Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2005 Proceedings

Americas Conference on Information Systems (AMCIS)

2005

A Semiotics View of Modeling Method Complexity - The Case of UML

Keng Siau
University of Nebraska-Lincoln, siauk@mst.edu

Yuhong Tian
University of Nebraska at Lincoln, yhtian@unlserve.unl.edu

A Semiotics View of Modeling Method Complexity — the Case of UML

Keng Siau
Department of Management
University of Nebraska – Lincoln
Email: ksiau@unl.edu

Yuhong Tian
Department of Management
University of Nebraska – Lincoln
Email: yhtian@unlserve.unl.edu

ABSTRACT

Unified Modeling Language (UML) is the standard modeling language for object oriented system development. Despite its status as a standard, UML's formal specification is fuzzy and its theoretical foundation is weak. Semiotics, the study of signs, provides us good theoretical foundation for UML research as UML graphical notations are some kinds of signs. In this research, we use semiotics to study the graphical notations in UML. We hypothesized that using iconic signs as UML graphical notations leads to more accurate representation and arouses fewer connotations than using symbolic signs. Since symbolic signs involve more learning efforts, we assume that expert users of UML will perform better with symbolic signs than novice users. We created an open-ended survey to test these hypotheses. The qualitative analysis of the survey process can help us gain in-depth understanding of the complexity of modeling language graphical notations. In addition, the introduction of semiotics in this research helps build a solid theoretical foundation of IS modeling method research.

Keywords

UML, Semiotics, Notations, Symbolic, Iconic, Connotations, Denotations.

INTRODUCTION

Information system modeling language is converging into one single standard — UML, the Unified Modeling Language. This unification has rescued system analysts from getting lost in the jungle of modeling methods. However, this is still much to be done with modeling language development. UML, together with its foundation — the object-oriented design methodology, is constantly evolving. As we gain more knowledge of the modeling process itself and the black box of human mind in the process of modeling and model interpretation, we can continuously improve the modeling language to make it serve us better. In this paper, we take UML as an example to show how the knowledge we gained 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 (Alemán, 2001; Dykman, 1999; Fontoura, 2001). One of the criticisms of UML is that it is large and complex, which can be daunting to novice users. Evaluation of modeling language complexity will help us identify ways to make the language easy to use (Siau & Cao, 2001, Siau & Tian 2001, 2002, Tian, 2002, Erickson & Siau, 2004).

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 political posters. This paper takes UML as an example to show how the knowledge we gained from semiotics can help in the evolution of modeling language. We argued that information system modeling language, which extensively uses visual signs (i.e., UML graphical notations) as basic language constructs, is subject to the principles of semiotics – the study of signs. We hypothesized that using iconic sign as UML graphical notation leads to more accurate representation and arouses fewer connotations than using symbolic signs. The survey helps us test the hypothesis.

The paper is organized as follows: First we introduce the prior research on modeling language/UML complexity. Then we cover some models/principles in semiotics, which form the justification and theoretical basis of the research. The remaining parts of the paper are the research method, procedures, results, and discussions.

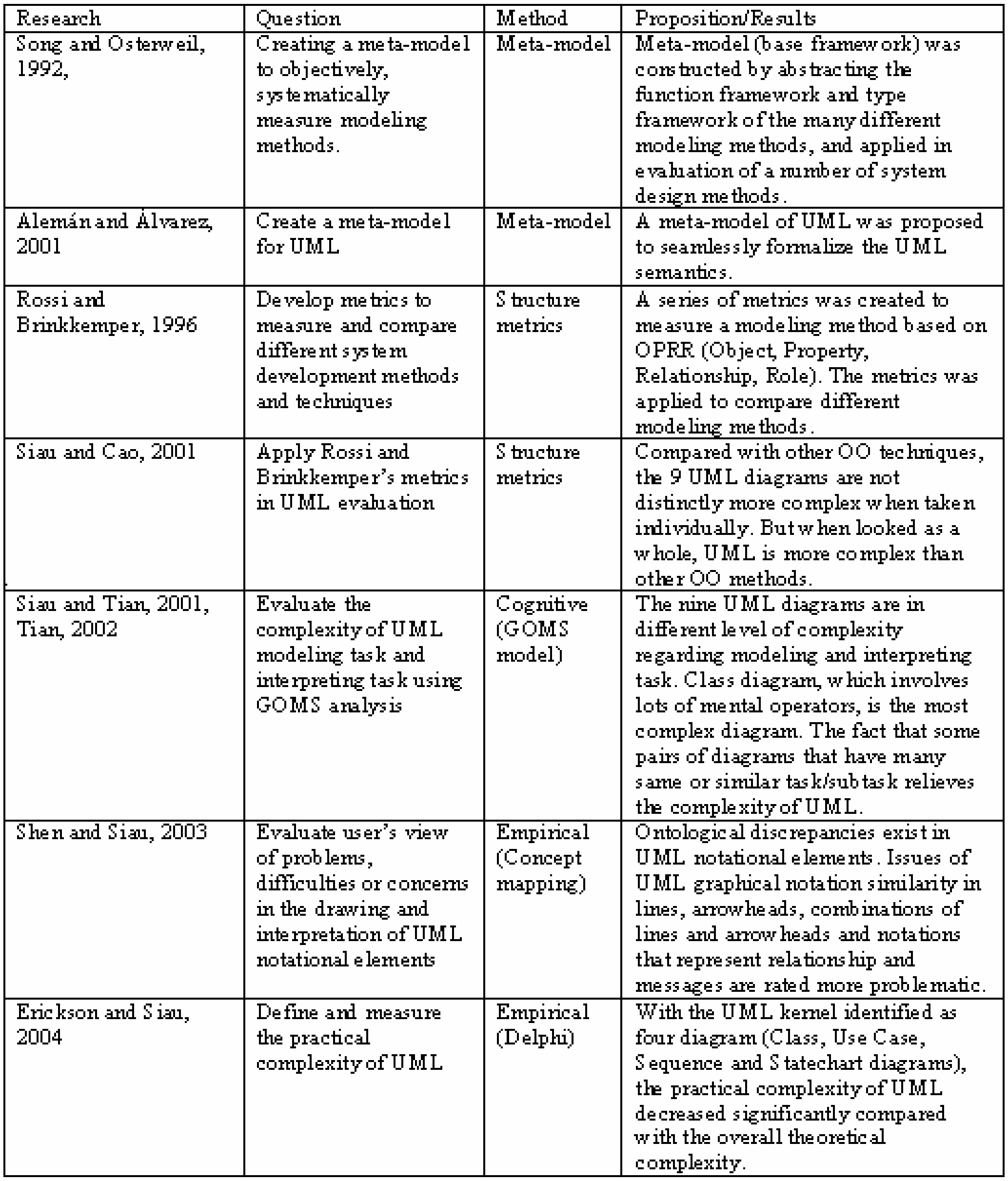
LITERATURE REVIEW

Compared with other visual modeling languages, including entity-relationship modeling, BPR flow charts, and state-driven languages, UML provides improved expressiveness and holistic integrity (OMG, 1999). UML is more expressive yet cleaner and more uniform than Booch, OMT, OOSE, and other methods because it removes the unnecessary differences in notation and terminology that obscure the underlying similarities of most of these methods. On the other hand, UML has been criticized as ambiguous, inadequate, and cognitively misdirected (Simon and Graham, 1999).

As a communication tool among all the stakeholders in software development including end users, system designers and developers, the key character of a modeling language is ease of use. The modeling language should be easy to operate (construction) and easy to understand (interpretation) (Siau and Tian, 2001, Tian, 2002). Therefore, an effective measure of the complexity of a modeling language has very significant theoretical and practical implications.

One way of measuring modeling language is to create meta-models of the modeling language (Song and Osterweil, 1992, Alemán and Álvarez, 2001), which makes different modeling languages using various constructs comparable to each other. Another method of non-empirical research on modeling language complexity is to use structure matrices (Rossi and Brinkkemper, 1996, Siau and Cao, 2001). Structure metrics provide an objective measure of the modeling language complexity. On the other hand, modeling and model interpretation are human mind processes. The explanation of these processes may require the use of psychology and cognitive theory. GOMS is a cognitive theory that describes the procedures required for accomplishing a general set of tasks in standard statements. It was used to analyze UML modeling task and interpreting task complexity (Siau and Tian, 2001, Tian, 2002). Last but not least important, the level of complexity of modeling language is basically based on perception of the stakeholders. Empirical research on UML complexity includes user's view of UML notational element (Shen and Siau, 2003) and user's view of practical complexity vs. theoretical complexity of UML (Erickson and Siau, 2004). Table 1 summarizes some of the existing work in the area.

Modeling method complexity is not an independent phenomenon. Wand and Weber (1993) introduced the ontology defined by Bunge into information modeling. The resulted theory is referred to as BWW ontology. According to this widely accepted theoretical foundation of IS model and modeling, an information modeling method construct must have a counterpart in the real world. Construct excess (construct of modeling method does not have a real work counterpart) or construct deficiency (no modeling construct exists for a corresponding real world object) are two situations that a good modeling method must try to avoid. However, the real world could be so extremely complex that a modeling method can never be able to model it exactly and at the same time be reasonably wieldy. The important issue of a modeling method is to determine the appropriate level of real world complexity it intends to capture. In most of the 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. It is the complexity of the real world problem that makes a modeling method complicated. Therefore, a modeling method faces a dilemma between expressive power and simplicity: simplified modeling method is at risk of failing to model the complex real world due to insufficient constructs. To compensate for that, a simplified modeling method tends to assign many denotations to one single construct (construct overloading). When there are excessive construct overloadings, cognitive complexity, i.e., user's difficulty in understanding the constructs, will become inevitable. On the other hand, a complex modeling method with correspondent number of constructs to the real world, as recommended by the "BWW" ontology theory (Wand & Webber, 1993), is able to model the real world accurately. But it will inevitably sacrifice the simplicity of modeling method by adding too many model elements and notational constructs. This is the case of UML: these excessive constructs and the resultant mappings between them is the reason of the structural complexity of UML as revealed by the metrics analysis of UML conducted by Siau and Cao (2001).



Research	Question	Method	Proposition/Results
Song and Osterweil, 1992,	Creating a meta-model to objectively, 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, and applied in evaluation of a number of system design methods.
Alemán and Álvarez, 2001	Create a meta-model for UML	Meta-model	A meta-model of UML was proposed to seamlessly formalize the UML semantics.
Rossi and Brinkkemper, 1996	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, 2001	Apply Rossi and Brinkkemper's metrics in UML evaluation	Structure metrics	Compared with other OO techniques, the 9 UML diagrams are not distinctly more complex when taken individually. But when looked as a whole, UML is more complex than other OO methods.
Siau and Tian, 2001, Tian, 2002	Evaluate the complexity of UML modeling task and interpreting task using GOMS analysis	Cognitive (GOMS model)	The nine UML diagrams are in different level of complexity regarding modeling and interpreting task. Class diagram, which involves lots of mental operators, is the most complex diagram. The fact that some pairs of diagrams that have many same or similar task/subtask relieves the complexity of UML.
Shen and Siau, 2003	Evaluate user's view of problems, difficulties or concerns in the drawing and interpretation of UML notational elements	Empirical (Concept mapping)	Ontological discrepancies exist in UML notational elements. Issues of UML graphical notation similarity in lines, arrowheads, combinations of lines and arrowheads and notations that represent relationship and messages are rated more problematic.
Erickson and Siau, 2004	Define and measure the practical complexity of UML	Empirical (Delphi)	With the UML kernel identified as four diagram (Class, Use Case, Sequence and Statechart diagrams), the practical complexity of UML decreased significantly compared with the overall theoretical complexity.

Table 1. Research on the complexity of modeling methods.

Prior research on UML complexity only revealed the facts of complexity, seldom did they mention why certain constructs or issues are more complex to UML users, nor did they provide constructive suggestions to solve the problems. Modeling method theories such as BWW ontology can not solve the expressive power vs. simplicity dilemma of modeling method in that a modeling method with constructs that correspond one-to-one to real world counterparts (as suggested by BWW ontology) can become extremely complex and unwieldy.

One solution to the expressive power vs. simplicity dilemma is to make the UML graphical notations:

- Intuitive, easier to understand, memorize and use, thus relieve the cognitive overloading due to too many constructs involved, and allow the modeler/interpreter to assign more cognitive space to the model itself rather than to recall and memorize graphical notations of modeling constructs;
- Unambiguous, so that one notation means the same modeling construct to different users.

The knowledge we gained from the study of signs can help us achieve these goals. Semiotics can not only explain why certain constructs or issues are more complex than the others and what kind of visual signs are more intuitive, thus easier to learn/use, it also tells us how to make a modeling language simpler, more concise yet accurate. In addition, previous research in modeling method is typically weak in comprehensive theory foundations. The introduction of semiotics into modeling method complexity measure helps to build solid theoretical foundation of modeling method research.

THEORETICAL FOUNDATION –SEMIOTICS AND WHAT IT CAN DO HERE

Semiotics is the study of signs, or the general theory of representation (Morris, 1946). 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 represents a range of studies in art, literature, anthropology, and mass media rather than an independent academic discipline. Use of semiotics in IT area is not a new phenomenon. HCI has vigorously applied principles of semiotics for decades (Anderson, 1990; De Souza 1993). Semiotics in HCI mainly focuses on analyzing the visual signs, i.e., icons in GUI design, using semiotics approach. Nadin pointed out that human computer interaction is semiotics applied, "If there is a science of interface (computing interface or any other kind), then this science is semiotics" (Nadin, 1981). Other research that applied semiotics in IT area include semiotics framework of information evolution (Desouza & Hensgen, 2002 & Hensgen et al., 2003), and semiotics framework of information system classification and development (Barron, 1999). Some researchers focus on the social dimension of semiotics and its role in information system development, such as the ontology chart proposed by Stamper *et al. (*2000) for user requirements acquisition. Nevertheless, there is no prior application of semiotics in modeling language evaluation. We believe that modeling language, as its name indicates, is subject to the general guidance of knowledge of signs, i.e., semiotics.

Two models

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

(1) Saussure's dyadic model:

- A '**signifier' -** the *form* which the sign takes
- The '**signified' -** the *concept* it represents.

The relationship between the signifier and the signified is referred to as 'signification'

(2) Peirce's triadic model:

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

For example, a straight line in UML represents link, the line itself is a "signifier" or "representamen" and the construct of *link* it represents is "signified" or "object". Variants of Peirce's triad are often presented as **'the semiotic triangle**' (as if there were only one version). The following is a frequently referenced version, which changes only the unfamiliar Peircean terms (Chandler, 2002):

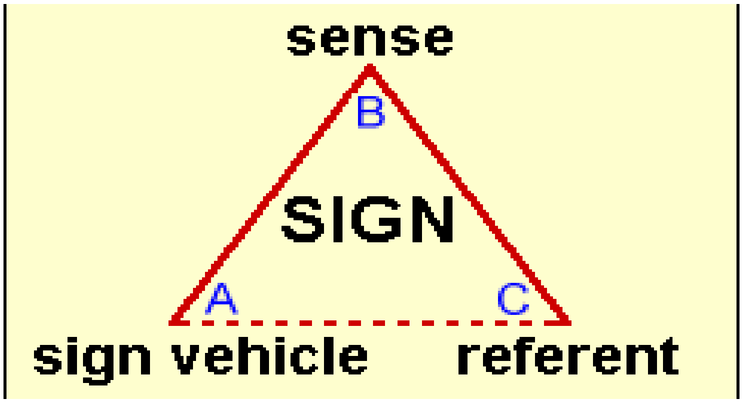


Figure 1. A variant of Peire's semiotic triad model (Chandler, 2002)

In the case of modeling method, 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 (making sense).

Modes of Signs

Most semiotics researchers stressed that signs differ in how arbitrary/conventional (or by contrast 'transparent') they are. Hence there are three modes of signs (Chandler, 2002).

- **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 (recognizably looking, sounding, feeling, tasting or smelling like it) - being similar in 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 (physically or causally) to the signified - this link can be observed or inferred: e.g. 'natural signs' (smoke, thunder), personal 'trademarks' (handwriting).

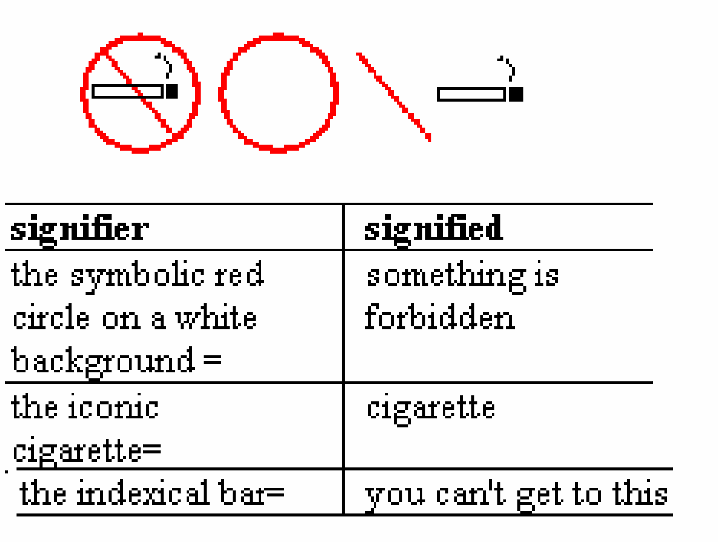
These categories are not mutually exclusive. A sign could very well be all three at the same time, i.e., part of the sign is symbolic, whereas the other part of the signs could be iconic or indexical. A good example is Chinese characters. One major category of Chinese characters is called "shape and sound" in which half of the character (which is iconic) looks like the thing (referent) it represents and the other half of the character (which is symbolic) has no meaningful relationship with the referent but purely 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 (Fiske, 1982). As Chandler (2002) 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 is, the more the signifier is constrained by the signified (for example, a picture is constrained by the objectives inside the picture because the picture must resemble the objective it is representing), and the 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 need to learn those conventions to understand the sign, for example, a word in foreign language. The above discussions on degree of signifier constrained by the signified indicate 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.

Underwood (2003) provided an example of analyzing modes of signs and the signifier/signified of signs (Figure 2).



signifier	signified
the symbolic red circle on a white background =	something is forbidden
the iconic cigarette=	cigarette
the indexical bar=	you can't get to this

Figure 2. Analyzing modes of signs (Underwood, 2003)

In a system analysis and design method, if the graphical notation of a modeling method has a higher degree of convention, in other words, it is more arbitrary, more symbolic and less iconic, it will demand more of the interpreter/user's learning effort, and vice versa. Thus a qualitative analysis of a modeling method's graphical notation regarding the three modes or the level of its conventions can serve as an evaluation of the ease of use of the modeling method.

Denotation vs. connotation

Denotation is the definitional, "literal", "obvious" or "commonsense" meaning of a sign.

Connotations are *any other* meanings associated with the sign to the interpreter, the socio-cultural and "personal" associations (ideological, emotional etc.) of the sign. These are typically related to the interpreter's class, age, gender, and ethnicity etc. The phrases "IRA terrorists" and "IRA freedom fighters" denote the same people, but they connote something quite different.

Of course, connotations of a sign are not necessarily social or emotional. Condon *et al.* (2004) studied the connotations of "save as…" command in Microsoft Word and found variations of connotations and inconsistent denotations of the term. Similar research can be done on modeling method by evaluating the connotations and denotations of the graphical notations of a modeling method. UML graphical notations with too many connotations in the eyes of an end user are not good notation/signs. Similarly, if end users tend to get incorrect/inconsistent denotation of one UML notation, that UML notation is not simple in nature.

RESEARCH MODEL AND PROPOSITIONS

UML is a modeling language using graphical notations as signs. The dyadic (two factors) and triadic (three factors) models of UML graphical notations are:

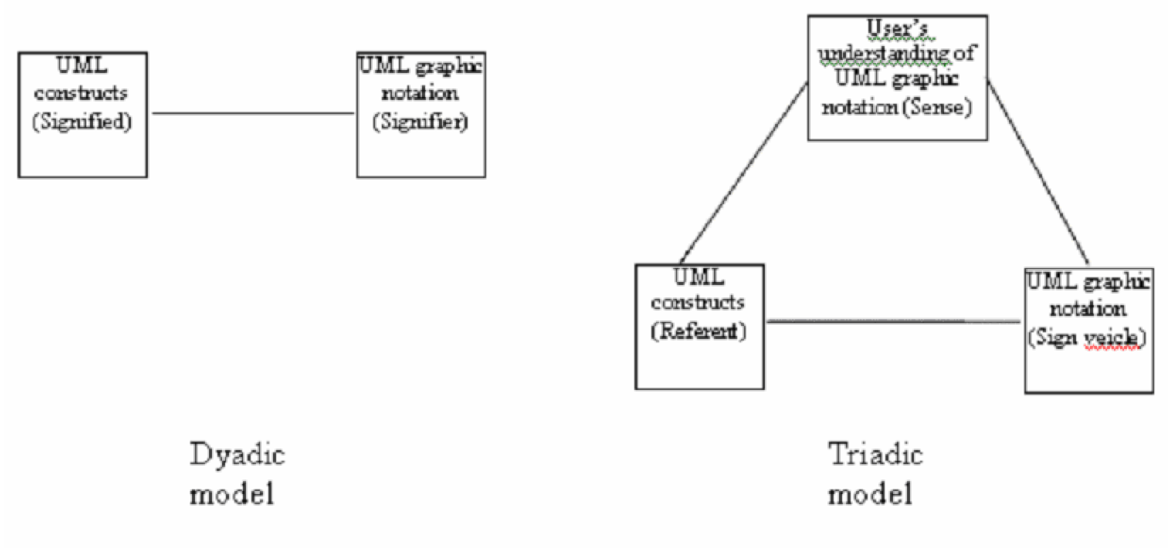

UML constructs (Signified)
UML graphic notation (Signifier)
Dyadic model
User's understanding of UML graphic notation (Sense)
UML constructs (Referent)
UML graphic notation (Sign veicle)
Triadic model

Figure 3. The dyadic and triadic model of UML graphic notations

As we mentioned before, the three modes of sign are different in how the signifier is constrained by the signified, i.e., they are differing in how arbitrary/conventional they are. An arbitrary/conventional graphical notation is hard to learn and understand by the interpreter/user because it involves more agreed convention that requires ones to learn.

Conventions are in hierarchical layers. Modeling method notations that involve some existing layers of convention are not necessarily hard to learn for expert users. For example, there are agreements among system analysts that certain graphical notations always represent the same/similar modeling construct, e.g., a line drawn between two graphical notations usually means there is some relationship between the two notations. Another example in UML is the usage of string. The string involves human language conventions, which are purely arbitrary but are part of existing convention (or prior knowledge) that is already leaned when we are modeling.

As Chandler (2002) pointed out, the purely symbolic signs are the unmotivated. Symbolic signs combined with some iconic feature are more motivated. The less motivated the sign is, the more learning of an agreed convention it requires. The iconic and indexical signs, especially the iconic signs, are more motivated and therefore are much easier to learn, thus leading to more accurate representation. As mentioned before, denotation means the definitional meaning of a sign, whereas connotation means any other meanings associated with the sign to the interpreter. We can use user interpretation of the denotation and connotation of the sign (specifically, UML notation) to show how accurate the sign represents the referent. We create our proposition as follows:

The UML graphical notations that are more motivated will demand less learning effort and represent the referent more accurately. Consequently:

1. The number of connotations connected with high motivated signs should be less, and the denotation of these sign should more consistently be the signs' referent.

Signs are in hierarchical layers of conventions, and expert users may possess more learned prior knowledge of conventions, therefore:

2. Expert users may have better performance on interpretation of less motivated sign than novice users.

RESEARCH PROCEDURE

An open-ended questionnaire in the form of a table is used to solicit user's view of denotation and connotation of the UML graphical notations. The graphical notations include both symbolic and iconic.

For each notation, the subjects were asked to write down whatever UML construct she/he thinks the left notation should represent (denotations) and also write down as many UML modeling constructs as possible that the notation on the left column can represent (connotations).

The participants of the research are graduate students majoring in management information system or computer science and software engineering. Another requirement of the subjects is that they must have taken the course of UML, and preferably have used UML in practice.

After the data is collected, content analysis technique is used to break down the response into separate significations/connotations. The total number of connotations for each UML graphical notations is summarized. Statistical tests will be conducted to see whether there is significant difference in accurate representation (denotation) between the symbolic and the iconic UML notations. Statistical tests will also be used to test whether the level of connotation is contingent on the two types of UML notations. Differences of expert vs. novice user performance regarding these dependent variables will also be tested.

In the second part of the research, we will invite subjects to create some new graphical notations that they think can "better represent" UML constructs. We believe the results will be very interesting and very helpful for future UML evolution and new modeling method development.

Open-ended Questionnaires

Eliciting denotations and connotations

Denotation, by definition, is the designed referent of the UML signs. In this research, we conducted a study to find the denotation and connotations of the UML graphical notations that are in different level of motivation.

The immediate issue is how to elicit denotation/connotations. Osgood *et al.* (1957) 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*. (2004) used short "what for" interview to uncover the denotation and connotations of the sign of "save as…"command to users. The researchers continuously ask "what for" until the answers formed a closed loop or the interviewee felt that the questions were unanswerable. Content analysis technique is then used to break down the response into separate significations.

However, Condon *et al.* (2004) pointed out that on-spot interview method has two problems: interviewee may have difficulty coming up with the right word, or coming up with the right word at exact the moment. To solve the problems, we choose to use open-ended question survey to elicit connotations. According to Galloway (1997), the advantages of open-ended questions include: greater freedom of expression; no bias due to limited response ranges; and respondent can qualify their answers.

Selecting UML graphical notations

To create our questionnaires, the first issue is to select appropriate notations that represent symbolic signs and iconic signs.

According to OMG (2003), UML has four types of graphical construct: icons, two-dimensional symbols, paths and strings. Although UML specification used the words "icons" and "symbols" here, they are not as rigorously defined as in the context of semiotics. The reason is they are used casually and interchangeably in the specification. As mentioned before, no matter how iconic a sign is, some level of conventions will always involve. Especially in UML, a graphical notation can never be pure iconic. The classification of UML notation in terms of mode of sign may be difficult and subjective. However, mode analysis example given by Underwood (2003) provides us a good example of qualitative analysis method to classify UML notations into appropriate type of signs. Based on the analysis, we choose 8 signs that are symbolic and 8 signs that are more iconic. There are more symbolic signs in UML than so-called iconic signs. The frequencies that different UML notations are actually used by practitioners vary widely (Erickson and Siau, 2004). It would make less practical meaning if the notations we choose for this study are seldom used in practice. Therefore we choose eight more important and frequently used symbolic notations as shown in the upper half of table 2.

PILOT STUDY AND RESULTS DISCUSSIONS

As a pilot study, we selected 10 PhD students who are familiar with UML to complete the questionnaires. The experiment is timed. Participants are not allowed to use reference books, neither can they discuss with other people about the questionnaires before they submit their answers. The results are shown in table 2.

The results are mixed. The number of connotations appears to be lower for iconic than symbolic. The most iconic sign, the sticker man in use case, has almost no connotations. Low number of connotation means more accurate representations.

However, the pilot study revealed some problems that need to be addressed in the future full-scale study. For example, the number of correct denotations given by the subjects are not necessarily high in iconic than in symbolic. This can easily be explained by the fact that we choose the most important/used symbolic signs in UML when designing the questionnaires. These signs, such as class, association, state, are the most used/practiced by the subjects, therefore are recalled more correctly. Based on our proposition, the iconic signs should outperform symbolic signs if they are used/practiced at the same level. Specifically, when the subjects start to learn UML, we believe the iconic signs will create the lowest burden of memorizing. UML 2.0 has been released and there are some new notations added. In the future full-scale research, we will be able to choose more signs that are at the same level of usage to avoid this problem.

Regarding the performance difference between expert user and novice users, the pilot study shows certain support to our proposition that expert users have higher performance/acceptance of symbolic notations. There is one subject who has used UML intensively. His performance is better than the other subjects for both symbolic and iconic notations. Interestingly, he accepts the symbolic notations so well that he refuse to create any alternative notations in part II of the questionnaires by commenting: “these graphics make sense to me as they are…”

Part II of the questionnaires invites subjects to create new graphical notations that they think can “better represent” UML constructs. The pilot study yields some interesting results. The alternatives provided by the subjects make the notations either more iconic or more distinct compared to other similar notations. Examples of these suggested notations are presented in the table 3. The research is on going and more results will be available in the near future. The examples provided here show that we don’t necessarily have to add too much visual detail to a notation to make it more iconic. The suggested new notations are not very visually clustered, but they are more representative of their corresponding constructs, in other words, more close to iconic signs.

IMPLICATIONS AND FUTURE RESEARCH

This paper takes UML as an example to show how the knowledge we gained from semiotics can help in the evolution of modeling language. Information system modeling language, which extensively uses visual signs (i.e., UML graphical notations) as basic language constructs, is subject to the principles of semiotics. Evaluation of modeling language complexity from the semiotics perspective can help us identify ways to make the language easy to use. The findings of this research will have important academic and practical implications. It can help reveal the problems of current UML graphical notations and create modeling graphical notations that are more intuitive and easier to learn and interpret.

In addition to completing the study in a full scale, some future research may include:

- Use cognitive psychology theory to explain the detail of human mind in interpreting/using symbolic notation and iconic notations;
- Create a pool of alternative UML graphical notations based on the part II questionnaires. Ask subjects to rate them in terms of how good of a representation they are.
- Use the highly rated alternative UML graphical notations to create a domain model that is informational equivalent to a UML model built by current notations. Compare the performance of the two models in terms of user’s interpretation accuracy and interpretation speed.

Admittedly, use of students as subjects is problematic. Most students don’t have actual modeling experience except for coursework project. Practitioner’s view of the UML graphic notation complexity might be different from that of the beginning learners. An UML graphic might not seem to be as complex as it does initially after one intensively use it in, for example, dozens of projects. An expert UML user automates the complex notation and thus is able to spare more cognitive space on actual modeling task. However, if we can design/edit UML according to the principle of semiotics, we believe that an expert UML user, after the initial necessary training of the new version, can achieve even better performance than they do with the current UML version.

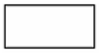
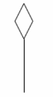
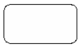
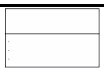
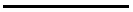
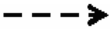
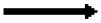
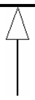
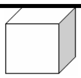
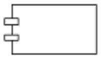
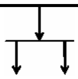
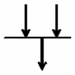
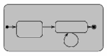
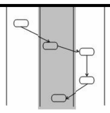
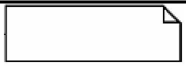
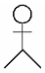
UML Construct graphical notation	Symbolic (s) /Iconic (I)	Denotations-Number of subjects who give correct answers (out of 10)	Number and summarized connotations
Class/object: 	S	8	8 connotations: Class, object, focus of control, signal, data type, note, state, package
Aggregation: 	S	4	7 connotations: Composition, relationship, association, inheritance, connection, generalization, restraints
Action state: 	S	4	4 connotations: Activity state, action state, class, object
Interface: 	S	1	3 connotations: Class, object, interface,
Association/link: 	S	6	3 connotations: Link, association, relationships
Dependency: 	S	5	7 connotations: Relationship, Link, data flow, transition, message, message pass/call, conditional flow
Message and Stimulus: 	S	3	3 connotations: Data flow, message pass, call, create, destroy, activity transition
Generalization: 	S	3	4 connotations: Flow, inheritance, transition, inherit
Node: 	I	4	5 connotations: Class, node, component, database server, system
Component: 	I	4	5 connotations: Note, component, interface, package, interface
Fork: 	I	5	2 connotations: Join, branch
Join: 	I	6	2 connotations: Fork, branch
Composite state:  Note: refer to the whole shaded area.	I	2	4 connotations: State chart diagram, subsystem, activity diagram, state chart
Swim lane:  Note: refer to the whole shaded area.	I	4	2 connotations: Subsystem, sequence diagram
Note: 	I	7	2 connotations: class, object
Actor: 	I	9	1 connotations: people

Table 2. Pilot study results.

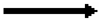
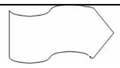
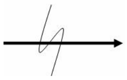
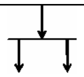
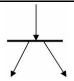
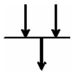
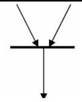
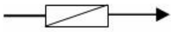
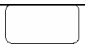
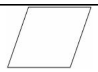
Current UML Notations	Suggested New Notations	Descriptions
Message and Stimulus: 	 or 	The first one imitates paper scroll to represent the physical character of a message. The second one uses lightning bolt to represent the stimulus.
Fork: 		The directions of the lines better represent the idea of "separation"
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.
Action state: 		This shape better represents the dynamic nature of action stage and also differentiates the construct more from class, object etc.

Table 3. Examples of Suggested New UML Notations.

REFERENCES

1. Andersen, P.B. (1990) A Theory of Computer Semiotics, Cambridge University Press, Cambridge, UK.
2. Alemán, José Luis Fernández and Álvarez, Ambrosio Toval. (2001) Seamless formalizing the UML semantics through metamodeling, ch14 in Unified Modeling Language: Systems Analysis, Design and Development Issues, Keng Siau, Terry Halpin (eds.), Idea Group Publishing.
3. Barron, T.M., Chiang, R.H, and Storey, V.C. (1999) A semiotics framework for information system classification and development. *Decision Support Systems*, 25, 1-17.
4. Barthes, R. Smith, M. Knott, R. Berger, J. and Alty, J. (1994) Minimizing conceptual baggage: making choices about metaphor. In People and Computers IX. HCI'94. Glasgow, August.179-194.
5. Chandler, D. (2002) Semiotics for beginners, (available online at http://www.aber.ac.uk/media/Documents/S4B/semiotic.html)
6. Condon, C., Perry, M. and O'Keefe, R. (2004) Denotation and Connotation in the Human-Computer Interface: The 'save as…' Command, *Behavior & Information Technology*, 23, 1, 21-31.
7. De Souza, C. S. (1993) The semiotic engineering of user interface languages, *International Journal of Man- Machine Studies*, 39, 7, 53-773.
8. Desouza, K. C. & Hensgen, T., (2002) Information" in Organizations: An Emergent Information Theory and Semiotic Framework, Emergence, 4, 3, 95–114
9. Dykman N., Griss M., & Kessler, R. (1999) Nine suggestions for improving UML extensibility, UML'99 – The Unified Modeling Language, 1723, 236-248
10. Erickson, J. & Siau, K. (2004) Theoretical and Practical Complexity of Unified Modeling Language: Delphi Study and Metrics Analyses, International Conference of Information System (ICIS 2004).
11. Fiske, J. (1982) Introduction to communications Studies, London: Routledge.
12. Fontoura M.; de Lucena, C.J.P. (2001) Extending UML to improve the representation of design patterns, *Journal Of Object-Oriented Programming*, 13, 11, 12-19

13. Galloway, A. (1997) Questionnaire Design & Analysis, (available online at http://www.tardis.ed.ac.uk/~kate/qmcweb/qcont.htm

14. Hensgen, T., Desouza K.C., Evaristoc, J. R., and Kraft, G.D., (2003) Playing the "cyber terrorism game" towards a semiotic definition, *Human Systems Management 22*, 51–61.

15. Morris, C.W. (1946) Signs, Language and behavior, Prentice-Hall, New York.

16. Peirce, C. S. (1931) Collected Writings (8 Vols.). Charles Hartshorne, Paul Weiss & Arthur W Burks (eds.). Cambridge, MA: Harvard University Press, 58.

17. Nadin, M. (1997) Signs and System, in Signs and Systems. A Semiotic Introduction to Systems Design, Cambridge: University Press.

18. Nadin, M. (1981) The integrating function of the sign in Peirce's semiotic, in Proceedings of the C.S. Peirce Bicentennial International Congress K.L. Ketner, J. M. Ransdell, C. Eisele, M. Fisch, C. Hardwick, (eds.), 23, 363-366

19. OMG. (2003) OMG Unified Modeling Language Specification, Object Management Group, Version 1.5.

20. Osgood, Charles E, George J Suci & Percy H Tannenbaum. (1957) The Measurement of Meaning, Urbana, IL: University of Illinois Press.

21. Shen, Z. & Siau, K, (2003) An empirical evaluation of UML notational elements using a concept mapping approach, paper presented at the 2003 International Conference of Information System (ICIS 2003)

22. Siau, K. and Cao, Q (2001) Unified Modeling Language (UML)—a Complexity Analysis, *Journal of Database Management*, 11, 1, 26-34.

23. Siau, K. and Tian, Y. (2002) Analyzing Unified Modeling Language Using GOMS, paper presented at the 12th Workshop on Information Technology and Systems. Barcelona, Spain.

24. Siau, K. and Tian, Y. (2001) The Complexity of Unified Modeling Language—A GOMS Analysis, paper presented at the 2001 International Conference of Information System (ICIS 2001)

25. Simons A. J. H. Graham, I. (1999) 30 Things That Go Wrong In Object Modelling With UML 1.3, In: Kilov H, Rumpe B. and Simmonds I. (eds.) Precise Behavioral Specification of Businesses and Systems, Kluwer Academic Publishers.

26. Song, Xiping and Osterweil, Leon J. (1992) Toward Objective, Systematic Design-Method Comparisons, *IEEE Software*, 9,3, 43-53

27. Stamper, R. Liu, K., Hafkamp, M., and Ades, Y.(2000) Understanding the roles of signs and norms in organizations –a semiotic approach to information system design", *Behavior & information technology*,19,1, 15-27.

28. Tian, Y. (2002) The Complexity of Unified Modeling Language—A GOMS Analysis, thesis, University of Nebraska-Lincoln,

29. Underwood, M. (2003) Introductory models & basic concepts: semiotics (available on line at http://www.cultsock.ndirect.co.uk/MUHome/cshtml/index.html)