THE IMPACT OF USE CASE FORMAT ON THE UNDERSTANDING OF SYSTEM REQUIREMENTS BY NOVICE AND EXPERIENCED USERS

BALSAM A. MUSTAFA

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy (Computer Science)

Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia

JULY 2009

To My Father with Love

ACKNOWLEDGEMENT

Alhamdulillah! Finally I have finished my thesis! However, the completion of my thesis would not have been possible without the assistance of many people who gave their support in different ways. I would like to thank my supervisors: Prof. Safaai Bin Deris and Prof. Erik Arisholm, for your suggestions, continued encouragement, and patience to guide me through my research. I have learned a lot from you. I would like to thank Prof. Tajudin from the Faculty of Education at the University of Technology Malaysia for discussing with me the theories of Cognitive Psychology related to my research at the early days of my study.

A very special thank to Prof John Sweller from the University of New South Wales in Australia, he never disappointed me and was generous with me by his time and experience. My thanks and appreciation to Dr. Andrew Burton-Jones from the University of British Colombia in Canda for his continued encouragement and invaluable suggestions and discussions during this work, his responses to my emails at the difficult times were of great help.

I would like to express my gratitude to my family, my father and sisters. I have always needed to work hard to achieve my goals in life and they have always been there for me as an unwavering support. I dedicate this work to my father, to honor his love, prayers, and support during years.

ABSTRACT

The effective analysis and specification of requirements is critical in software development. Faults in the requirements may later have significant impact on the quality of the software system. Ineffective communication between users and developers is a major cause of failures of software projects. Use case model is a powerful and widely recognized tool for elicitation and specification of functional software requirements in object oriented methodology. It has been advocated as a way to negotiate and communicate requirements between system analysts and stakeholders. However, issues concerning the format, level of details and the communication capability of use cases are still unclear and debatable. This study uses theories from cognitive psychology on human understanding to derive hypotheses on the effect of the format of use case model on user understanding. In this study, comprehension of the functional requirements are compared between experienced and novice users. Particularly, the effect of differences in use case format on novice and experienced users performance in both familiar and unfamiliar domains were explored and if combining the textual description of a use case with diagrams of different levels of detail improves their understanding. Two controlled experiments were conducted; one to assess the performance of novice users, the other to assess more experienced users. The results of both experiments provide evidence that support the propositions that individuals who view *text with use case diagram (simple or detailed)* will develop higher level of understanding of the system requirements in less time when compared to individuals who view a *text only* model. The results of both experiments provide no evidence that support the propositions of the benefit of the simple diagram for improving novice users understanding, and the detailed diagram for aiding experienced users when combined with the text description. It is also found that neither the observed level of prior domain knowledge nor the observed level of analysis method knowledge has a significant effect on the level of "understanding" that users developed regarding a system requirements. Finally, our analysis shows no considerable differences in performance in the experiments tasks between novice and experienced users, which mean that the effect of experience on users understanding is still an open issue and needs further research in the future.

ABSTRAK

Analisis yang berkesan dan spesifikasi keperluan adalah perkara kritikal dalam pembangunan perisian komputer. Kesilapan pada keperluan akan memberi impak yang besar kepada kualiti sistem perisian yang dihasilkan. Komunikasi yang tidak berkesan di antara pengguna dan pembangun sistem adalah punca utama kepada kegagalan sesuatu projek pembangunan perisian. Kes gunaan adalah satu alatan yang berkesan dan digunakan secara meluas untuk mendapat spesifikasi keperluan fungsian dalam metodologi pembangunan berorientasikan objek. Ia diperakui sebagai satu kaedah untuk berunding dan berhubung antara penganalisis sistem dengan mereka yang berkepentingan dengan sistem tersebut. Namun begitu, isu-isu berkaitan format kes gunaan, tahap perincian dan keupayaan untuk menjadi alat komunikasi yang baik masih kurang jelas dan boleh dipertikaikan. Kajian ini menggunakan teori psikologi kognitif berasaskan pemahaman manusia untuk menghasilkan hipotesis mengenai kesan format model kes gunaan ke atas pemahaman pengguna. Dalam kajian ini, bagaimana seorang pengguna berpengalaman dalam kaedah pemodelan berbanding dengan pengguna baru dalam memahami keperluan fungsian yang didapati dari model dikaji. Secara khususnya kami meneroka kesan ke atas perbezaan pada format kes gunaan ke atas prestasi pengguna baru dan pengguna berpengalaman dalam domain yang di ketahui dan tidak diketahui, juga samaada kombinasi teks huraian kes gunaan dengan gambarajah pada tahap perincian yang berbeza boleh memperbaiki pemahaman mereka. Dua ekperimen makmal terkawal telah dijalankan; satu untuk menilai pencapaian pengguna baru dan satu lagi untuk menilai pencapaian pengguna berpengalaman. Hasil kedua-dua ekperimen di atas membuktikan yang pernyataan bahawa individu yang menggunakan teks model kes pengguna bersama gambarajah (ringkas atau terperinci) boleh meningkatkan tahap pemahaman keperluan sistem dalam masa yang lebih singkat berbanding dengan menggunakan model yang menggunakan teks sahaja. Hasil ekperimen tersebut juga tidak memberi bukti yang menyokong pernyataan bahawa faedah gambarajah ringkas adalah untuk pemahaman oleh pengguna baru, dan gambarajah terperinci adalah untuk membantu pengguna berpengalaman apabila disatukan dengan huraian teks sahaja. Kajian ini juga mendapati bahawa tahap pengetahuan awalan mengenai domain dan tahap pengetahuan kaedah analisis tidak memberi apa apa kesan yang bermakna kepada tahap pemahaman yang dikumpulkan oleh peserta mengenai keperluan sistem. Akhir sekali keputusan kajian kami menunjukkan tiada perbezaan yang ketara dari segi pencapaian dalam tugasan ekperimen antara pengguna baru dan pengguna berpengalaman. Ini bermakna kesan pengalaman ke atas pemahaman pengguna masih menjadi isu semasa dan perlu kajian lanjut.

TABLE OF CONTENTS

THE IMPACT OF EXPERIENCE ON UNDERSTANDING USE CASE MODELS BY EXPERIENCED USERS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Root Causes of Software Faults

Information technology (IT) is playing a crucial role in contemporary society. Information systems have become a vital component of successful businesses and organizations for the enormous advantages it has in easing the delivery of information. Moreover, information systems have been used by many organizations as strategic resources to attain or retain competitive advantages, as illustrated by the pervasive adoption of electronic commerce and enterprise systems (Siau & Xin, 2008). The increase production of software and information systems may indicates that system development projects are often successful, and that system development methods have matured enough to provide relatively low risk and high return business opportunities. However, according to industry studies, failure in software projects is common in many organizations. Studies pointed out that the errors which appear at the early stage of the software development can affect the reliability, cost, and safety of a system. Conclusions from these studies points to requirements specification as one major source of errors in software development and that the "*highest density of major defects found through the* *use of software inspections was during the requirements phase*" (Kelly & Kovington, 1994)

The report of the 'Consultancy Standish Group International Inc.' (2003 Chaos Report) a globally respected source of independent primary research and analysis of IT project performance, with an analysis of 13,522 IT projects revealed that 66% of all IT projects failed either because of over budgeting, over time, or failed to meet 20% or more of the business requirements for the system, 15% of the projects failed completely and were canceled prior completion (Figure 1.1). The average cost overrun was 43% and 82% of the challenged projects delivered with time overrun. However, these numbers are a significant improvement over the previous survey conducted in 1995.

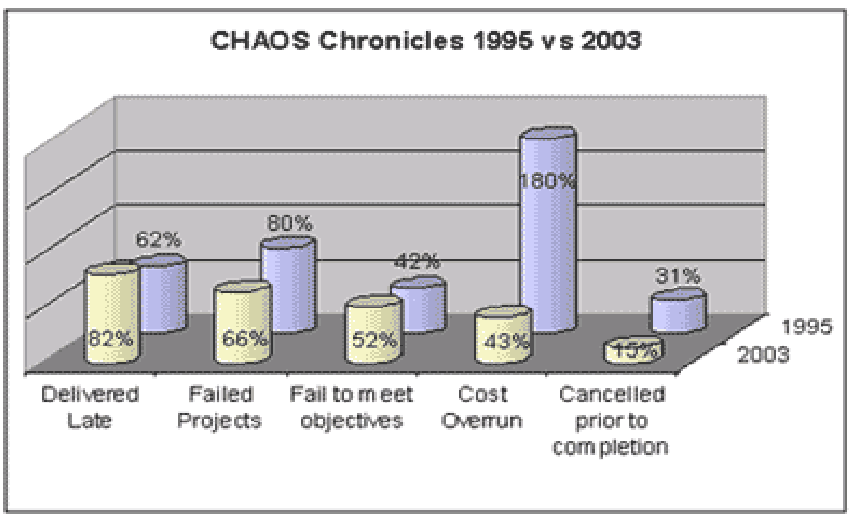

Figure 1.1 Chaos Report (2003)

Similar statistics were found in 2004 "CHAOS Report" (Johnson, 2006) when it was estimated that only 29% of software projects in large enterprises succeeded (i.e.

produced acceptable results that were delivered almost on-time and on-budget (Figure 1.2), 53% were “challenged” (significantly over budget and schedule), and 18% failed to deliver any substantial result. The projects that were in trouble have an average budget overrun of 56%. This represents a serious and chronic risk-control problem.

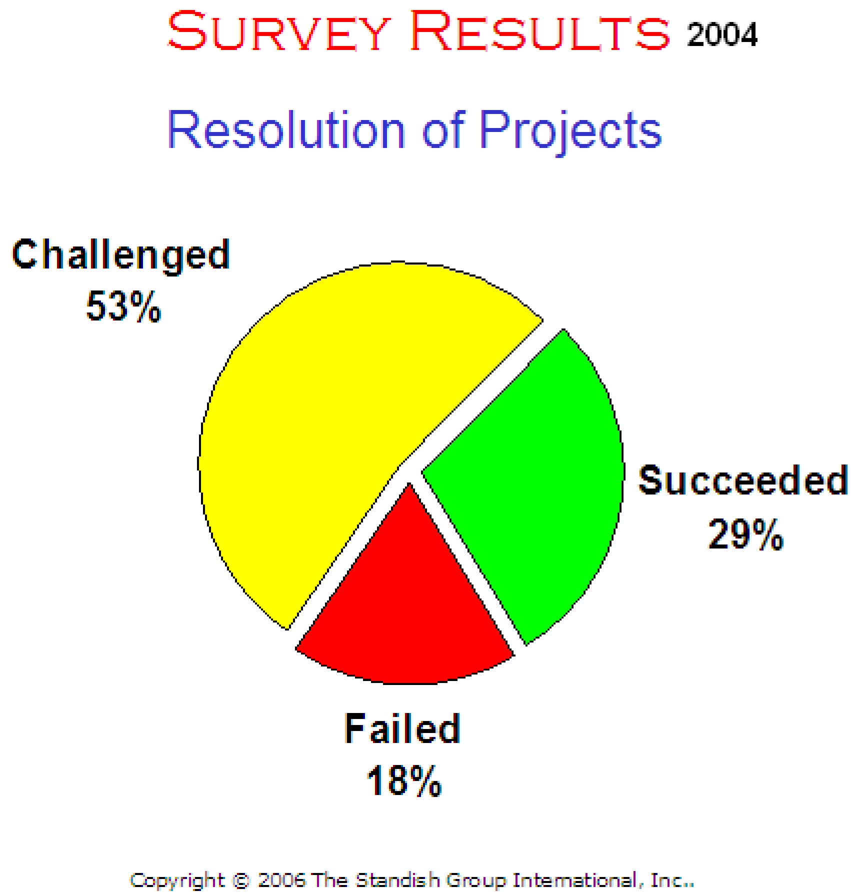

Figure 1.2 CHAOS Survey Results 2004

Other studies as the one conducted by “Consultant Capers Jones” in Marlborough, Mass., US stated:

"Large software systems...have one of the highest failure rates of any manufactured object in human history" (Ross, 2005).

Most IT experts agree that such failure occur more than they should. The failures are all over the world, they happen in every country, to large and small companies, in commercial, non profit and governmental organization, (Ross, 2005) (Appendix L). Charette (2005) indicates that the business and social costs of these failures run well into billions of dollars a year. Moreover he argued that small-medium size businesses spends about 4 to 5 percent of revenue on information technology while those that are highly IT dependent such as financial and telecommunication companies spending more than 10 percent on it. In organizations IT is now one of the largest corporate expenses after employee costs. Most of that money goes to fund new software projects in order to create better future for the organization and its customers.

1.2 Why Do Software Projects Fail?

Researchers over the past twenty years have studied the factors that could increase the high failure rate. A sample of these factors include the lack of early project management (Lyythinen *et al.*, 1996), underestimation of the cost and effort associated with a project (Jorgensen, 2006), and the lack of structured development techniques (methodology) for developing a system design (Coad & Yordon, 1991). One of the most common factors repeatedly attributed to system failure, is the lack of accurate communication in the early system development process (Holtzblatt & Beyer, 1995; Damian *et al*., 2006). Chaos survey (1995; 2003) helps highlight the historic problem of bridging the communication gap between users who understand business, and system experts who understand technology. The Standish Group Report (“Chaos1995”) provides useful list of factors that practitioners view as important in contributing to system development failure (Table 1.1):

In this table, many of the top ten factors (1,2,3,7,8) are concerned with communication in the early phase of the system development process. This phase is commonly referred to as “**System Analysis**”. The survey results suggest that when considering factors for project failure, practitioners recognized the importance of communication in early phases of planning.

Table 1.1: Factors Affecting System Development Failure "Chaos"1995

Rank	Project Failure Factor	% Responses
1	Lack of User Input	12.8
2	Incomplete Requirements / Specifications	12.3
3	Changing Requirements and Specifications	11.8
4	Lack of Executive support	7.5
5	Technological incompetence	7.0
6	Lack of Resources	6.4
7	Unrealistic expectations	5.9
8	Unclear Objectives	5.3
9	Unrealistic Time Frames	4.3
10	New Technology	3.7

1.3 Information Requirements Determination

It has been recognized that determining correct and complete information requirements is essential for designing an information system. As mentioned in section 1.2, many information systems failures can be attributed to a lack of clear and specific information requirements. Information requirements determination, also known as requirements definition, requirements gathering, requirements elicitation, and requirements engineering, is concerned about figuring what to build (Holtzblatt & Beyer, 1995). It includes any activity undertaken by systems developers and analysts to specify the required functions in the proposed system (Browne & Ramesh, 2002). Four tasks to be performed in requirements determination have been identified (Siau & Xin, 2008).

- *Requirements specification*: to understand the organizational situation that the system under consideration aims to improve and describe the needs and constraints of the system under development.
- *Requirements negotiation*: to establish an agreement on the requirements of the system among the various stakeholders involved in the process.
- *Requirements representation*: to develop a mapping of real-world needs onto a requirements model.
- *Requirements validation*: to ensure that the derived specification corresponds to the original stakeholder's needs and conforms to the internal and/or external constraints set by the enterprise and its environment.

During the development of software project systems analysts gather information about users' needs and the requirements expected in the proposed system and sometimes negotiate the requirements when they are unrealistic or confliction with others. This is done through direct discussions with system stakeholders and observations.

1.4 Conceptual Modeling

In the early stages of information systems development, analysts use conceptual modeling to build a representation of the domain understudy. High quality conceptual modeling work is important because it facilitates early detection and correction of system development errors (Wand & Weber, 2002).

The *American National Standards Institute* ANSI/SPARC definition of a conceptual data model is any model that is independent of the underlying hardware and software. Definitions of researchers in the area (Topi & Ramesh, 2002; Batra & Davis, 1992) did not also refer to any implementation details. Conceptual modeling is directly related to two tasks in requirements engineering- requirements representation and requirements validation. In requirements representation, conceptual models are created to map real-world needs. During requirements validation, users of the system verify whether their needs have been correctly specified by viewing the conceptual models (Siau & Xin, 2008).

Many conceptual modeling techniques have been devised in different methodologies. As the Unified Modeling Language (UML) widely adopted in modeling, many system analysts employ Use Case diagrams to model information requirements. Use cases are used to capture the user functional requirements. Each use case describes an element of the functionality of a system, which gives users a result of value. The sum of these use cases defines the total functionality of the system. The system is then designed and implemented to support the use cases (Siau & Xin, 2008).

1.5 Human Factors in IS Development

Researchers in the field of information systems have realized the need to pay attention to the human dimensions in systems development. In particular, when identifying requirements people must communicate effectively and share a common understanding of the work problems and the required solutions (Holtzblatt & Beyer, 1995). For example, when a system analyst represents a domain in a conceptual model, the model should be understandable by other developers, or end users. In this social process, human factors have significant implications. Human factors may include human's cognitive abilities, personality, knowledge, and motivation of the participants in the requirements determination process (Siau & Xin, 2008). Researchers in the area of conceptual data modeling pointed out the need for more research in human factors in information systems development. Topi and Ramesh (2002, P.11) stated that:

“ We need a better understanding of the psychological processes in data modeling and the ways the tools affect these processes. This will enable us to find a firm theoretical basis for human factors research on data modeling. Researchers in this area should be

interested not only in the characteristics of the current models, but the reasons underlying the potential performance differences between various approaches to data modeling.”

1.6 Motivation for the Research

During the analysis phase of information systems development, system analysts and developers capture and represent system requirements using conceptual models such as entity relationship diagram, Data flow diagram, Class diagram, Use case diagram. Considering that the requirements specification process must support effective communication between stakeholders who do not share common background (Holtzblatt & Beyer, 1995) and the fact that the failure of system development projects are attributed to faulty or incomplete requirements, it is then extremely important for the analyst and the success of the system to ensure that the conceptual models developed in the early phases of the system development must support communication between users and developers in defining and documenting system requirements as accurately as possible. The models should allow users to verify whether the analyst’s understanding of the system requirements reflects the reality as perceived by the users (Parsons & Cole, 2005).

From this perspective, research to evaluate techniques that represent requirements should focus on their capacity to facilitate this verification. To improve the performance and selection of analysis technique this work argues for the empirical evaluation of (Use Cases), one of the Unified Modeling Language (UML) models. UML is the standard for specifying, visualizing, constructing, and documenting the components of software systems (Booch *et al.*, 1999). The UML has been adopted as a standard for the object oriented modeling and has already found widespread popularity in various domains. However, UML diagrams are not widely rated in terms of understandability by end users and developers alike (Agarwal & Sinha, 2003). Use case models serve as basis for deriving other UML conceptual models. Thus it is important to ensure the quality of these models.

This research will investigate whether use case models are useful in the analysis process of a software system by enabling users to verify that their needs are documented. There are two **research questions** in this study:

1- Does the format of use case model influence the understanding and the patterns of performance, when individuals have to solve tasks on the basis of their previously acquired knowledge? And which use case, text only or text accompanied with diagram better support user understanding of the domain requirements?

2- How does the degree of detail in a Use Case diagram that accompanies text in a use case model affect user comprehension of the domain requirements?

1.7 Research Objectives

Use Cases is the technique used in UML for handling the functional requirements in a software development project and serve as a means of communication between different stakeholders in a project. The models developed as a result of requirements analysis are represented in diagrammatic form, supplemented with textual description for those parts that cannot be captured diagrammatically. In order to reduce the possibility for differences in understanding stem from using different formats for the use case models it would be useful to evaluate the extent of the different stakeholders' understanding to the model and also to detect differences in interpretation. This study has three objectives:

1- To present an empirical method for comparing different formats of use case model, and investigate the effect of the format on understanding the model by novice end users.

2- To extend the research by investigating high knowledge users to gain a comprehensive view of how stakeholders of different degrees of knowledge in the modeling technique may understand the use case models for the purpose of identifying system requirements.

3- To find out the impact of experience on understanding different formats of use case model.

To accomplish these objectives, a research instruments for the comparison of Use case formats was developed and two comparative studies were designed and implemented. The empirical instruments that used in lab experiments was developed from a combination of comprehension test, and verification test and finally a comparison between the performance of participants in the two studies was undertaken.

1.8 The Research Scope

In studying any research problem it is important to identify the research scope which is narrow enough to be effectively researched and yet touches an issue of significant potential impact. The researcher needs to focus on specific area for improvement. This research will focus on one of the techniques used within the unified modeling language (UML), the standard modeling language in the Object-Oriented methodology of information systems development. The UML has been widely accepted as the standard for object-oriented analysis and design (OOAD) (Kobryn, 1999). It has been an important part of the software development landscape since its introduction in 1997. UML models are used by professional developers to communicate their work to project stakeholders. For developers, eliciting high level goals early in the software development process is crucial. It focuses the developer on the problem domain and the needs of the stakeholders, rather than the implementation of the system. Despite the movement toward UML as a standard modeling language in practice, there appears very few empirical researches on the effectiveness of various modeling techniques in UML (Dobing & Parsons, 2000).

"***USE CASE***" model is one of the key modeling techniques of UML which is utilized in the analysis phase for capturing and describing the functional requirements of a system. Use Case models can also be harnessed in communication between stakeholders in project development. It is therefore essential that use case models support the development process and promote understanding of the requirements among stakeholders. There is currently no detailed account of the cognitive processes involved in understanding software requirements. While there are several studies on how programmers understand programs (Burkhardt *et al.*, 2002) we have not been able to find any study on human understanding of use case models. This lack of research on Use Case models understandability means that the guidelines and practices on how use cases should be designed to provide a base for better understanding of the requirements, is highly subjective. We intend to investigate how understandability of Use Case models may depend on the format of the model and how different stakeholders understand use case models.

Two reasons provided for focusing on Use Case technique as an analysis tool, first, solving problems in the early stages of the development process that employ these models can reduce the cost and effort of fixing these problems later. Second, although considerable attention is devoted to object oriented development methods and the standard language (UML) in information systems field, the evaluation of existing methods is not keeping pace with the rapid growth of the systems development methods (Siau & Tan, 2005). The implication of that is:

1. By failing to evaluate currently used object oriented methods, organizations may not clearly comprehend the usefulness and effectiveness of these methods.

2. The lack of object oriented methods evaluation including (UML), impede practitioners and researchers trying to understand the strength and weaknesses of various methods. This understanding is a critical knowledge for improving existing methods or designing new ones.

1.9 Research Implications

1.91 Theoretical Perspective

- In view of the paucity of empirical research on the effectiveness of various techniques of UML, it is significant from an academic perspective to independently evaluate the capabilities and limitations of UML techniques. Such evaluation can contribute to the development of theoretical underpinnings of UML, and to an improvement in its modeling power and usability.

- This work takes a step towards empirical validation of the theoretical basis regarding the understandability of (UML) models. It presents an empirical methodology with control, which can hopefully be used to study the effects of other factors on understanding the models, and other dependent variables as well as studying the rest of the modeling diagrams in UML.

- As UML become complex with each version, it is useful to focus on the core diagrams of UML particularly in teaching object oriented and software development methods. This work is relevant to instructors in universities and software practitioners as well.

1.92 Practical Perspective

From a practical standpoint, this work aids to the appropriate application of UML in systems development projects. It has implications for both method developers who want to know the strength and weaknesses of various methods, as well as for practitioners who want to use comparisons as a practical tool for selecting methods. From an analyst's perspective, this work assists in strategies at the beginning of the analysis phase, when decisions are made regarding the format and level of abstraction to be actually used to create requirements models that meet the user's needs. It would be useful if modelers could create models that convey accurate information with an easy-to-understand manner.

1.10 Organization of the Thesis

The remaining chapters are organized in the following manner. Chapter 2 outlines literature review of previous research in system analysis and design methods. Chapter 3 outlines the main topics related to use cases. Chapter 4 outlines the theoretical foundation for the proposed empirical method which led to develop the hypotheses that will form the basis for the two empirical studies in this research. Chapter 5 outlines the proposed design for both empirical studies and highlights the experimental procedures used in the study. Chapters 6 and 7 present the analysis of the empirical work, Chapter 8 presents general discussion and finally conclusions are made in chapter 9.

References

Agarwal, R., and Sinha, A. (2003). Object Oriented Modeling with UML: A Study of Developer's Perception. *Comm. of ACM*. 146: 248-256.

Aguirre-Urreta, M. , and Marakas, G. (2008). Comparing Conceptual Modeling Techniques: A Critical Review of the EER and OO Empirical Literature. *The Data Base for Advances in Information Systems*. 39(2): 9-32.

Ambler, S.W. (2004). *The Object Primer, Agile Model-Driven Development with UML 2.0*. 3rd edition. Cambridge: Cambridge University Press.

Anda, B., and Jorgenson, M. (2001). Understanding Use Case Models. *Proceedings of the 15th European Conference on Object-Oriented Programming (ECOOP2001)*. 18-22 June. Budapest, Hungary, 402-428.

Anderson, K., and Polanski, T. (2001). Object-Oriented Analysis and Design- CSCI 6448. On line at:

http://www.cs.colorado.edu/~kena/classes/6448/s01/homeworks/hw02.html .

http://www.cs.colorado.edu/~kena/classes/6448/s01/examples/tomp3.pdf . accessed 10/2/2009

Avison, D.E. and Fitzgerald, G.(2003). *Information systems Development: Methodologies, Techniques, and Tools*. 3rd edition. McGraw-Hill.

Bajaj, A. (2004). The Effect of Number of Concepts on The Read ability of Schemas. *Requirements Engineering*. 9:261-270.

Baker, T.(1999). *Doing Social Research*. 3rd edition. McGraw- Hill.

Bannert, M. (2002). Managing Cognitive Load- Recent Trends in Cognitive Load Theory. *Learning & Instruction*. 12:139-146.

Bartlett, F.C. (1932). *Remembering: A Study in Experimental Social Psychology*. Cambridge University Press.

Batra, D., Hoffer, J., and Bostrom, R. (1990). Comparing Representations with Relational and EER Models. *Comm. of ACM*. 33(2): 126-139.

Batra, D., and Davis, J. (1992). Conceptual data modeling in database design. *International Journal of Man Machine studies*. 37(1): 83-101.

Batra, D., and Satzinger, J. (2006). Contemporary Approaches and Techniques for the Systems Analyst. *Journal of Information Systems Education*. 17(3).

Bennet, T., and Wennberg, P. (2003). Maintaining Verification Test Consistency Between Executable Specifications and Embedded Software in a Virtual System Integration Laboratory Environment. *Proceedings of the 28th Annual NASA Goddard Software Engineering Workshop (SEW'03)*. 3-4 December. USA, 221-228.

Berard, E.(2006). Be Careful With Use Cases. The Object Agency Inc. On line at http://www.toa.com/pub/use_cases.htm, accessed 18/2/2009

Biddel, R. Nobel, J. and Tempero, E.(2002). Essential Use Cases and Responsibility in Object Oriented Development. *Proceeding of 25th Australasian Computer Science Conference (ACSC2002)*. 28 Jan.–1 Feb. Melbourne, Australia, 7-16.

Bodart, F., and Weber, R. (1996). Optional Properties Versus Sub Typing in Conceptual Modeling: A Theory and Empirical Test. *Proceedings of International Conference on Information Systems*. 16-18 December. Ohio, USA.

Bodart, F., Patel, A., and Weber, R. (2001). Should Optional Properties be Used in Conceptual Modeling? A Theory and Three Empirical Tests. *Information Systems Research*. 12(4):384- 405.

Boehm, B.(1981). *Software Engineering Economics*. Engelwood Cliffs, N.J.: Prentice-Hall.

Booch, G., Jacobson, I. and Rumbaugh, J. (1999). *The Unified Modeling Language, User Guide*. Addison-Wesley.

Briand, L., Wust, J., and Lounis, H. (1999). A Comprehensive Investigation of Quality Factors in Object-Oriented Designs: An Industrial Case Study. *Proceedings of the 21st International Conference on Software Engineering*. LA, California, 345-354.

Browne, G. J., and Ramesh, V. (2002). Improving Information Requirements Determination: a Cognitive Perspective. *Information Management*. 59(8):625-645.

Burkhardt, J. , DE´ Tienne, F. , and Wiedenbeck, S.(2002). Object-Oriented Program Comprehension: Effect of Expertise, Task and Phase. *Empirical Software Engineering*. 7:115-156.

Burton-Jones, A., and Meso, P.M. (2006). Conceptualizing Systems for Understanding: An Empirical Test Decomposition Principles in Object-Oriented Analysis. *Information Systems Research*. 17(1): 38-60.

Burton-Jones, A., and Meso, P.M. (2008). When Does a Conceptual Model Promote Understanding? The Effects of Decomposition Quality and Multiple Forms of Information on Users' Understanding of a Domain. Working paper. Sauder School of Business, University of British Colombia.

Butcher, K. (2006). Learning from Text with Diagram: Promoting Mental Model Development and Inference Generation. *Journal of Educational Psychology*. 98(1):182-197.

Carkenord, B. (2007). Book Review "The Complete Guide to Software Testing*". The Bridge Electronic Magazine*, Vol. Spring 2007, 16.

Carlson, R., Chandler, P., and Sweller, J. (2003). Learning and Understanding Science Instructional Material. *Educational Psychology*. 95: 629-640.

Carney, R., and Levin, J. (2002). Pictorial Illustrations Still Improve Students' Learning From Text. *Educational Psychology Review*. 14(1): 5-26.

Chaiyasut, P., and Shanks, G. (1994). Conceptual Data Modeling Process: A Study of Novice and Expert Data Modelers. *Proceedings of the 1st International Conference on Object-Role Modeling* (ORM-1). July. Magnetic Island, Australia, 310-333.

Chaos (1995). on line at http://net.educause.edu/ir/library/pdf/NCP08083B.pdf accessed 21/01/2009.

Chaos (2003) on line at http://www.costxpert.com/resource_center/Chaoscompared.pdf accessed 25/02/2009

Charbonneau, S. (2003). Modeling Use Cases with the Borland Suite of Tools. On line at: http://dn.codegear.com/article/images/31962/ModelingUseCases.pdf accessed 22/2/2009

Charette, R. (2005). Why Software Fails. IEEE Spectrum, Sep2005 issue. Online at http://www.spectrum.ieee.org/sep05/1685 accessed 21/01/2009

Cheng, P., Lowe, R.K, and Scaife, M. (2001). Cognitive Science Approaches to Understanding Diagrammatic Representations. *Artificial Intelligence Review*. 15: 79-94.

Chi, M., Glaser, R., and Rees, E. (1982). Expertise in Problem Solving. In R. Sternberg (Ed.) *Advances in the Psychology of Human Intelligence.* 1:7-75. NJ: Erlbaum.

Christensen, L. (1991). *Experimental Methodology*. 5th edition. Allyn and Bacon.

Coad, P., and Yourdon, E. (1991). *Object Oriented Analysis*. 2nd edition. Yourdon Press.

Cockburn, A. (2001). *Writing Effective Use Cases*. Addison Wesley.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*, 2nd edition. Lawrence Erlbaum Assoc Inc.

Constantine, L., and Lockwood, L.A.(2000). Structure and Style in Use Cases for User Interface Design. On line at http://www.foruse.com accessed 12/1/2009

Constantine, L., Lockwood, L.A.(2003). Usage-Centered Software Engineering: An Agile Approach to Integrating Users, User Interfaces, and Usability into Software Engineering Practice. *Proceedings of the 25th International Conference on Software Engineering*. 3-10 May. Portland, Oregon, 746-747.

Cook, T.D., Campbell, D. T. (1979). Quasi-Experimentation: Design and Analysis for Field Settings. Boston: Houghton Mifflin Company.

Cox, K. , Aurum, A. , and Jeffery, R. (2004). An Experiment in Inspecting the Quality of Use Case Descriptions. *Journal of Research and Practice in Information Technology*, 36(4): 211-227.

Damian, D. , Lanubile, F. , and Mallardo, T. (2006). The Role of Asynchronous Discussions in Increasing the Effectiveness of Remote Synchronous Requirements. *Proceedings of (ICSE'06)*. 20-28 May. Shanghai, China, 917-920.

Davrondjon, G. , and Wang, Y. (2003). The Cognitive Process of Comprehension. *Proceedings of the 2nd IEEE International Conference on Cognitive Informatics (ICCI'03)*. 18-20 August. London, UK, 1251-1254.

Dobing, B., and Parsons, J. (2000). Understanding the Role of Use Cases in UML: A Review and Research Agenda. *Journal of Database Management.* 11(4): 28-36.

Dobing, B., and Parsons, J. (2005). Current Practices in the Use of UML. *24th International Conference on Conceptual Modeling*. 26-28 October. Austria.

Douglass, B. P. (1998). UML for Systems Engineering. *Electronic Systems Technology & Design*. 37(11): 44-49.

Ericsson, K. A., and Kintsch, W. (1995). Long-Term Working Memory. *Psychological Review*. 102: 211-245.

Ericsson, K.L., and Lehmann, A.C. (1996). Expert and Exceptional Performance: Evidence of Maximal Adaptation to Task Constraints. *Annual Review of Psychology*. 47: 273-305.

Firesmith, D. (2004). Generating Complete, Unambiguous and Verifiable Requirements from Stories, Scenarios, and Use Cases. *Journal of Object Technology.* 3(10):27-39.

Fowler, M. (1997). *UML Distilled*. Addison-Wesley.

Franch, X. , and Botella, P. (1998). Putting Non-Functional Requirements into Software Architecture. *Proceedings of the 9th International Workshop on Software Specification and Design (IWSSD98)*. 16-18 April. Washington, USA:IEEE, 60-67.

Gemino, A. , and Wand, Y. (1997). Empirical Comparison of Object-Oriented and Dataflow Models. *Proceedings of the 18th International Conference of Information Systems*. 15-17 December. Atlanta, Georgia, 446- 447.

Gemino, A., and Wand, Y. (2003). Evaluating Modeling Techniques Based on Model of Learning. Comm. of ACM. 46(10):79-85.

Gemino, A. (2004). Empirical Comparisons of Animation and Narration in Requirements Validation. *Requirements Engineering*. 9:153-168

Gemino, A., and Wand, Y. (2004). Framework for Empirical Evaluation of Conceptual Modeling Techniques. *Requirements Engineering*. 9(4): 248-260.

Gemino, A., and Wand, Y. (2005).Complexity and Clarity in Conceptual Modeling: Comparison of Mandatory and Optional Properties. *Data & Knowledge Engineering*. 55:301-326.

Goldman, S. (2003). Learning in Complex Domains: when and why do multiple representations help?. *Learning and Instruction.* 13: 239-244.

Hair, J., Anderson, R., and Tatham, R. (1992). *Multivariate Data Analysis*. 3rd edition. New York: Macmillan Publishing.

Harmon, P., and Watson, M. (1998). *Understanding UML: The Developer's Guide with Web-based Application in Java*. San Francisco: Morgan Kaufmann.

Hinds, P. , and Patterson, M. (2001). Bothered by Abstraction: The Effect of Expertise on Knowledge Transfer and Subsequent Novice Performance. *Journal of Applied Psychology*. 86:1232-1243.

Holtzblatt, K., and Beyer, H. (1995). Requirements Gathering: The Human Factor. *Comm. of ACM*. 38(5): 31-33.

Hopkins, WG. (2000). Measures of Reliability in Sports Medicine and Science. *Sports Medicine*. 30:1-15.

Jacobson, I. (1987). Object Oriented Development in an Industrial Environment. *SIGPLAN Notices*. 22(12):183-191.

Jacobson, I., Christerson, M., Jonsson, P., and Overgard, G. (1992). *Object Oriented Software Engineering: A Use Case Driven Approach*. MA: Addison -Wesley.

Jacobson, I. (1994). *Object Oriented Software Engineering. A Use Case Driven Approach*. Addison-Wesley.

Jacobson, I., and Ericsson, M. (1995). *The Object Advantage: Business Process Reengineering with Object Technology*. MA: Addison -Wesley.

Jacobson, I., Booch, G., and Rambough, J. (1999). *The Unified Software Development Process.* MA: Addison-Wesley.

Johnson, J. (2006). My Life Is Failure: 100 Things You Should Know to be a Successful Project Leader. Standish Group International, West Yarmouth, MA.

Jorgensen, M., and Gruschke, T.(2006). Assessing Uncertainty of Software Development Effort Estimates: The Learning from Outcome Feedback. *11th IEEE International Software Metrics Symposium (METRICS'05)*. 19-22 September. Italy,1-10.

Kalyuga, S., Chandler, P., and Sweller, J. (1997). Levels of expertise and user-adapted formats of instructional presentations: A cognitive load approach. *Proceedings of 6th International Conference on User Modeling (UM97)*. 2-5 June. Sardinia, 261-272.

Kalyuga, S., Chandler, P., and Sweller, J. (1998). Levels of Expertise and Instructional Design. *Human Factors*. 40:1-17.

Kalyuga, S., Chandler, P., and Sweller, J. (2000). Incorporating Learner Experience into the Design of Multimedia Instruction. *Journal of Educational Psychology*. 92:126-136.

Kalyuga, S., Ayres, P. Chandler, P., and Sweller, J. (2003). The Expertise Reversal Effect. *Educational Psychology*. 38: 23-33.

Kelly, J. and Covington, R. (1994). Results of a Formal Method Demonstration Project. *Proceedings of WESCON'94.* 27-29 September. CA, USA, 62-66.

Khatri, V., Vessey, I. , Ramesh V., and Clay, P. (2006).Understanding Conceptual Schemas: Exploring the Role of Application and IS Domain knowledge. *Information Systems Research.* 17(1): 81-99.

Kim,Y., and March, S. (1995). Comparing Data Modeling Formalisms. *Comm. of ACM*. 38(4):103- 115.

Kirschner, P. (2002). Cognitive Load Theory: Implications of Cognitive Load Theory on the Design of Learning. *Learning and Instruction.* 12:1-10.

Kobryn, C. (1999). A Standardization Odyssey. *Comm. of ACM*. 42(10): 29-37.

Kobryn, C (2002). Will UML 2.0 be agile or awkward?. *Comm. of ACM.* 45(1):107-110.

Kozma, R. B. (2003). The Material Features of Multiple Representations and their Cognitive and Social Affordances for Science Understanding. *Learning and Instruction*. 13: 205-226.

Kulhavy, R., Lee, B. J., and Caterino, L. C. (1985). Conjoint Retention of Maps and Related Discourse. *Contemporary Educational Psychology*. 10:28-37.

Kulhavy, R. W., Stock, W. A., Peterson, S. E., Pridemore, D. R., and Klein, J. D. (1992). Using Maps to Retrieve Text: A test of Conjoint Retention. *Contemporary Educational Psychology*. 17: 56-70.

Lambiotte, J. G., Dansereau, D. F., Cross, D. R., and Reynolds, S. B. (1989). Multi Relational Semantic Maps. *Educational Psychology Review.* 1: 331-367.

Larman, G. (2001). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*. 2nd edition. Prentice- Hall.

Linos, P. (1993). Facilitating the Comprehension of C Programs. *Proceedings of the 2nd IEEE Workshop on Program Comprehension*. 8-9 July. Capri, Italy, 55-63.

Linic, J. (2007). Information Systems Modeling with Use Cases. *Proceedings of the ITI 2007- 29th International Conference on Information Technology Interfaces*. Cavtat, Croatia, 139-144.

Lowe, R.K. (1993). Constructing a Mental Representation from an Abstract Technical Diagram. *Learning and Instruction.* 3(3):157-179.

Lyythinen, K., Mathiassen, L. and Ropponen, J.(1996). A Framework for Software Risk Management. *Journal of Information Technology*.11(4).

Maiden, N., and Sutcliffe, A. (1992). Analysing the Novice Analyst: Cognitive Models in Software Engineering**.** *Journal of Man-Machine Studies*. 367: 719-740.

Malan, R., and Bredemeyer, D. (2001). Functional Requirements and Use Cases. BREDEMEYER CONSULTING. White paper, 1-10.

Mayer, R., Heiser, J., and Lonn, S. (2001). Cognitive Constraints on Multimedia Learning: When Presenting More Material Results in Less Understanding. *Journal of Educational Psychology*. 93:187-198.

Mayer, R. (1989). Models for Understanding. *Review of Educational Research.* 59: 43-64.

Mayer, R., and Gallini, J. (1990). When is an Illustration Worth Ten Thousand Words?. *Journal of Educational Psychology*, 82(4):715-726.

Mayer, R. E., and Sims, V. K. (1994). For Whom is a Picture Worth a Thousand Words? Extension of a Dual-Coding Theory of Multimedia Learning. *Educational Psychology*. 86(3):389- 401.

Mayer, R. E. (1999). Multimedia Aids to Problem-Solving Transfer. *Educational Research*. 31: 611-623.

Mayer, R. E. (2001). *Multimedia Learning*. New York: Cambridge University Press.

Mayer, R. E., and Moreno, R. (2002). Aids to Computer-Based Multimedia Learning. *Learning and Instruction*. 12:107-119.

Mayer, R. (2003). The Promise of Multimedia: Using the Same Instructional Design Methods across Different Media. *Learning and Instruction.* 13:125-139.

Mayer, R. (2005). *The Cambridge Hand Book of Multimedia Learning*. Cambridge: Cambridge University Press.

McBreen, P. (1998). Using Use Cases for Requirements Capture. McBreen Consulting. Available on line at http://www.mcbreen.ab.ca/ accessed 10/1/2009.

McNamara, D.S, Kintsch, W.(1996). Learning From Text, Effects of Prior Knowledge and Text Coherence. *Discourse Processes.* 22: 247-288.

Merrick, P., and Barrow, P. (2005). The Rationale for OO Associations in Use Case Modeling. *Journal of Object Technology*. 4(9):123-142.

Miller, G. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review*. 63(2): 81-97.

Moody, D. (2002). Complexity Effects on End User Understanding of Data Models: An Experimental Comparison of Large Data Models. *The 6th European Conference on Information Systems (ECIS'2002)*. 6-8 June. Gdansk, Poland, 482-496.

Moody, D. (2004). Cognitive Load Effects on End User Understanding of Conceptual Models.: An Experimental Analysis. LNCS (3255), 129-143. Springler-Verlag.

Moor, G.C., and Benbasat, I. (1991). Development of an Instrument to Measure Perceptions of Adopting an Information Technology Innovations. *Information Systems Research.* 2(3):192- 222.

Nottis, M. , . Kastner, E.(2005). The Effect of Instructional Modality and Prior Knowledge on Learning Point Group Symmetry. *Journal of Science Education and Technology*. 14(1):51-58.

Novick, L. (2006). The Importance of Both Diagrammatic Conventions and Domain-Specific Knowledge for Diagram Literacy in Science: The Hierarchy as an Illustrative Case. LNAI (4045), 1-11. Springer-Verlag.

Opdahl, A., and Barbier, F. (2001). Ontological Analysis of Whole-Part Relationships in OO-models. *Information Software Technology*. 43: 387-399.

Oxford University (2002). *Concise OXFORD English Dictionary*. 10th edition. New York: Oxford University Press Inc.

Paas, F., Renkl, A., and Sweller, J. (2003). Cognitive Load Theory and Instructional Design: Recent developments. *Educational Psychologist*. 38:1- 4.

Paas, F., Renkl, A., and Sweller, J. (2004). Cognitive Load Theory: Instructional Implications of the Interaction between Information Structures and Cognitive Architecture. *Instruction and Science*. 32:1-8.

Paivio, A. (1986). *Mental Representation: A dual coding approach.* New York: Oxford University Press.

Parsons, J, and Cole, L. (2005) What Do the Pictures Mean? Guidelines for Experimental Evaluation of Representation Fidelity in Diagrammatical Conceptual Modeling Techniques. *Data & Knowledge Engineering*. 55:327-342.

Peleg, M., and Dori, D. (2000). The Model Multiplicity Problem: Experimenting with Real-Time Specification Methods. *IEEE Transactions on Software Engineering*. 26(8): 742-759.

Pollock, E., Chandler, P., and Sweller, J . (2002). Assimilating Complex Information. *Learning and Instruction*. 12: 61–86.

Ramsey, R., Atwood, M. , and Van Doren, J. (1993). Flowcharts Versus Program Design Languages: An Experimental Comparison. *Comm. of AC*M. 26(6): 445-449.

Ratcliffe, M., Budgen, D.(2005). The Application of Use Cases in Systems Analysis and design specification. *Information and Software Technology*. 47: 623-641.

Rational Unified Process Documentation; IBM Corp. On line at: http://www128.ibm.com/developerworks/rational/ accessed 12/12/2008.

Reed, S. (2004). *Cognition: Theory and Application*. 6th edition. Wadsworth Publication.

Robinson, D. H., Robinson, S. L., and Katayama, A. D. (1999). When Words Are Represented in Memory Like Pictures: Evidence for Spatial Encoding of Study Materials. *Contemporary Educational Psychology*. 24:38–54.

Ross, PH.(2005).The Exterminator. IEEE Spectrum. Sep. 2005 issue. On line at http://www.spectrum.ieee.org/sep05/1454 accessed 2/2/2009.

Rosenburg, D., and Scott, K. (1999). Use *Case Driven Object Modeling With UML: A Practical Approach*. Addison-Wesley.

Rumbaugh J, Jacobson I, and Booch G. (1999). *Unified Modeling Language Reference Manual*. MA: Addison Wesley.

Scaife, M., and Rogers, Y. (1996). External Cognition: How do Graphical Representations Work. *Journal of Human Computer Studies*. 45:185-213.

Schneider,G., and Winters, J. (2001). *Applying Use Cases: A practical guide*. 2nd edition. Addison Wesley.

Schnotz, W. (2002). Towards an Integrated View of Learning from Text and Visual Displays. *Educational Psychology Review*. 14:101-119.

Scott, B., and Schwartz, N. (2007). Navigational Spatial Displays: The Role of Meta Cognition as Cognitive Load. *Learning and Instruction*. 17:89-105.

Shanks, G., Simsion, G., and Rembach, M. (1993). The Role of Experience in Conceptual Data Modeling**.** *Proceedings of 4th Australian Conference on Information Systems*. University of Queensland, Australia, 365-378.

Shanks, G. (1997). Conceptual Data Modeling: an Empirical Study of Expert and Novice Data Modelers. *Australian Journal of Information Systems*. 4(2):63-73.

Siau, K., Wand, Y., and Benbasat, I., (1996). Evaluating Information Modeling Methods- A Cognitive Perspective. *Workshop on Evaluation of Modeling Methods in System Analysis and Design (EMMSAD'96')*. 20-21 May. Crete, Greece, M1-M13.

Siau, K. (1999). Information Modeling and Method Engineering: A Psychological Perspective. *Journal of Database Management*. 10(4): 44-50.

Siau, K., and CAO, Q. (2001). Unified Modeling Language (UML)- A Complexity Analysis. *Journal of Database Management*. Jan-Mar: 26- 34.

Siau, K., and Loo, P. (2002). Difficulties in Learning UML: A Concept Mapping Analysis. *Proceeding of the 7th CAiSE/IFIP8.1 International workshop on evaluation of Modeling Methods.* Toronto,Canada, 102-108.

Siau, K., and Lee, L. (2004). Are Use Case and Class Diagram Complementary in Requirements Analysis?. *Requirements Engineering*. 9:229-237.

Siau, K., and Tan, X. (2005). Improving the Quality of Conceptual Modeling Using Cognitive Mapping Techniques. *Data & Knowledge Engineering*. 55:343-365.

Siau, K., Ericsson, J., and Lee, L.(2005). Complexity of UML: Theoretical vs. Practical Complexity: The Case of UML. *Journal of Database Management*. 16(3): 40-57.

Siau, K., and Xin, T. (2008). Mapping Techniques in Information Systems Development. *Journal of Computer Information Systems*.1:1-23.

Simon, H., and Larkin, J. (1987). Why a Diagram (Some times) Worth Ten Thousand Words. *Cognitive Science*. 11:65-99.

Simon, D.P., and Simon, H.A. (1978). Individual Differences in Solving Physics Problems. In R.S. Siegler (Ed.), Children's thinking: What develops? Chap.13: 325-361.

Solso, R. (1995). *Cognitive Psychology*, 4th edition. Allyn & Bacon.

Stevens, J. (1992). *Applied Multivariate Statistics for the Social Sciences*, 2nd edition, Lawrence Erlbaum Associates.

Straub, D. (1989). Validating Instruments in MIS Research. *Management Information Systems Quarterly*. 9:147-169.

Sutherland, R., Pipe, M.E., and Schick, K. (2003). Knowing in Advance: The Impact of Prior Event Information on Memory and Event Knowledge. *Journal of Experimental Child Psychology*. 84:244-263.

Sweller, J., Mawer, R., and Ward, M. (1983). Development of Expertise in Mathematical Problem Solving. *Journal of Experimental Psychology*. 112(4):639-661.

Sweller, J. (1988). Cognitive Load During Problem Solving: Effects on Learning. *Cognitive Science*. 12:257-285.

Sweller, J., Chandler, P, Tierney, P., and Cooper, M (1990). Cognitive Loads as a Factor in the Structuring of Technical Material. *Journal of Experimental Psychology*. 119:176-192.

Sweller, J, and Chandler, P. (1994). Why Some Material is Difficult to Learn?. *Cognition & Instruction*. 12(3):185-233.

Sweller, J., Van Merrie¨nboer, J. J., and Paas, F.G. (1998). Cognitive Architecture and Instructional Design. *Educational Psychology Review*. 10(3):251-296.

Sweller, J. (2003). Evolution of Human Cognitive Architecture. In Ross, B. (ed.), Psychology of Learning and Motivation, 43, 215-266, Academic Press, San Diego.

Sweller, J. (2004). Instructional Design Consequences of an Analogy between Evolution by Natural Selection and Human Cognitive Architecture. *Instruction & Science*. 23(1/2): 9-31.

Tabachnick, B., and Fidell, L. (2001). *Using Multivariate Statistics*. 4th edition. MA: Allyn &Bacon.

Topi, H., and Ramesh, V. (2002). Human Factors on Data Modeling: A Review of Prior Research. *Journal of Database Management*. 13(2):3-19.

Van Merrienboer, J.J., and Sweller, J.(2005). Cognitive Load Theory and Complex Learning: Recent Development and Future Directions. *Educational Psychology*. 17(2): 147-175.

Vessey, I., Jarvenpaa, S., and Tractinsky, N. (1992). Evaluation of Vendor Products: CASE Tools as Methodology Companions. *Comm. of ACM*. 35(4):90-105.

Vessey, I., and Conger, S. (1994). Requirements Specification: Learning Object, Process, and Data Methodologies. *Comm. of ACM*. 37(5):102-113.

Voss, J. F., Greene, T.R., Post, T.A., and Penner, B.C. (1983). Problem-Solving Skill in the Social Sciences. *The Psychology of Learning and Motivation*. 17:165-213.

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

Wand, Y., and Weber, R. (2002). Research Commentary: Information Systems and Conceptual Modeling — A Research Agenda. *Information Systems Research.* 13(4): 363-376.

Whitten, L., Bentley, L.D., and Dittman, K.C. (2004). *System Analysis and Design Methods*, 6th edition. McGraw -Hill.

Whittle, J. (2000). Formal Approaches to Systems Analysis Using UML: An Overview. *Journal of Database Management*. 11(4): 4-13.

Winn, W. (1989). The Role of Graphics in Training Documents: Toward an Explanatory Theory of How They Communicate. *IEEE Transactions on Professional Communication*. 32 (4):300-309.

Winn, W. (1990). Encoding and Retrieval of Information in Maps and Diagrams. *IEEE Transactions on Professional Communication*. 33(3):103-107.

Winn, W., and Li, T-Z. (1991). Diagrams as Aids to Problem Solving: Their Role in Facilitating Search and Computation. *Educational Technology Research and Development*. 39(1):17-29.

Winn, W. (1991). Learning from Maps and Diagrams. *Educational Psychology Review*. 3: 211-247.