real world due to insufficient constructs. To compensate for that, simplified modeling method tends to assign many denotations to one single construct (construct overloading). This creates problems for users to accurately interpret the modeling construct as the construct can mean a number of things in the real world. On the other hand, a complex modeling method with corresponding number of constructs to the real world, as recommended by the BWW ontology [58, 59], is able to model the real world accurately. However, it will inevitably need to sacrifice simplicity to include many modeling elements and notational constructs in the modeling method. The resulting modeling method can become extremely complex and unwieldy. Complexity is a known issue with UML even though UML may not have all the corresponding real world constructs. The excessive number of constructs is one of the reasons for UML's structural complexity, as revealed by the metrics analysis of UML conducted by Siau and Cao [40] and cognitive analysis done by Siau and Loo [43].

Prior research on UML and modeling method complexity only revealed the facts of complexity. They seldom explained why certain methods or constructs are more complex to users. They also did not provide constructive

Table 1 Studies on complexity of modeling methods

Research	Question	Method	Proposition/results
Song and Osterweil [53]	Create a meta-model to objectively and systematically measure modeling methods	Meta-model	Meta-model (base framework) was constructed by abstracting the function framework and type framework of the many different modeling methods. The meta-model was applied to evaluate a number of system design methods
Alemán and Álvarez [1]	Create a meta-model for UML	Meta-model	A meta-model of UML was proposed to seamlessly formalize the UML semantics
Rossi and Brinkkemper [35]	Develop metrics to measure and compare different system development methods and techniques	Structure metrics	A series of metrics was created to measure a modeling method based on OPRR (Object, Property, Relationship, Role). The metrics was applied to compare different modeling methods
Siau and Cao [40]	Apply Rossi and Brinkkemper's metrics in UML evaluation	Structure metrics	Compared with other OO techniques, the nine UML diagrams are not distinctly more complex when taken individually. But when taken as a whole, UML is more complex than other OO methods
Siau and Tian [48, 49]	Evaluate the complexity of UML modeling task and interpreting task using GOMS analysis	Cognitive (GOMS model)	The nine UML diagrams are on different level of complexity for modeling and interpreting tasks. Class diagram, which involves lots of mental operators, is the most complex diagram. The fact that some diagrams have many same or similar task/subtask alleviates the complexity of UML
Shen and Siau [36]	Evaluate user's view of problems, difficulties and concerns in drawing and interpreting UML notational elements	Empirical (Concept mapping)	Ontological discrepancies exist in UML notational elements. For example, minute differences in the drawing of lines (solid versus dotted) and arrowheads (filled versus stick) can mean different constructs
Erickson and Siau [14]	Define and measure the theoretical and practical complexity of UML	Empirical (Delphi)	With the UML kernel restricted to four diagrams (Class, Use Case, Sequence, and Statechart diagrams), the practical complexity of UML decreased significantly when compared to the overall theoretical complexity of UML

suggestions to reduce complexity while preserving the expressive power of a modeling construct.

Instead of focusing on identifying modeling constructs and their real world counterparts and vice versa, we examined the modeling constructs and their visual representations, i.e., the graphical notations of the modeling constructs, in this research. The objective is to create UML graphical notations that are:

- Intuitive: making them easier to understand, memorize, and use, thus relieving cognitive overload and allowing the modeler and interpreter to allocate more cognitive resources to the model itself rather than to recalling and memorizing graphical notations;
- Unambiguous: so that one graphical notation means the same modeling construct to different users.

The knowledge we gain from the study of signs can help us achieve these goals. Semiotics can not only explain why certain constructs are more complex than others and what kind of visual signs are more intuitive, but it can also tell us how to improve the graphical notations. In addition, previous research in modeling methods is typically weak in theoretical underpinning and foundations. Semiotics provides a solid theoretical foundation for modeling method research.

3 Theoretical foundation: semiotics

Semiotics is the study of signs or the general theory of representation [28, 30]. It concerns the properties of things in their capacity as signs. Signs take the form of words, images, sounds, odors, flavors, acts or objects. Semiotics, rather than being considered as an independent academic discipline, represents a range of studies in art, literature, anthropology, and mass media.

Use of semiotics in IT is not a new phenomenon. Human computer interaction (HCI) has vigorously applied principles of semiotics for decades [2, 5, 10]. Semiotics in HCI mainly focuses on analyzing the visual signs. For example, icons in GUI design utilize the semiotics approach. Nadin pointed out that HCI is semiotics applied: “If there is a science of interface (computing interface or any other kind), then this science is semiotics” [29]. Other areas that researchers applied semiotics include semiotics framework of information evolution [11, 20], and semiotics framework of information system classification and development [4].

Some researchers focused on the social dimension of semiotics and its role in information system development, as demonstrated in the ontology chart proposed by Stamper et al. [55] for user requirements acquisition and semiotic analysis applied in identifying communicating roles in agent-based information system modeling [3]. Organizational semiotics [21, 23, 24] was introduced to represent the schools of thoughts that understand business organizations as "systems where signs are created, transmitted, and consumed for business purpose" [22] (p. 40). As Stamper [55] (p. 350) pointed out, "Business is getting things done by using information. All information is 'carried' by signs…". Depending on their focus, researchers in organizational semiotics use different approaches such as the system-oriented approach, behavior-oriented approach, and knowledge-oriented approach. Among them were Mehler and Clarke [26] who used the system-oriented approach to analyze the effectiveness of hypertext. In spite of the efficacy in applying semiotics to information modeling and organizational contexts, there is no prior application of semiotics in modeling language evaluation. We believe that modeling language is subjected to the general principles of knowledge of signs (i.e., semiotics) and semiotics can contribute much to systems analysis and design research.

3.1 Two models

There are two dominant models of what constitutes a sign in semiotics.

(1) Saussure's dyadic model:

- The '*signifier*': the form which the sign takes;
- The '*signified*': the concept which the sign represents.

The relationship between the signifier and the signified is 'signification'. Following this dyadic model of semiotics, the relationship between the signified and signifier is not directly made, but rather is formed through the process of "sense making" or "interpretation". It is only through the process of "sense making" that the relationship between signs and its representation can be established. This is what the triadic semiotics model tried to reveal later.

(2) Peirce's triadic model:

- The *Representamen*: the form which the sign takes (not necessarily material);
- The *Interpretant*: not an interpreter but rather the sense made of the sign;
- The *Object*: the thing to which the sign refers.

For example, a rectangle with a name in UML represents a *class*. The rectangle itself is a "signifier" or "representamen" and the construct of *class* that it represents is the "signified" or "object". The "Interpretant" states that the rectangle only represents a *class* when the interpreter understands it to be a "class" in his/her eyes. Variants of Peirce's triad are often presented as '*the semiotic triangle*'. Figure 1 shows a frequently referenced version, which changes only the unfamiliar Peircean terms [8]. Note that the link between "sign vehicle" and "referent" is a dotted line whereas the links between "sign vehicle" and "sense", and between "sense" and "referent" are solid lines. This indicates that the relationship between "sign vehicle" and its "referent" can only be established through the process of "sense making".

In the case of our research on modeling methods, our goal is to make graphical notations (sign vehicles) of a modeling method effectively and accurately represent the referent in the eyes of the interpreter.

3.2 Modes of signs

Most semiotics researchers stressed that signs differ in how arbitrary/conventional (or in contrast, "transparent") they are, or how close the relationship between the signifier and the signified is. There are three modes of signs [8]:

- *Symbol/symbolic:* a mode in which the signifier does *not* resemble the signified but which is fundamentally arbitrary or purely conventional—so that the relationship must be learnt: e.g., language in general, traffic lights, etc.
- *Icon/iconic:* a mode in which the signifier is perceived as resembling or imitating the signified (recognizable look, sound, feel, taste or smell) or being similar by possessing some of its qualities: e.g., a portrait, a cartoon, etc.
- *Index/indexical:* a mode in which the signifier is *not arbitrary* but is *directly connected* in some way, either physically or causally to the signified. This link can be

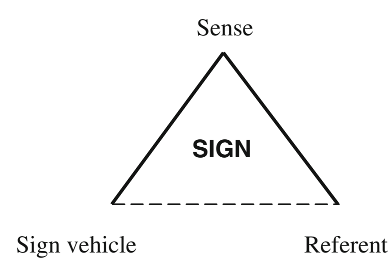


Fig. 1 Peire's semiotic triad model

observed or inferred: e.g., "natural signs" such as smoke or thunder, and personal "trademarks" such as handwriting. Indexes lie between symbols and icons.

These categories are not mutually exclusive. The classification is more of a continuum between extreme iconic and extreme symbolic. In addition, a sign could very well fall into all three categories at the same time. For example, a part of the sign could be symbolic while the other part could be iconic or indexical. A good example of this is in Chinese characters. One major category of Chinese characters is called "shape and sound." Half of the character is iconic, which means it looks like the thing (referent) it represents, while the other half of the character is symbolic, which means that it has no meaningful relationship to the referent but arbitrarily symbolizes the pronunciation of the character.

The three modes of signs are different in how arbitrary/conventional (or by contrast 'transparent') they are. Convention is the social dimension of signs. It is the agreement amongst the users about the appropriate uses of and responses to a sign [16]. As Chandler [8] pointed out:

> "The terms 'motivation' (from Saussure) and 'constraint' are sometimes used to describe the extent to which the signified determines the signifier. The more a signifier is constrained by the signified, the more 'motivated' the sign is: iconic signs are highly motivated; symbolic signs are unmotivated. The less motivated the sign, the more learning of an agreed convention is required".

Obviously, the less arbitrary the sign, the more constrained the signifier is by the signified. For example, a picture is constrained by the objective inside the picture because the picture must resemble the objective it represents. Thus, less convention is involved in the sign. When we say a sign is conventional, we mean that it is arbitrary because many conventions are involved in the sign and one needs to learn those conventions to understand the sign. An example is a word in a foreign language. The use of the term "degree" of signifier constrained by the signified indicates that the classification of symbolic, indexical, and iconic signs is not clear-cut. The degree of convention involved in a sign and the level of a sign's motivation are continuous rather than discrete.

In a system analysis and design method, a construct's graphical notation that is highly motivated involves less convention, is less arbitrary, and demands less of the interpreter's learning effort. An analysis of a modeling method's graphical notation can help to determine the ease of use of the modeling method. In this research, we focused on the iconic and symbolic graphical notations of UML as indexical graphical notations lie between these two ends.

3.3 Denotation versus connotation

Due to the importance of "sense making" in the triadic model, the understanding of denotation and connotations of a sign is emphasized in semiotics. Denotation is the definitional, literal or obvious meaning of a sign. Connotations are any other meanings associated with the sign or the socio-cultural and personal associations (ideological, emotional, etc.) of the sign. These are typically related to the interpreter's class, age, gender, ethnicity, etc. The phrases "IRA terrorists" and "IRA freedom fighters" denote the same people, but they connote quite different meanings.

Condon et al. [9] studied the connotations of the "Save as…" command in Microsoft Word and found variations of connotations and inconsistent denotations of the term. Similar research can be done on modeling methods by studying the connotations and denotations of graphical notations of a modeling method. For example, UML graphical notations with too many connotations may not be good notations as that can be focusing to users. Similarly, if end-users tend to inconsistently denote one UML notation, that UML notation may be problematic. In other words, a good graphical notation should enable consistent and accurate denotations, and arouse few connotations.

4 Research model

Unified modeling language is a modeling language using graphical notations as signs. The dyadic (2 factors) and triadic (3 factors) models of UML graphical notations are shown in Figs. 2 and 3.

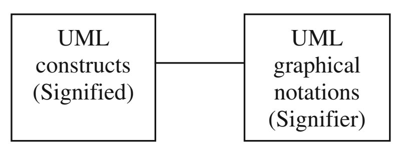


Fig. 2 Diatic model of UML graphical notations

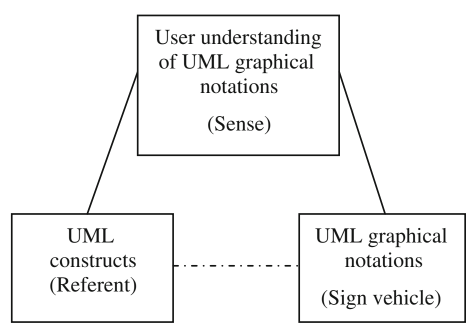


Fig. 3 Triadic model of UML graphical notations

As mentioned before, the three modes of signs differ in how close the relationship between the signifier and the signified is, or how the signifier is constrained by the signified, i.e., they differ in how arbitrary/conventional they are. It is hard for a user to learn and understand an arbitrary/conventional graphical notation because it involves more agreed conventions.

As Chandler [8] pointed out, purely symbolic signs are unmotivated. Symbolic signs combined with some iconic features are more motivated. The less motivated the sign is, the more learning of an agreed convention it requires. Iconic and indexical signs, especially the iconic signs, are more motivated and therefore are easier to recognize and learn, thus leading to more accurate representations. As discussed earlier, denotation is the definitional meaning of a sign, whereas connotation means any other meanings associated with the sign. We can use the user's interpretation of the denotation and connotation of the sign (specifically, UML graphical notations) to study how accurately, the sign represents the referent. UML graphical notations that are more motivated will demand less learning effort and represent the referent more accurately. Consequently, we hypothesized the following:

1. The denotation of iconic UML graphical notations will be more consistent with the signs' referent than symbolic UML graphical notations.
2. The number of connotations connected to iconic UML graphical notations will be less than symbolic graphical notations.

5 Research method and procedure

An open-ended questionnaire in the form of a table was used to solicit users' views of the denotations and connotations of UML graphical notations. Both symbolic and iconic graphical notations were used.

5.1 Classification of UML notations

To create the questionnaire, the first step was to select appropriate notations that represent symbolic signs and iconic signs. According to OMG [31], UML contains four types of graphical constructs: icons, two-dimensional symbols, paths, and strings. Although UML specification uses the words "icons" and "symbols", these terms are not rigorously defined as they are in the context of semiotics because they are used casually and interchangeably in the UML specification. As discussed earlier, no matter how iconic a sign is, some levels of conventions are always involved. In UML, a graphical notation may never be purely iconic. The classification of UML notation in terms of mode of sign may be difficult and subjective. The example of mode analysis given by Underwood [56] provided us a good qualitative analysis method to classify UML notations into appropriate type of signs. Peirce's second trichotomy, "What is the relation between the Sign and its Object?", is another useful method. The frequencies of different UML notations actually used by practitioners vary widely [15]. It would be less meaningful if the notations we chose for this study were seldom used in practice. Therefore, some frequently and commonly used UML graphical notations are selected.

5.2 Eliciting denotations and connotations

Denotation, by definition, is the designed referent of UML sign. In this research, we study the denotations and connotations of some common UML graphical notations. One issue is the elicitation of denotations and connotations. Osgood et al. [32] proposed a technique called the *semantic differential* for the systematic mapping of *connotations* (or "affective meanings"). The technique involves a pencil-and-paper test in which people are asked to give their impressionistic responses to a particular object, state or event by indicating specific positions in relation to at least nine pairs of bipolar adjectives on a scale of one to seven.

Condon et al. [9] used a short "what for" interview on users to uncover the denotations and connotations of the sign for the "save as…" command. The researchers continuously ask "what for" until the answers formed a closed loop or the interviewee felt that the questions were unanswerable. A content analysis technique is then used to break down the responses into separate significations.

However, Condon et al. [9] pointed out that on-the-spot interview method faces two problems: (1) the interviewee may have difficulty coming up with the right word, or (2) have difficulty coming up with the right word at the exact moment. To solve these problems, we chose to use an open-ended question survey to elicit connotations. According to Galloway [18], the advantages of open-ended questions include greater freedom of expression, no bias due to limited response ranges, and respondents' ability to qualify their answers.

5.3 Research design and procedure

The UML notations were classified into two groups: one group being more iconic, and the other being more symbolic. For each notation, subjects were asked, in the first column, to write down the UML constructs they thought the notations should represent based on what they have learned (denotations). In the second column, they were asked to write down what the notations could represent according to how they feel (connotations).

The participants of the research were graduate students majoring in management information systems, computer science or software engineering. Subjects were also required to have taken a course on UML, and to have experience working with UML.

After the data were collected, a content analysis technique was used to break the responses down into separate connotations. The total number of connotations for each UML graphical notations was summarized. Analysis was done to see whether there is a difference in accurate representation (denotation) between the symbolic and the iconic UML graphical notations, and whether the level of connotation is contingent on the two types of UML notations. We used content analysis rather than statistics test because of the limited number of UML notations available for our study. For our study, we had to choose only those UML graphical notations on the two extreme points of the iconic versus symbolic continuum, and these UML constructs should be commonly used.

In the second part of the research, we invited subjects to create some new graphical notations to "better represent" certain UML constructs.

6 Research results

In our study, we chose eight symbolic UML graphical notations and eight iconic UML graphical notations. We selected 20 college students who were familiar with UML to participate in the study. Participants were neither allowed to use reference books nor discuss their answers before submitting the questionnaires. The scores of the participants on each of the 16 UML graphical notations were averaged to remove individual differences. An example of iconic UML graphical notation would be the stick-man representing an actor. An example of symbolic UML graphical notation will be a rectangle representing a class or object. Tables 2 and 3 depict the eight symbolic UML graphical notations and the eight iconic UML graphical notations.

As can be seen from Tables 2 and 3, for denotations, iconic graphical notations have an average of 51.25% correct interpretations compared to 42.5% for symbolic graphical notations. When the worst (based on the number of correct interpretations) symbolic graphical notation and the worst iconic graphical notations are excluded from the computation, the iconic graphical notations have an average of 55.71% of correct interpretations compared to 47.14% for symbolic graphical notations. As such, the results support our first hypothesis, which states that the denotation of iconic UML graphical notations will be more consistent with the signs' referent than symbolic UML graphical notations.

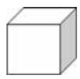
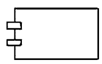
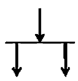
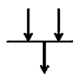
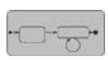
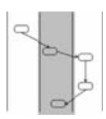
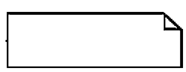
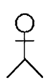
From Tables 2 and 3, the results show that the number of connotations is lower for iconic than symbolic UML graphical notations (i.e., 2.88 vs. 4.88). When the worst symbolic and iconic graphical notations are excluded from the computation, the number of connotations is 2.71 for iconic graphical notations and 5.14 for symbolic graphical notations. The most iconic UML graphical notation, the stick man, had almost no connotations. The results support our second hypothesis, which states that the number of connotations connected to iconic UML graphical notations is less than symbolic UML graphical notations. The low number of connotations for iconic UML graphical notations is probably a main reason for the higher accuracy in interpreting iconic UML graphical notations.

The second part of the study invited subjects to create new graphical notations that they thought could better represent UML constructs. We found the results very revealing and encouraging. The alternatives provided by the subjects made the graphical notations either more iconic or more distinct compared to other similar notations. Some examples of these suggested notations are presented in the Table 4. The examples show that we do not necessarily have to add too much visual details to a notation to make it more iconic. Although some proposed notations only slightly modified the original graphical notations, they showed that it was not difficult to enhance current UML notations to increase the signs' representation without too much visual clutter. The suggested new notations are not very visually cluttered, but they are more representative of their corresponding constructs, or in other words, closer to iconic signs. These propositions are very helpful for future UML versions and new modeling methods development.

7 Discussions

Our results show that iconic graphical notations are more accurately interpreted and invoke fewer connotations. The lower number of connotations restricts the number of possible interpretations and helps to prevent misinterpretation. Thus, one way to make UML easier for users is to use more iconic graphical notations. The second part of our study shows that this can be done quite easily. Adding just a small bar or a little square, or just slightly changing the shape could make a big difference. This can be seen from the alternative UML graphical notations created in our study (see Table 4). Admittedly, visual clutter may be a problem if signs are too iconic. However, as we discussed above, the classifications of signs are on a continuum. We are not proposing to include pictures into UML or use rich pictures as suggested by Peter Checkland in the Soft

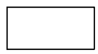
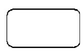
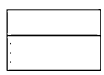
Table 2 Research results for iconic graphical notations

UML graphical notation	Type of graphical notation	Denotations (percentage of subjects who provided correct answers) (%)	Number of connotations
Node: 	Iconic	40	5
Component: 	Iconic	40	5
Fork: 	Iconic	50	2
Join: 	Iconic	60	2
Composite state:  Note: refer to the whole shaded area	Iconic	20	4
Swim lane:  Note: refer to the whole shaded area	Iconic	40	2
Note: 	Iconic	70	2
Actor: 	Iconic	90	1
Average	Iconic	51.25	2.88
Average (best 7 notations)	Iconic	55.71	2.71

Systems Methodology. It is possible to make the signs more intuitive while avoiding visual clutter.

In addition to the above two propositions, we observed two interesting results. First, from the content analysis, we found that expert UML users had better performance on interpretation of symbolic UML graphical notations than novice users. Although interesting, the result is not surprising. Conventions are in hierarchical layers and they can be learned by sign users. Modeling method notations that involve existing layers of convention are not necessarily harder to interpret for expert users. For example, certain graphical notations always represent the same modeling construct in the field, e.g., a line drawn between two graphical notations usually means that there is a relationship or an association between the two graphical notations. Another example in UML is the use of string. The string involves human language conventions, which are purely arbitrary but are part of existing conventions that have already been learned. Also, according to cognitive theory, expert users automate their learned knowledge while they perform information processing tasks and, therefore, tend to have better performance than novices [37]. A UML graphical notation will be more frequently used by an expert than a novice. Thus, experts in UML performed better in interpreting symbolic graphical notations than novices simply because they have used the symbolic graphical notations more often. One subject who had used UML intensively performed the best in interpreting both symbolic and iconic notations. Interestingly, he accepted the symbolic notations so well

Table 3 Research results for symbolic graphical notations

UML graphical notation		Type of graphical notation	Denotations (percentage of subjects who provided correct answers) (%)	Number of connotations
Class/object:		Symbolic	80	8
Aggregation:		Symbolic	40	7
State:		Symbolic	40	4
Interface:		Symbolic	10	3
Association/link:		Symbolic	60	3
Dependency:		Symbolic	50	7
Message and stimulus:		Symbolic	30	3
Generalization:		Symbolic	30	4
Average		Symbolic	42.5	4.88
Average (best 7 notations)		Symbolic	47.14	5.14

that he refused to create any alternative notations in Part II of the questionnaire by commenting "These graphics make sense to me as they are…".

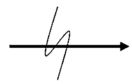
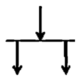
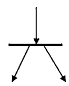
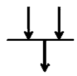
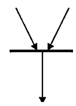
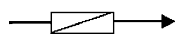
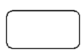
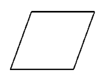
Second, the study reveals that the number of correct denotations given by the subjects for symbolic signs is not lower than the numbers of correct denotations for iconic sign if the symbolic signs are within the UML kernel or core (see [14]). Typically, the kernel of a modeling language consists of 20% of the modeling constructs that are used 80% of the time. Thus, the UML kernel is the core part of UML language, that part that is used to model 80% percent of the common problems. After three rounds of Delphi study [14], participants identified the most important diagrams in UML as class, use case, sequence, and statechart diagrams. In addition, at least 90% of the experts in the Delphi study agreed that those four diagrams should comprise the UML kernel. Thus, those symbolic UML graphical notations that are part of the UML kernel are interpreted more accurately and are not necessarily worse off than iconic UML graphical notations. For example, the graphical notation for class, a symbolic sign, is interpreted correctly 80% of the time. That is only worst than the most iconic UML graphical notation, the stick-man for actor. This can be explained by the fact that kernel or core notations, such as class, association, and state, are used more frequently by the subjects. Therefore, they are very familiar to UML users and are recalled more accurately.

These two observations point out that the degree of usage of a sign has an impact on its interpretation. An UML graphical notation can be used more because the user is an expert in UML, and has learned and used the UML graphical notation more extensively. An UML graphical notation can also be used more because it is part of the UML kernel or core.

8 Conclusions

Information system modeling languages, which extensively use visual signs (e.g., UML graphical notations) as basic language constructs, are subjected to the principles of semiotics. Evaluation of modeling language complexity

Table 4 Examples of suggested new UML graphical notations

Current UML notations		Suggested new notations	Descriptions
Message and stimulus:		 or 	The first one imitates paper scroll to represent the physical characteristic of a message. The second one uses lightning bolt to represent the stimulus
Fork:			The directions of the lines better represent the idea of "separation"
Join:			The directions of the lines better represent the idea of "combination"
Association/link:			Adding a thin rectangle formed by two triangle blocks to represent the idea of association and better differentiate it from other notations that involve lines
State:			This shape better represents the dynamic nature of a state and also differentiates the construct from other constructs such as class, object etc.

from the semiotics perspective can help us explain why certain modeling languages seem more complex to users. It can also help us identify ways to make the language easier to use and more intuitive; thus, facilitating learning and improving interpretation.

This research on UML graphical notations illustrates that semiotic concepts and frameworks can help in the evaluation and design of modeling languages. The research results support our hypotheses that the number of connotations is lower for iconic UML graphical notations when compared to symbolic UML graphical notations. However, the number of correct denotations given by the subjects for symbolic UML graphical notations is not necessarily lower than those for iconic UML graphical notations if the symbolic UML graphical notations belong to UML constructs that are within the UML kernel or core. The results also showed that expert users have more accurate interpretation of symbolic graphical notations. The research indicates that there are many factors we need to consider in designing and selecting graphical notations for modeling constructs, such as a sign's effectiveness (symbolic vs. iconic), the users level of experience (expert vs. novice), and the usage of the notations (kernel vs. non-kernel).

The findings of this research have important academic and practical implications. The study reveals the problems of current UML graphical notations and proposes some new graphical notations that are more intuitive and easier to learn and interpret. Academics and researchers can further explore the use of semiotics in guiding the development of modeling languages. Practitioners can make use of our results in the design and development of CASE tools to support systems analysis and design activities.

Our study has its fair share of limitations. The number of subjects in our study is small (i.e., 20 subjects). Future research can replicate this study using larger sample size and using different constructs from other modeling methods and languages. Also, the research used students as subjects. Most students did not have extensive industrial modeling experience. Practitioners' views of the UML graphical notations complexity may differ from that of students. Expert users automate the complex notations and are able to allocate more cognitive resources on actual modeling tasks. Students, nevertheless, are good surrogates for modeling professionals. Some of the subjects in our study were working professionals and had used UML for a number of years. Moreover, if we can revise UML graphical notations according to the principles of semiotics, we believe that even expert UML users can achieve better performance than they do with the current version of UML graphical notations.

References

1. Alemán J, Álvarez A (2001) Seamless formalizing the UML semantics through metamodeling. In: Siau K et al (eds) Unified modeling language: systems analysis, design and development issues. Idea Group Publishing, Hershey
2. Andersen P (1990) A theory of computer semiotics. Cambridge University Press, Cambridge
3. Barjis J, Chong S, Dietz J, Liu K (2002) Development of agent-based e-commerce systems using semiotic approach and DEMO transaction concept. Int J Inf Technol Decis Mak 1(3):491–510
4. Barron T, Chiang R, Storey V (1999) A semiotics framework for information system classification and development. Decis Support Syst 25:1–17. doi:10.1016/S0167-9236(98)00088-8
5. Barthes R, Smith M, Knott R, Berger J, Alty J (1994) Minimizing conceptual baggage: making choices about metaphor. In: People and Computers IX. HCI'94. Glasgow, August, pp 179–194
6. Batra D (2007) Cognitive complexity in data modeling: causes and recommendations. Requir Eng 12(4):231–244. doi:10.1007/s00766-006-0040-y
7. Booch G, Rumbaugh J, Jacobson I (2005) The unified modeling language user guide, 2nd edn. Addison Wesley, New York
8. Chandler D (2002) Semiotics for beginners. Available online at http://www.aber.ac.uk/media/Documents/S4B/semiotic.html
9. Condon C, Perry M, O'Keefe R (2004) Denotation and connotation in the human-computer interface: the 'save as…' command. Behav Inf Technol 23(1):21–31. doi:10.1080/01449290310001637415
10. De Souza C (1993) The semiotic engineering of user interface languages. Int J Man Mach Stud 39(7):53–773
11. Desouza K, Hensgen T (2002) On information in organizations: an emergent information theory and semiotic framework. Emergence 4(3):95–114. doi:10.1207/S15327000EM0403-07
12. Dobbing B, Parsons J (2008) Dimensions of UML diagram use: a survey of practitioners. J Database Manage 19(1):1–18
13. Dykman N, Griss M, Kessler R (1999) Nine suggestions for improving UML extensibility. In: UML'99—the unified modeling language. 1723:236–248
14. Erickson J, Siau K (2007) Theoretical and practical complexity of modeling methods. Commun ACM 50(8):46–51. doi:10.1145/1278201.1278205
15. Evermann J, Wand Y (2009) Ontology based object-oriented domain modeling: representing behavior. J Database Manage 20(1):48–77
16. Fiske J (1982) Introduction to communications studies. Routledge, London
17. Fontoura M, de Lucena C (2001) Extending UML to improve the representation of design patterns. J Object Oriented Program 13(11):12–19
18. Galloway A (1997) Questionnaire design and analysis. Available online at http://www.tardis.ed.ac.uk/~kate/qmcweb/qcont.htm
19. Gemino A, Parker D (2009) Use case diagrams in support of use case modeling: deriving understanding from the picture. J Database Manage 20(1):1–24
20. Hensgen T, Desouza K, Evaristoc J, Kraft G (2003) Playing the "cyber terrorism game" towards a semiotic definition. Hum Syst Manage 22:51–61
21. Liu K (2004) Virtual, distributed and flexible organizations: Studies in organizational semiotics. Kluwer, Dordrecht
22. Liu K (2005) Requirement reengineering from legacy information systems using semiotics techniques. Syst Signs Actions Int J Commun Inf Technol Work 1(1):38–61
23. Liu K, Clarke R, Andersen P, Stamper R (2001) Information, organization and technology: studies in organizational semiotics. Kluwer, Dordrecht
24. Liu K, Clarke R, Andersen P, Stamper R (2002) Organizational semiotics: evolving a science of information systems. Kluwer, Dordrecht
25. Loucopoulos P, Kadir WMNW (2008) BROOD: business rules-driven object oriented design. J Database Manage 19(1):41–73
26. Mehler A, Clarke R (2002) Systemic functional hypertexts. An architecture for social semiotic hypertext systems. In: New directions in humanities computing. Proceedings of the 14th joint international conference of the association for literary and linguistic computing and the association for computers and the humanities (ALLC/ACH '02), July 24–28. University of Tübingen, Seiten, pp 68–69
27. Meziane F, Athanasakis N, Ananiadou S (2008) Generating natural language specifications from UML class diagrams. Requir Eng 13(1):1–18. doi:10.1007/s00766-007-0054-0
28. Morris C (1946) Signs. Language and behavior. Prentice-Hall, New York
29. Nadin M (1981) The integrating function of the sign in Peirce's semiotic. In: Ketner KL, Ransdell JM, Eisele C, Fisch M, Hardwick C (eds) Proceedings of the C.S. Peirce Bicentennial International Congress, vol 23. pp 363–366
30. Nadin M (1997) Signs and system, in signs and systems. A semiotic introduction to systems design. Cambridge University Press, Cambridge
31. OMG (2003) OMG unified modeling language specification. Object Management Group, Version 1.5
32. Osgood C, George J, Percy H (1957) The measurement of meaning. University of Illinois Press, Urbana
33. Peirce C (1931) Collected writings (8 vols). In: Hartshorne C et al (eds) Harvard University Press, Cambridge, p 58
34. Reinhartz-Berger I, Sturm A (2008) Enhancing UML models: a domain analysis approach. J Database Manage 19(1):74–94
35. Rossi M, Brinkkemper S (1996) Complexity metrics for systems development methods and techniques. Inf Syst 21(2):209–227
36. Shen Z, Siau K (2003) An empirical evaluation of UML notational elements using a concept mapping approach. In: Paper presented at the 2003 International Conference of Information System (ICIS 2003)
37. Siau K (1999) Information modeling and method engineering: a psychological perspective. J Database Manage 10(4):44–50
38. Siau K (2004) Informational and computational equivalence in comparing information modeling methods. J Database Manage 15(1):73–86
39. Siau K (2007) The future of information systems engineering. Requir Eng 12(4):199–202. doi:10.1007/s00766-007-0059-8
40. Siau K, Cao Q (2001) Unified modeling language—a complexity analysis. J Database Manage 12(1):26–34
41. Siau K, Erickson J, Lee L (2005) Theoretical versus practical complexity: the case of UML. J Database Manage 16(3):40–57
42. Siau K, Lee L (2004) Are use case and class diagrams complementary in requirements analysis?—an experimental study on use case and class diagrams in UML. Requir Eng 9(4):229–237. doi:10.1007/s00766-004-0203-7
43. Siau K, Loo P (2006) Identifying difficulties in learning UML. Inf Syst Manage 23(3):43–51
44. Siau K, Rossi M (2009) Systems analysis and design: evaluation techniques for conceptual and data modeling methods. Inf Syst J
45. Siau K, Tan X (2005) Improving the quality of conceptual modeling using cognitive mapping techniques. Data Knowl Eng 55(3):343–365. doi:10.1016/j.datak.2004.12.006
46. Siau K, Tan X (2006) Using cognitive mapping techniques to supplement UML and UP in information requirements determination. J Comput Inf Syst 46(5):59–66
47. Siau K, Tan X (2008) Use of cognitive mapping techniques in information systems development. J Comput Inf Syst 48:49–57

48. Siau K, Tian Y (2001) The complexity of unified modeling language—a GOMS analysis. Paper presented at the 2001 International Conference of Information System (ICIS 2001)
49. Siau K, Tian Y (2002) Analyzing unified modeling language using GOMS. Paper presented at the 12th Workshop on Information Technology and Systems. Barcelona, Spain
50. Siau K, Wang Y (2007) Cognitive evaluation of information modeling methods. Inf Softw Technol 49(5):455–474. doi:10.1016/j.infsof.2006.07.001
51. Simons H, Graham I (1999) 30 Things that go wrong in object modelling with UML 1.3. In: Kilov H et al (eds) Precise behavioral specification of businesses and systems. Kluwer, Dordrecht
52. Smolander K, Rossi M (2008) Conflicts, compromises, and political decisions: methodological challenges of enterprise-wide E-business architecture creation. J Database Manage 19(1):19–40
53. Song X, Osterweil L (1992) Toward objective, systematic design-method comparisons. IEEE Softw 9(3):43–53. doi:10.1109/52.136166
54. Stamper R (1996) Signs, norms, and information systems. In: Holmqvist B et al (eds) Signs at work. Walter De Gruyter, Berlin
55. Stamper R, Liu K, Hafkamp M, Ades Y (2000) Understanding the roles of signs and norms in organizations—a semiotic approach to information system design. Behav Inf Technol 19(1):15–27. doi:10.1080/014492900118768
56. Underwood M (2003) Introductory models and basic concepts: semiotics. Available on line at http://www.cultsock.ndirect.co.uk/MUHome/cshtml/semiomean/semio1.html
57. VanderMeer D, Dutta K (2009) Applying learner-centered design principles to UML sequence diagrams. J Database Manage 20(1):25–47
58. Wand Y, Weber R (1993) On the ontological expressiveness of information systems analysis and design grammars. J Inf Syst 3(4):217–237
59. Zhang H, Kishore R, Ramesh R (2007) Semantics of the MibML conceptual modeling grammar: an ontological analysis using the Bunge-Wang-Weber framework. J Database Manage 18(1):1–19