Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

4-1999

Information modeling and method engineering: A psychological perspective

Keng SIAU
Singapore Management University, klsiau@smu.edu.sg

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Computer Engineering Commons, and the Databases and Information Systems Commons

Citation

SIAU, Keng. Information modeling and method engineering: A psychological perspective. (1999). *Journal of Database Management*. 10, (4), 44-50.
Available at: https://ink.library.smu.edu.sg/sis_research/9395

This Journal Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

Journal of Database Management

Copyright ©1999, Idea Group Publishing.

Information Modeling and Method Engineering: A Psychological Perspective

Keng Siau
University of Nebraska-Lincoln, USA

Information modeling is the cornerstone of information systems analysis and design. Information models, the products of information modeling, not only provide the abstractions required to facilitate communication between the analysts and end users, but they also provide a formal basis for developing tools and techniques used in information systems development. The process of designing, constructing, and adapting information modeling methods for information systems development is known as method engineering. Despite the pivotal role of modeling methods in successful information systems development, most modeling methods are designed based on common sense and intuition of the method designers with little or no theoretical foundation or empirical evidence. Systematic scientific approach is missing! This paper proposes the use of cognitive psychology as a reference discipline for information modeling and method engineering. Theories in cognitive psychology are reviewed in this paper and their application to information modeling and method engineering are also discussed.

Even though research in systems analysis and design has been going on for over 40 years, successful software development is still an art rather than a science. In the 1980s, Jones (1986) observed that a typical project was one year late and 100% over budget. Yourdon (1989) reported application backlogs of four to seven years or more. The maintenance phase typically consumed up to 70% of the programmer's effort, and it was errors, not enhancements, that accounted for 40% of maintenance (Rush, 1985). Page-Jones (1988) wrote: "It looks as if traditionally we spend about half of our time making mistakes and the other half of our time fixing them."

We are, however, no better as we move toward the end of this century. The IBM's Consulting Group (Gibbs 1994) released the results of a survey of 24 leading companies that had developed large distributed systems. The numbers were unsettling: 55% of the projects cost more than budgeted, 68% overran their schedules, and 88% had to be substantially redesigned. A recent high-profile failure is the Denver Airport baggage-handling system, responsible for delaying the opening of the airport. The Standish Group research (Chaos 1995) predicted that a staggering 31.1% of projects would be canceled before they ever get completed and 52.7% of projects would cost 189% of their original estimates.

In the early days of computerized information systems, technological failure was the main cause in the failure of business data processing systems (Avison & Fitzgerald 1995). Today, the failure of information systems is rarely due to technology that is on the whole reliable and well tested. Failure is more likely to be caused by miscommunication and misspecification of requirements. Similar sentiments were echoed in the Standish Group's report (Chaos, 1995) which listed incomplete requirements and specifications as the second most important factor that caused projects to be challenged and the top factor that caused projects to be impaired and ultimately canceled (Chaos, 1995). A recent survey of hundreds of Digital's staff and an analysis of the corporate planning database revealed that on average, 40% of the requirements specified in the feasibility and requirements phase of the life

Manuscript originally submitted October 5, 1998; Revised January 5, 1999; Accepted April 28, 1999 for publication.

cycle were redefined in the later phases. This cost Digital an average of 50% more than the budgeted amount (Hutchings & Knox, 1995).

The process of investigating the problems and requirements of the user community, and building an accurate and correct requirement specification for the desired system is known as information modeling (Siau, 1999; Siau & Rossi, 1998; Siau et al., 1997; Mylopoulos, 1992, Rolland & Cauvet, 1992; Kangassalo, 1990).

Information Modeling

Information modeling is the process of formally documenting the problem domain for the purpose of understanding and communication among the stakeholders (Siau, 1999; Siau, 1998; Mylopoulos, 1992). Information modeling is central to information systems analysis and design, and takes place in the early phases of the software development life cycle. The product of the information modeling process is one or more information models (e.g., data flow diagrams, entity-relationship diagrams, use cases, activity diagrams, sequence diagrams). Information model provides a conceptual basis for communicating and thinking about information systems (Willumsen, 1993), and a formal basis for tools and techniques used in the design and development of information systems (Kung & Solvberg, 1986).

Information models are constructed using information modeling method, which can be defined as an approach to perform modeling, based on a specific way of thinking, consisting of directions and rules, and structured in a systematic way (Brinkkemper 1996). There is no shortage of information modeling methods in the field. In fact, it is a "methodology jungle" out there (Avison & Fitzgerald, 1995). Olle et al. (1982) and Bubenko (1986) stated that the field was inundated by hundreds of different modeling methods. Recently, Jayaratna (1994) estimated that there were more than a thousand brand name methodologies worldwide. The quest to develop the next modeling method has been wittily termed the YAMA (Yet Another Modeling Approach) syndrome (Oei et al., 1992) and NAMA (Not Another Modeling Approach) hysteria (Siau et al., 1996). Even the new kid on the block, object oriented approach, has more than a dozen variants. Despite the "impressive" number, miscommunication and misspecification continue (Chaos, 1995).

To reduce the chances of misunderstanding and miscommunication during information modeling, the use of natural and intuitive modeling constructs (e.g., entity, relationship, object) in information modeling methods has been stressed and advocated (e.g., Chen, 1976; Coad & Yourdon, 1991). This, they claimed, would enable end-users to better understand the information depicted in the information model and to pinpoint incomplete or incorrect information in the model.

Method Engineering and Modeling Constructs

Modeling constructs are semantic primitives that are used to organize and represent knowledge about the domain of interest (Sernades et al., 1989). Modeling constructs form the core of an information modeling method. Method engineering is the process of designing, constructing, and adapting modeling methods for the development of information systems (Siau, 1999; Siau, 1998; Brinkkemper, 1996). To design, construct, and adapt methods, we need to understand the role and value of each modeling construct.

The importance of modeling constructs can be viewed from two perspectives: ontology and epistemology of information systems analysis and design. Ontology is concerned with the essence of things and the nature of the world (Wand & Weber, 1993; Avison & Fitzgerald, 1995). The nominalist position in ontology argues that "reality is not a given immutable 'out there', but is socially constructed. It is the product of human mind" (Hirschheim & Klein, 1989). The choice of modeling constructs, therefore, directly influences what the modeling method regards as important and meaningful versus what it suggests as unimportant and irrelevant. For example, the use of the entity-relationship (ER) approach emphasizes entities and relationships but ignores the processes involved. The use of the object-oriented (OO) approach, on the other hand, emphasizes objects and the behavior of objects.

Epistemology relates to the way in which the world may be legitimately investigated and what may be considered as knowledge (Avison & Fitzgerald, 1995). The choice of modeling constructs constrains how one can know or learn about reality—the basis of one's claim to knowledge (Klein & Lyytinen, 1983; Walsham, 1993). Users of the entity-relationship approach, for example, would focus on identifying entities and relationships whereas users of data-flow diagram (DFD) would emphasize the eliciting of processes, data flows, external entities, and data stores from the problem domain.

Despite the importance of modeling constructs, not much research has been done in this area. Most modeling constructs are introduced based on common sense, superficial observation, and intuition of researchers and practitioners. Theoretical foundation and empirical evidence are either non-existent or considered non-essential. For example, Coad and Yourdon (1991, p. 16) nicely summed up the practitioners' scant concern:

> "It would be intellectually satisfying to the authors if we could report that we studied the philosophical ideas behind methods of organization, from Socrates and Aristotle to Descartes and Kant. Then, based on the underlying methods human beings use, we could propose the basic constructs essential to an analysis method. But in truth we cannot say that, nor did we do it. "(emphasis added)

With this laissez-faire attitude, one can not help but cast doubts on the usefulness and importance of some of these modeling constructs. It is probable that some of these constructs are not actually actors in the modeling drama, but merely incidental artifacts, created by researchers to help them categorize their observations. These artifacts may play no significant role whatsoever in modeling the real world. A reference discipline to guide the design, construction, and adaptation of modeling constructs for information modeling methods is needed!

In this paper, we propose the use of cognitive psychology as a reference discipline in the engineering of methods and the studying of information modeling. Card et al. (1983, p. 1) wrote "advances in cognitive psychology and related sciences lead us to the conclusion that knowledge of human cognitive behavior is sufficiently advanced to enable its applications in computer science and other practical domains." Moray (1984) also argued for the use of knowledge accumulated in cognitive psychology to understand and solve applied problems. Researchers in human-computer interaction have demonstrated that such an effort is valuable and essential in building a scientific understanding of the human factors involved in end-users interaction with computers. We believe that similar effort will be useful in information modeling and method engineering.

Human Information-Processing System

To understand the representation and use of knowledge by humans, we need to approach it from a human information-processing perspective. The information-processing paradigm views thinking as a symbol-manipulating process and uses computer simulation as a way to build theories of thinking (Simon, 1979). It attempts to map the flow of information that a human is using in a defined situation (Gagne et al., 1993) and tries to understand the general changes of human behavior brought about by learning (Anderson 1995).

According to Newell and Simon (1972), all humans are information-processing systems (IPS) and hence come equipped with certain common basic features. Although some of the processes used by the system may be performed faster or better by some than by others, the nature of the system is the same. One of the popular and most well-known human information-processing model is the Adaptive Control of Thought (ACT) proposed by Anderson (1983, 1995) (see Figure 1).

An ACT production system consists of three memories: working, declarative, and production. Working memory contains the information that the system can currently access, consisting of information retrieved from long-term declarative memory as well as temporary structures deposited by encoding processes and the action of productions (Anderson, 1983). Declarative and production are long-term memory. The former is the facts and the latter is the processes or procedures that operate on facts to solve problems. Declarative knowledge is knowing that something is the case whereas procedural knowledge is knowing how to do something (Gagne et al., 1993).

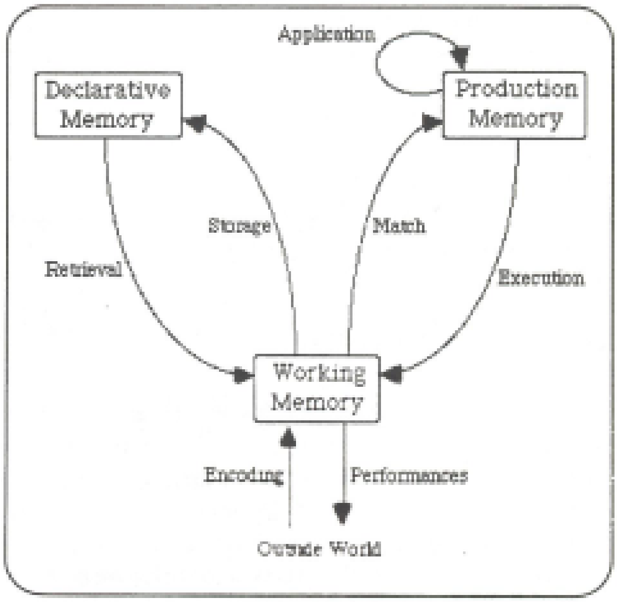

Figure 1: The ACT Architecture

Encoding deposits information about the outside world into working memory whereas performance converts commands in working memory into behavior. The storage process can create permanent records in declarative memory of the contents of working memory and can increase the strength of existing records in declarative memory. The retrieval process retrieves information from declarative memory into working memory. During the match process, data in working memory are put into correspondence with the conditions of productions. The execution process deposits the actions of matched productions into working memory. The whole process of production matching followed by execution is known as production application.

Working Memory

The working memory is activation based; it contains the activated portion of the declarative memory plus declarative structures generated by production firings and perception. Working memory is a temporary memory that cannot hold data over any extended duration. Information in this memory store decays within about 10 seconds (Murdock, 1961) unless it is rehearsed. In addition to its limited duration, working memory is also of limited capacity. Miller (1956) claimed that working memory holds 7± 2 units of information while Simon (1974) claimed that it holds only about 5 units. Whatever the actual number, the important point is that it is small. Because of its small size, working memory is often referred to as the "bottleneck" of the human information-processing system.

Declarative Knowledge

There are two types of long-term memory — declarative and procedural. The long-term declarative memory is represented in the form of a semantic net. A basic unit of declarative knowledge in the human information-processing system is proposition and is defined as the smallest unit of knowledge that can possess a truth value (Anderson, 1983). Complex units of knowledge are broken down into propositions. Propositions have at least two parts. The first is called the relation. Verbs and adjectives typically make up the relations of a proposition. The second part of the proposition is called the argument, which is determined by the nouns in the proposition. Arguments are given different names depending on their role in the proposition. Arguments may be subjects, objects, goals (destination), instruments (means), and recipients.

The declarative knowledge for the ER approach can be represented as propositions as shown below. Each proposition comprises a relation, followed by a list of arguments:

(i) represent, entity, rectangle
(ii) represent, relationship, diamond
(iii) comprise, ER, entity
(iv) comprise, ER, relationship

These four propositions can be depicted diagrammatically using Kintsch's system as shown in Figure 2.

In ACT, individual propositions can be combined into networks of propositions. The nodes of the propositional network stand for ideas, and the linkages represent associations among the ideas (Anderson, 1983). Figure 3 shows the network of propositions for the ER approach.

Procedural Knowledge

Unlike declarative knowledge, which is static, procedural knowledge is represented in the form of productions. Each piece of knowledge is called a production because it "produces" some bit of mental or physical behavior. Productions are formally represented as IF-THEN contingency statements in which the IF part of the statements contains the conditions that must exist for the rule to be fired and the THEN part contains the action that will be executed when the conditions are met. The productions are also known as condition-action pairs and are very similar to the IF-THEN statement in programming languages. For example, the following is the production rule for identifying a relationship construct in the ER model.

IF	Figure is a diamond shape
THEN	Figure represents a relationship construct

Productions can be combined to form a set. A production system, or production set, represents all of the steps in a mental or physical procedure. The productions in the production systems are related to one another by the goal structure. In other words, each production contributes in some way to achieve the final goal behavior. The use of goals and subgoals in productions creates a goal hierarchy that interrelates the productions into an organized set. For example, Table 1 shows a production system to understand an ER diagram.

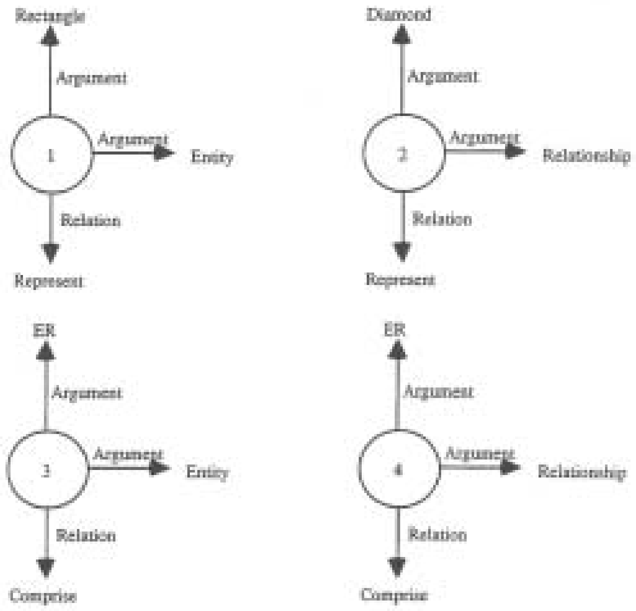

Figure 2: Diagrammatic Representation of Propositions for ER Approach

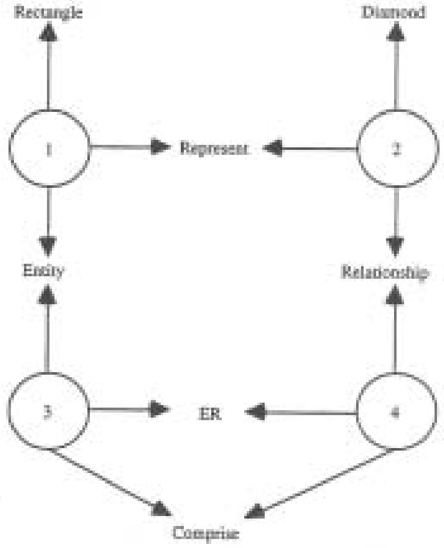

Figure 3: Network of Propositions for ER Approach

Domain-General Versus Domain-Specific. Procedural knowledge can be discussed from two dimensions. The first dimension refers to the degree to which procedural knowledge is tied to a specific domain, with the anchor points of the continuum being termed as domain-general and domain-specific (Gagne et al,. 1993). Domain-general knowledge is knowledge that is applicable across domains and domain-specific knowledge is specialized because it is specific to a particular domain. The term domain refers to any defined area of content and can vary in its breadth.

Degree of Automation. The second dimension can be labeled as "degree of automation" with the end points of the

P1	IF	Goal is to understand ER diagram
	THEN	Set subgoal of identifying meaningful chunks of information in ER diagram
P2	IF	Subgoal is to identify meaningful chunks of information in ER diagram
	THEN	Set subgoal of identifying entity in ER diagram and set subgoal of identifying relationship in ER diagram
P3	IF	Subgoal is to identify entity in ER diagram and symbol is a rectangle
	THEN	Symbol represents an entity
P4	IF	Subgoal is to identify relationship in ER diagram and symbol is a diamond
	THEN	Symbol represents a relationship

Table 1: A Production System to Understand an ER Diagram

continuum being called automated and controlled (or conscious) (Gagne et al., 1993). An automated process or procedure is one that consumes none or very few of the cognitive resources of the information-processing system. Controlled process, on the other hand, is knowledge that underlies deliberate thinking because it is under the conscious control of the thinker.

Implication on Information Modeling and Method Engineering

Researchers develop methods, and methods can be reengineered. By contrast, we cannot change the design of human information-processing system. Although the human subsystem is intelligent and adaptive, we cannot change the basic properties that define its strengths and weaknesses. If an information model is to be easy to understand and to function as an effective communication tool, the information modeling method must be compatible with our information processing characteristics. It is, therefore, important for us to consider this constraint when engineering methods and modeling information.

Limitation of Working Memory

The magic number 7 ± 2 has important implication on information modeling and method engineering. Firstly, if there are more than seven chunks of information required to be absorbed by the readers at any one time, the working memory capacity might be exceeded which means that some information might not be acquired. This is consistent with the recommendations by researchers and practitioners (e.g., Hawryszkiewycz, 1991) that there should be no more than seven processes on a data flow diagram. If this is true, some sort of leveling technique, similar to the one employed by data flow diagram, might be needed to limit the amount of information to an information model. Alternatively, the information model should be designed and laid out in such a way that at any time no more than seven pieces of information need to be processed together.

Secondly, if an information modeling method has more than seven modeling constructs, cognitive overload might occur. For instance, it would be difficult for a novice user to remember what each of the construct means if there are more than seven of them. The capacity of working memory serves as a threshold on the number of modeling constructs that can be incorporated into a modeling method. As such, the complexity of Unified Modeling Language (UML) and the number of different diagrams used in UML are causes for concern.

Declarative Knowledge

Declarative knowledge deals with facts. With respect to method engineering, declarative knowledge will consist of facts about the modeling constructs - what they are and what they represent. Since declarative knowledge is one type of long term memory, the larger the number of constructs in a modeling method, the more time is required to learn them. Training time is something that end-users are very reluctant to invest in. One of the reasons for the popularity of entity-relationship (ER) and object-oriented (OO) approaches is that a very small number of constructs is involved and that results in their simplicity. Also, using constructs that tap into existing declarative knowledge facilitates the transfer of knowledge and reduces the training time. For example, many researchers and practitioners claimed that entity-relationship and object-oriented approaches are intuitive and natural. Although research results vary, the constructs used by entity-relationship and object-oriented are undeniably simpler than a modeling method based on algebra or predicate logic, especially from the end-users' perspective.

Procedural Knowledge

Procedural knowledge is knowledge about how to do something. This is one of the most problematic areas in information modeling. For example, the most common criticism of object-oriented approach is the difficulty in identifying objects (e.g., Wand & Woo, 1993). The fuzziness of constructs is also a problem with entity-relationship modeling where one is often not sure when to use relationship, attribute, or even entity to represent something in the real world. For example, Goldstein and Storey (1990) found that users of an automated database design tool had difficulty distinguishing between relationships and attributes. Codd (1990) wrote "one person's entity is another person's relationship." It is, therefore, vital that when engineering methods, we need to precisely define the constructs and specify when and how to use a construct. Saying that the world is made up of objects does not help the analysts or the end-users in information modeling. Metamodeling, which describes the procedural and representational aspects of modeling methods, is a good way of documenting the procedural knowledge of a method. Forcing method engineers to perform metamodeling ensures that they think through and sort out the details involved in using a construct.

Domain-Specific Versus Domain-General Knowledge

Research has shown that experts in a specific domain have more and better conceptual or functional understanding of the domain, automated basic skills in the domain, and domain-specific problem-solving strategies. Domain experts, in contrast to novices, have the ability to perceive large meaningful patterns; highly procedural and goal oriented knowledge; less need for memory search and general processing; and specialized schema which drive performance. The possession of domain specific knowledge, however, is a problem during information modeling. To facilitate end-users' understanding of information model, it is important to use intuitive constructs that the end-users can relate to and recall easily. This has been the argument put forth for the goodness of ER and OO approaches.

Another aspect that is related to method engineering is the advantages of using domain-general constructs in methods. Domain-general constructs facilitate the transfer of knowledge from one method to another. As the degree of overlap of the modeling constructs that underlie two methods increases, transfer also increases. Situation method, which is an information system development method tuned to the situation of the project at hand, might be a problem from this perspective unless it makes use of well-known and easy to understand modeling constructs.

Degree of Automation

Working memory limitation impacts end-users much more significantly than analysts . For analysts, the meaning of each construct is in the long term memory, not the working memory. The knowledge has been internalized and automated by the analysts. Automated skills require little cognitive effort and allow the problem solver to perform necessary, routine mental operations without thinking much about them. On the other hand, remembering what each of the construct stands for would be a controlled process for the end-users. They need to consciously and deliberately think about them. Controlled process requires cognitive effort and is subjected to the limitation of working memory. Thus, when engineering methods, we need to consider the effect on end-users that are not at the automated stage in using modeling methods and will probably never attain the automated stage. Modeling methods, which are convoluted and highly technical, might be an excellent tool for analysts at automatic stage but will be a poor communication vehicle between analysts and end-users.

Conclusion

This research attempts to bring the wealth of knowledge in cognitive psychology to bear on the practical problems of information modeling and method engineering. The goal is to apply and adapt cognitive psychology theories and techniques for information modeling and method engineering research and help to span the gap between science and the practice of information modeling. In this paper, we look at some cognitive psychology theories and a popular cognitive architecture, Adaptive Control of Thoughts, and discuss their implication on information modeling and method engineering.

Acknowledgment: This research is supported by a Research Grant-In-Aid funding from the University of Nebraska-Lincoln (LWT/06-185-92501).

References

"Chaos" (1995). Standish Group Report on Information System Development. http://www.standishgroup.com/chaos.html.

Anderson, J.R. (1995). *Learning and Memory: An Integrated Approach.* John Wiley & Sons.

Anderson, J.R. (1983). *The Architecture of Cognition.* Cambridge, MA: Harvard University Press.

Avison, D.E., & Fitzgerald, G. (1995). *Information Systems Development: Methodologies, Techniques, and Tools.* (2nd ed.). McGraw-Hill, London.

Brinkkemper, S. (1996). Method Engineering: Engineering of Information Systems Development Methods and Tools. *Information & Software Technology*, 38, 275-280.

Bubenko, J.A. (1986). Information Systems Methodologies — A Research Review. In T.W. Olle, H.G. Sol, & A.A. Verrijn-Stuart (eds.), *Information Systems Design Methodologies: Improving the Practice* (pp. 289-318). Elsevier Science Publishers, North-Holland.

Card, S.K., Moran, T.P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Hillsdale, NJ: Erlbaum.

Chen, P.P. (1976). The Entity-Relationship Model: Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1, 1, 9-36.

Coad, P., & Yourdon, E. (1991). *Object-Oriented Analysis.* (2nd ed.). Prentice Hall.

Codd, E.F. (1990). *The Relational Model for Database Management:* Version 2. MA: Addison-Wesley.

Gagne, E.D., Yekovich, C.W., & Yekovich, F.R. (1993). *The Cognitive Psychology of School Learning.* Harper Colins.

Gibbs, W. (1994). Software's Chronic Crisis. *Scientific American,* September , 86-95.

Goldstein, R.C. & Storey, V. (1990). Some Findings on the Intuitiveness of Entity-Relationship Concepts. In F.H. Lochovsky (Ed.), *Entity-Relationship Approach to Database Design* (pp. 9-23). ER Institute.

Hawryszkiewycz, L.T. (1991). *Introduction to Systems Analysis and Design.* (2nd ed.) Prentice Hall.

Hirschheim, R., & Klein, H.K (1989). Four Paradigms of Information Systems Development. *Communications of the ACM*, 32, 10.

Hutchings, A.F., & Knox, S.T. (1995). Creating Products — Customers Demand. *Communications of the ACM*, 38, 5, 72 - 80.

Jayaratna, N. (1994). *Understanding and Evaluating Methodologies, NIMSAD: A Systemic Framework.* McGraw-Hill, Maidenhead.

Jones, C. (1986). *Programming Productivity.* New York: McGraw-Hill.

Kangassalo, H. (1990). Foundations of Conceptual Modeling: A Theory Construction View, In: *Information Modeling and*

Knowledge Bases, (pp. 20-29). IOS Press: Amsterdam.

Klein, H.K., & Lyytinen, K. (1983). The Poverty of Scientism in Information Systems. In: E. Mumford et al. (Eds.). *Research Methods in Information Systems.* North Holland: Amsterdam.

Kung, C.H., & Solvberg, A. (1986). Activity Modeling and Behavior Modeling. In: T.W. Olle, H.G. Sol, & A.A. Verrijn-Staut (Eds.), *Information Systems Design Methodologies: Improving the Practice* (pp. 145-171). North-Holland: Amsterdam.

Miller, G. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review*, 63, 81-97.

Moray, N. (1984). The Usefulness of Experimental Psychology. In: K. Lagerspetz & P. Niemi (Eds.), *Psychology in the 1990s* (pp. 225-235). North Holland.

Murdock, B.B., Jr. (1961). The retention of Individual Items. *Journal of Experimental Psychology*, 62, 618-625.

Mylopoulos, J. (1992). Conceptual Modeling and Telos. In: P. Loucopoulos & R. Zicari (Eds.), *Conceptual Modeling, Databases and Case* (pp. 49-68). New York, John Wiley & Sons.

Newell, A., & Simon, H.A. (1972). *Human Problem Solving.* Englewood Cliffs, NJ: Prentice Hall.

Oei, J.L.H., van Hemmen, L.J.G.T, Falkenberg, E.D., & Brinkkemper, S. (1992). *The Meta Model Hierarchy: A Framework for Information Systems Concepts and Techniques.* Katholieke Universiteit Nijmegen, Department of Informatics, Faculty of Mathematics and Informatics, Technical Report No. 92-17, 1-30.

Olle, T.W., Sol., H.G., & Verrijn-Stuart (1982). Information Systems Design Methodologies: A Comparative Review. *Proceedings of the CRIS 82 Conference.* North-Holland, Amsterdam.

Page-Jones, M. (1988). *The Practical Guide to Structured Systems Design.* (2nd ed.). Prentice Hall.

Rolland, C., & Cauvet, C. (1992). Trends and Perspectives in Conceptual Modeling. In: P. Loucopoulos & R. Zicari (eds.), *Conceptual Modeling, Databases and Case* (pp. 27-32). New York, John Wiley & Sons.

Rush, G. (1985). A Fast Way to Define System Requirements. *Computerworld,* XIX, 40, October 7.

Sernades, C., Fiadeiro, J., Meersman, R., & Sernadas, A. (1989). Proof-Theoretic Conceptual Modeling: The NIAM Case Study. In: E.D. Falkenberg and P. Lindgreen (Eds.), *Information System Concepts: An In-depth Analysis* (pp. 1-30). Elsevier Science Publishers B.V., North-Holland.

Siau, K. (1998). Method Engineering for Web Information Systems Development - Challenges and Issues. *Association for Information Systems 1998 Americas Conference (AIS'98).* Maryland, USA, August 14-16, 1017-1019.

Siau, K. (1999). Method Engineering: An Empirical Approach. Fourth CAiSE/IFIP8.1 *International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design* (EMMSAD'99). Heidelberg, Germany, June 14-15, I1-I12.

Siau, K., & Rossi, M. (1998). Evaluating Information Modeling Methods - Evaluation Techniques. *Thirty-first Hawaii International Conference on System Sciences* (HICSS-31). Big Island of Hawaii, January 6-9, Vol. V, 312-314.

Siau, K. (1998). The Psychology of Method Engineering. Third CAiSE/IFIP8.1 *International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design* (EMMSAD'98). Pisa, Italy, June 8-9, P1-P12.

Siau, K. (1997). Using GOMS for Evaluating Information Modeling Methods. *Second CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design* (EMMSAD'97). Barcelona, Spain, June 16-17, P1-P12.

Siau, K., Wand, Y., & Benbasat, I. (1996). Evaluating Information Modeling Methods — A Cognitive Perspective. *Workshop on Evaluation of Modeling Methods in Systems Analysis and Design.* Crete, Greece, M1-M13.

Siau, K., Wand, Y., & Benbasat, I. (1995). A Psychological Study on the Use of Relationship Concept — Some Preliminary Findings. *Lecture Notes in Computer Science — Advanced Information Systems Engineering*, 932, 341-354.

Siau, K., Wand, Y., & Benbasat, I. (1997). Information Modeling and Cognitive Biases - An Empirical Study on Modeling Experts. *Information Systems*, 22, 2&3, 155-170.

Siau, K.L., Wand, Y., & Benbasat, I. (1996). When Parents Need Not Have Children — Cognitive Biases in Information Modeling. *Lecture Notes in Computer Science — Advanced Information Systems Engineering*, 1080, 402-420.

Simon, H.A. (1974). How Big is a Chunk ? *Science,* 183, 482-488.

Simon, H.A. (1979). *Models of Thought. New Haven,* CT: Yale University Press.

Walsham, G. (1993). *Interpreting Information Systems in Organizations.* John Wiley & Sons, New York.

Wand, Y., & Weber, R. (1993). On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. *Journal of Information Systems,* 3, 217-237.

Wand, Y., & Woo, C. (1993). Object-Oriented Analysis — Is it Really that Simple? *Proceedings of the Workshop on Information Technologies and Systems.* Orlando.

Willumsen, G. (1993). Conceptual Modeling in IS Engineering, In: *Executable Conceptual Models in Information Systems Engineering,* Trondheim (pp. 11-21).

Yourdon, E. (1989). *Modern Structured Analysis.* Prentice Hall. 29

Dr. Keng Siau is an assistant professor in the Department of Management at the University of Nebraska, Lincoln (UNL). He received his Ph.D. degree in Business Administration from the University of British Columbia (UBC) where he majored in Management Information Systems and minored in Cognitive Psychology. He has published more than 15 journal articles and these articles have appeared in journals such as MISQ, IEEE Computer, Information Systems, Data Base, Journal of Information Technology, International Journal of Human-Computer Studies, Transactions on Information and Systems, and others. In addition, he has published over 30 refereed conference papers in proceedings such as ICIS, ECIS, HICSS, CAiSE, AIS, DSI, and HCI. His primary research interests are Systems Analysis and Design, Web-based Systems Development, and Human-Computer Interaction. Dr. Siau has served as the Organizing Chair and Program Chair for the International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD).

Reproduced with permission of copyright owner. Further reproduction prohibited without permission.