IGI PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.igi-global.com

ITJ 4655

This paper appears in the publication, Journal of Database Management, Volume 20, Issue 1
edited by **Keng Siau © 2009, IGI Global**

Use Case Diagrams in Support of Use Case Modeling:
Deriving Understanding from the Picture

Andrew Gemino, Simon Fraser University, Canada
Drew Parker, Simon Fraser University, Canada

ABSTRACT

Use case modeling in the Unified Modeling Language (UML) is a popular text-based tool for systems analysis and design. Use cases can be used with or without supporting use case diagrams. This paper uses an experiment to explore the effectiveness of including a use case diagram with a set of use cases. The Cognitive Theory of Multimedia Learning is used to hypothesize that the use case diagram improves the effectiveness of use cases for novice users by providing visual cues aiding model viewers in selecting and integrating relevant information. The level of understanding developed by participants viewing either uses cases or use cases with a supporting use case diagram was measured using comprehension, retention, and problem solving tasks. Results showed that participants viewing the use cases with the supporting diagram developed a significantly higher level of understanding, as measured by performance on the problem solving task, than participants provided with use cases alone. This analysis suggests practitioners should consider combining a visual representation, such as a use case diagram, with text-based use cases to achieve higher levels of understanding in persons viewing these descriptions.

Keywords: *conceptual modeling; system analysis; unified modeling language (UML); use case modeling*

INTRODUCTION

The Unified Modeling Language (UML) offers a standard language specification to support an object-oriented approach to systems analysis and design. The use case is a text-based description defined in the UML that provides a structured sequence of processes within a system (Jacobson, Ericsson, & Jacobson, 1994). Use cases are a popular modeling technique amongst UML practitioners (Batra, 2008; Dobing & Parsons, 2008) and use cases have received significant research attention (Burton-Jones & Meso, 2006; Siau & Loo, 2006). While text is a rich, familiar and expressive modeling tool, the exclusive use of text across multiple use cases may be difficult for users to conceptualize, particularly as the size of the modeled system increases.

Given the popularity of use cases, it seems reasonable to consider whether a diagram in support of use case modeling, specifically in this case the use case diagram, provides a more effective method for communicating system analysis information than text-based use cases alone. This question is of interest as Dobing and Parsons (2000) found that while use case narratives and use case diagrams were the UML tools most likely to be used in interacting with users, that 42 percent of respondents indicated that use case diagrams provide insufficient value to justify their cost. In a subsequent survey, Dobing and Parsons (2008) found that the use case diagram seems to be gaining popularity: for client validation, implementation, documentation, and clarification, respondents believed the use case diagram to be at least moderately useful.

Communication of analysis information is recognized as an important factor in information system development success. The oft-quoted CHAOS report (Standish Group, 1994) and more recent reports (Charette, 2005) suggest that poorly defined system requirements and poor communication with users remain important inhibitors to development success. This paper hypothesizes that understanding of text-based tools such as use cases could be significantly enhanced by incorporating diagrams conveying the information in a graphical format. The Cognitive Theory of Multimedia Learning (CTML) developed by Mayer (2001) recognizes that both graphical and textual cognitive channels are involved in developing understanding and supports this assertion.

An experiment was undertaken to compare the effectiveness of use cases with and without supporting use case diagrams in conceptual modeling. To accomplish this, we take the view that techniques should be compared on how well they support the development of an understanding of the domain they represent (Gemino & Wand, 2003). The CTML (Mayer, 2001) is used to hypothesize that diagrams improve the effectiveness of use case delivery by providing visual cues aiding model viewers in selecting and integrating relevant domain information into effective cognitive representations. To test understanding, we use a problem solving task (Bodart, Patel, Sim, & Weber, 2001; Burton-Jones & Meso, 2006; Gemino, 1999) that requires reasoning about the domain and focuses attention on higher levels of understanding.

BACKGROUND

The term "use case" refers to a complete sequence of events in the system as understood from a user's perspective. In other words, a use case represents the actions associated with an actor's "use" of the system (Jacobson et. al., 1994). The use case has become an important part of object-oriented analysis methods

(Siau & Cao, 2001) and is prevalent in early requirements analysis (Dobing & Parsons, 2008).

Kobryn (1999) has argued that use cases include simple and natural notations that are easy to understand for stakeholders, analysts, and designers. This simplicity makes use cases ideal tools for interacting with users. A key to the success of the use case remains the lack of formalism enabling stakeholders and analysts to communicate (Jacobson, Booch, & Rumbaugh, 1999). Improving the effective communication between designers, analysts and users addresses a primary factor in system development failure and increases the chances the resulting system will address the business challenges it was intended to support.

Previous Research in UML Use Case Modeling

A significant amount of research has studied the UML (Agarwal, De, & Sinha, 2003; Burton Jones & Meso, 2006; Evermann & Wand, 2005; Fedorowicz & Villeneuve, 1999; Siau & Cao, 2001; Siau & Loo, 2006). Much of the focus has been placed on theoretical work relating to diagramming techniques (Douglass, 1998; Halpin & Bloesch, 1999; Mellor, 2002). While much has been said regarding the potential benefits of use case modeling (Jacobson et. al., 1999; Kobryn, 1999), surprisingly little empirical research has considered these claims. For example, Dobing and Parsons (2006) found little or no empirical research on the effectiveness of use case modeling.

While UML modeling is popular, it also has critics. Douglass (1998) and Siau, Erickson, and Lee (2005) have suggested that the UML is large and can be complex for users. Halpin and Bloesch (1999) suggested UML models are designed for software engineering and are less suitable for validation of conceptual models. Dori (2002) and Shoval and Kabeli (2005) have suggested that it is difficult in UML to integrate structural and process elements of system designs. In regard to use cases, Dobing and Parsons (2006) suggested that use case modeling faces two significant challenges. One challenge is that use cases tend to isolate stakeholders from object class models. This results in a lack of information on classifications and categories within the system. They argue that information in the Class Diagram is valuable in developing understanding and is not provided by use cases. A second challenge is the lack of formalism, which allows use cases to mix conceptual, design and implementation details in the same description. This mixture of design and conceptual elements may cause confusion for stakeholders and reduce the effectiveness of the stakeholder/analyst communication. Both of these challenges offer an opportunity to extend understanding with a diagram. These criticisms suggest the need for empirical evidence (Johnson, 2002; Wand & Weber, 2002).

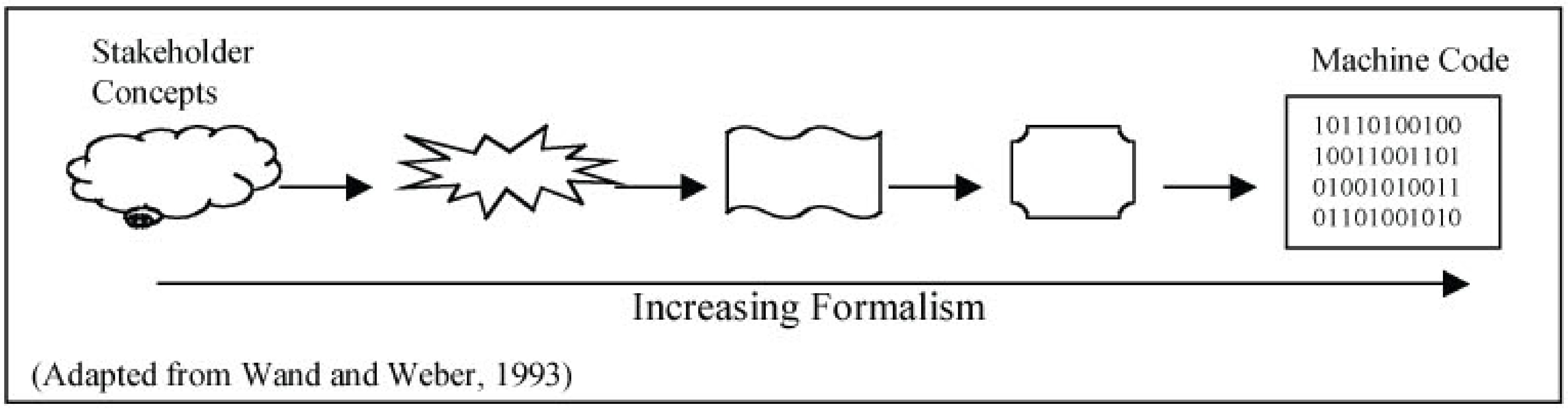
Separating Conceptual Modeling from Requirements Engineering

To understand how uses cases support conceptual modeling, it is important to outline basic elements in system development process. The information system development process can be viewed as a series of increasingly formal representations ending in machine executable code (Wand & Weber, 1993). This process is depicted in Figure 1.

Three generic roles involved in this development process include stakeholders, analysts and developers as shown in Figure 2. The least formal representations of the system are the concepts held by stakeholders. Analysts can be portrayed as interacting with stakeholders to develop initial representations of the system, which are referred to here as conceptual models (Everman, 2005; Wand & Weber, 2002). It should be noted that software engineering practices, such as Agile Development, that do not aim to develop structured requirements would not assume a separation between developer and analyst nor separate conceptual modeling from software development (Angioni, Carboni, Pinna, Sanna, Serra, & Soro, 2006; Meso & Jain, 2006). Still, the evolution of this formalization would hold true. An iterative process produces conceptual models then can serve as a foundation for the development of more formal requirements in a process of requirements engineering. Analysts develop formal requirements primarily to communicate system details with developers. Developers can then use formalized requirements as an input for the software construction process to develop the eventual machine code for the system (the system artifact).

The role of the analyst in this process is to communicate system details in such a way as to develop a common understanding of the system between developers, analysts, and stakeholders. This view suggests analysts are involved in two distinct processes. The first involves interacting with stakeholders to develop an understanding of the system. This is conceptual modeling (CM). CM involves eliciting initial ideas about the system, representing them, and then hav-

Figure 1. System development as a process of increasingly formal representations



(Adapted from Wand and Weber, 1993)

ing stakeholders interpret and validate these requirements. The second process formalizes this conceptual understanding into a set of requirements. This second process is defined here as requirements formalization (RF). CM and RF are related processes that facilitate the common objective to reason and communicate about a domain.

Because the processes are related, the same techniques are often suggested for use in both CM and RF. The target audiences for CM and RF, however, differ in both experience with the system and experience with formal modeling languages such as the UML. It is not clear that the same modeling techniques will be useful for both audiences. Use cases have often been suggested as useful tools for interacting with stakeholders, and hence could support CM, but they can also inform developers about process issues. In this study, the focus is placed on use cases as they pertain to the process of conceptual modeling and the interaction between analysts and stakeholders.

THEORETICAL FOUNDATIONS

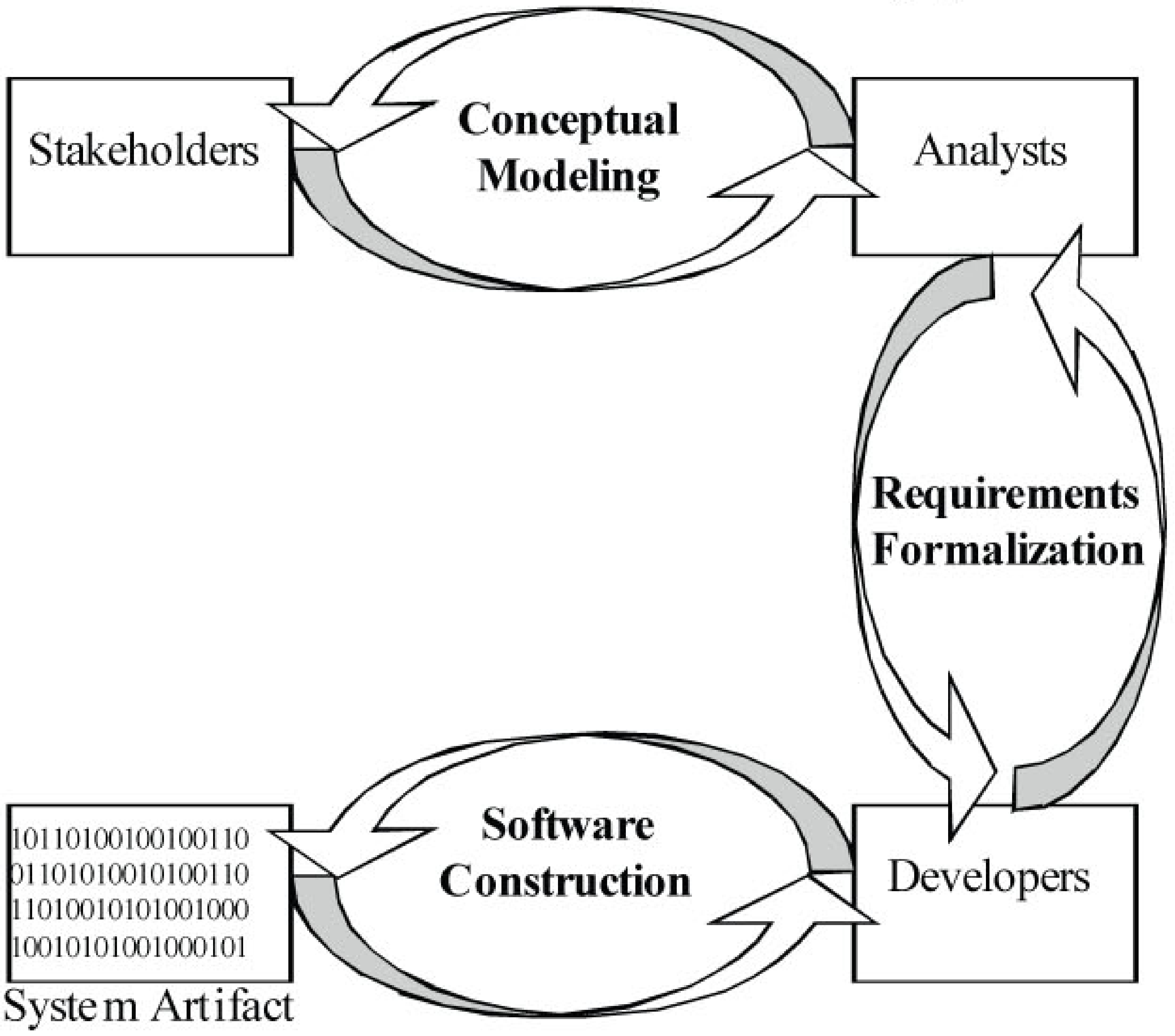
CM involves the elicitation and collection of domain information to develop understanding and support communication and can be viewed as a process of learning (Gemino & Wand, 2003). This is true for the person developing the model as well as the person viewing it. The design of CM techniques may be informed, therefore, by theories of how humans develop understanding from the graphics and words they are presented with. Mayer (2001) suggests two contrasting views of learning—information acquisition and knowledge construction. These views are discussed briefly below.

Information Acquisition and Knowledge Construction

Learning as information acquisition suggests learning is adding to long-term memory. The person looking at the model can be viewed as an “empty vessel” that can be filled with the information provided in the model. The model creator presents information to model viewers. The model viewer receives information and stores it in memory. The responsibility for learning in this view rests on the model creator to deliver appropriate information. The goal is to deliver required information efficiently. In the information acquisition view, the conceptual model is a standard vehicle for efficient information delivery to people viewing the model.

An alternative view of the learning process is that of knowledge construction. This view suggests knowledge is personally constructed. Two model viewers presented with the same conceptual model may come away with different learned outcomes. This occurs as the model viewers attempt to make sense of the information presented, There is a process where model viewers integrate new information provided by the model with information that has each person has available in long term memory. It is at this integration point where knowledge is constructed. Knowledge construction

Figure 2. Roles and interaction in the analysis and design process



suggests the model viewer is an active sense maker rather than a passive receiver of information. The model creator's role in the knowledge construction view is to assist the model viewer in their sense-making by not only presenting information but also determining what information to pay attention to and how to better relate the information to prior experience.

A Model of Conceptual Modeling as Knowledge Construction

We use the model of knowledge construction as a framework for reasoning about conceptual modeling (Gemino & Wand, 2003). The model viewer, in this framework, is constructing knowledge by actively organizing and integrating newly presented information with previous experiences. Three antecedents of the process are suggested: (1) content, (2) presentation method, and (3) model viewer characteristics. The content represents the domain information to be communicated. The presentation method is the way in which content is presented to the viewer. Viewer characteristics are attributes of the person viewing the model prior to viewing the content. These characteristics include knowledge and experience with the domain and with the modeling methods used to present information. This model is depicted in Figure 3.

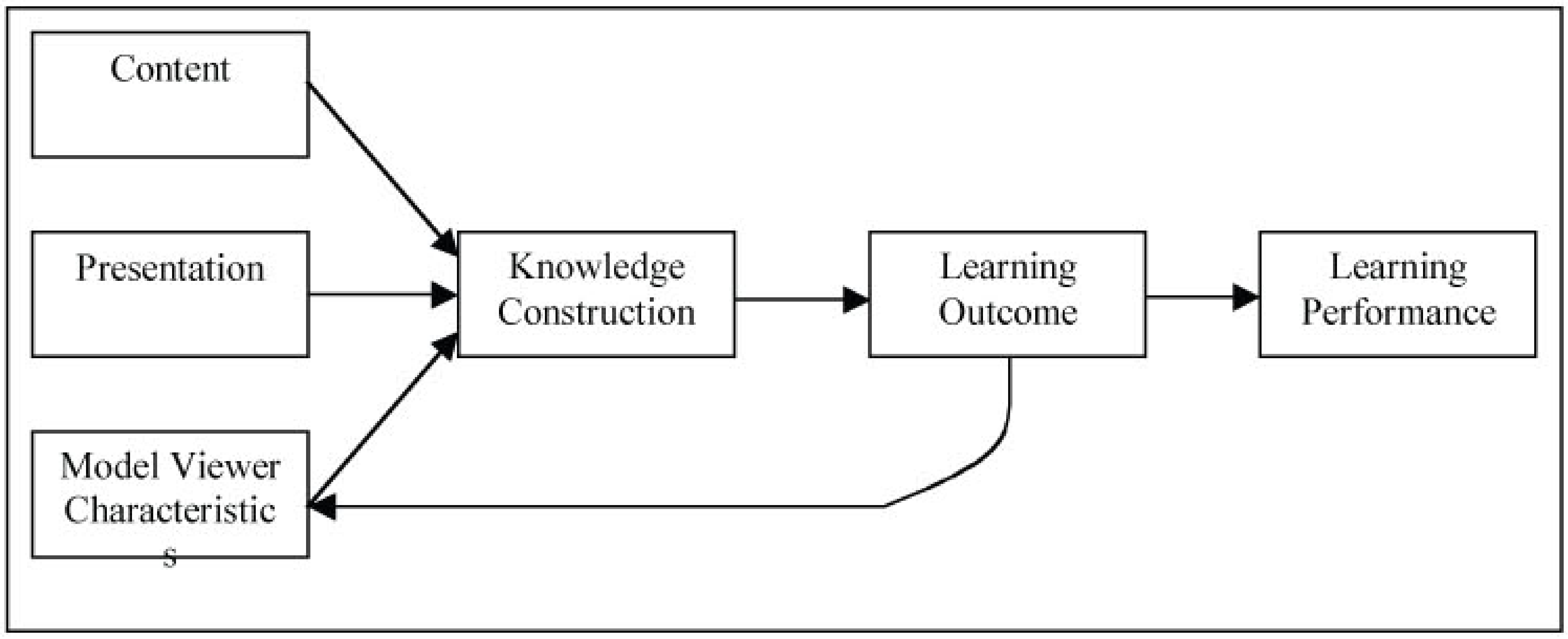
The construction process is where the sense making activity is hypothesized to occur. The results of knowledge construction are encoded into the long-term memory. This new knowledge is termed the learning outcome. The learning outcome modifies the model viewer's characteristics as shown in Figure 3. Learning outcomes can then be observed, only indirectly, through learning performance tasks.

The Cognitive Theory of Multimedia Learning

Conceptual modeling techniques often combine graphical symbols with words. Messages that combine graphics and words are defined by Mayer (2001) as "multimedia messages." He defines multimedia based on presentation modes (verbal and visual) of the person receiving this information rather than on the media used to present this information (video, written word, speaker, etc.). The Cognitive Theory of Multimedia Learning (CTML) provides a theoretical perspective for considering the level of understanding developed by a person viewing explanative material, such as an analysis diagram in conceptual model validation. The theory is based on work by Baddeley (1992) and Paivio (1986) and has been developed using over a decade of empirical work (Mayer, 1989, 2001).

The theory is focused on the interaction between a person and the information presented to him or her. The CTML suggests there are two pathways in cognition, verbal and visual. While independent, these channels communicate in working memory. When a person views presented material, relevant information from the verbal and visual channels is selected into working memory. This information is organized to create separate visual and verbal models in working memory. These two visual and verbal models then interact

Figure 3. Elements of learning process in conceptual modeling (Gemino & Wand, 2003, p. 82)

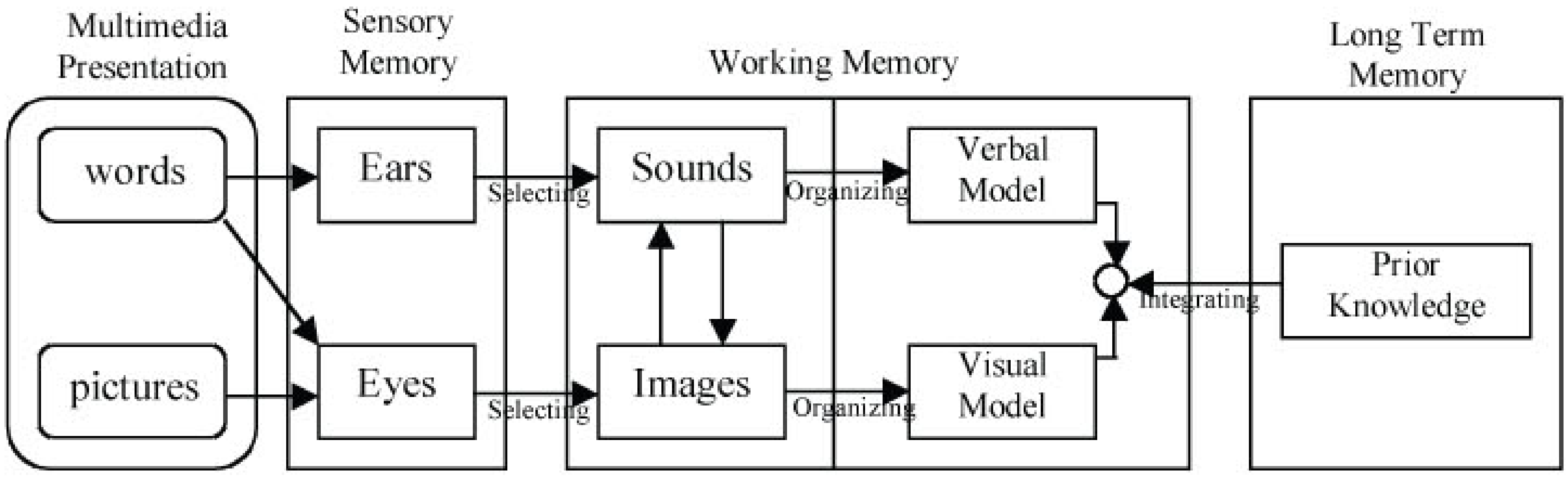


and are subsequently integrated with prior knowledge in long-term memory to develop new knowledge. An overview is provided in Figure 4.

In the CTML, an understanding of verbal and visual information is developed through three stages of memory. In the first stage, *sensory memory*, information is selected into one of the two dual coding pathways. This first stage of memory requires viewers to pay attention to appropriate pieces of information and filter other irrelevant or less relevant information out. This stage is a process of selecting information. Experts and novices are likely to have different abilities to select appropriate information so the amount of cognitive activity required to select appropriate information will vary between individuals. The selected information is then incorporated into visual and verbal models in the second stage of memory called *working memory*. Working memory is used to organize the selected information. There is the opportunity for some interaction between the visual and verbal models in working memory. These linkages help to increase the sophistication of the cognitive model and can improve the integration process. In the final stage of memory, the verbal and visual models from working memory are integrated with *long-term memory*. This integration results in the level of understanding developed by the person viewing the content. This three-stage memory process describes what we will refer to here as the knowledge construction process.

There are two additional important considerations in the knowledge construction process. The first is that the three memory stages described above occur simultaneously and continuously during cognitive processing. Information is continually being selected, organized and integrated and all three of these memory stages must be supported at the same time. The second consideration is that human beings have limited cognitive resources. Only a limited amount of processing power is available to support each of the three stages. If, for example, the content is presented in a confusing way for the viewer, it

Figure 4. The cognitive theory of multimedia learning (adapted from Mayer, 2001, p. 59)



is likely that more cognitive activity must be shifted to the memory selection stage, thereby slowing down or limiting the ability to of the viewer to organize and integrate the selected information. The attention for new content competes against the other stages of memory required for organizing and integrating previously received information. To develop understanding, it is therefore important to provide information that is not only easily recognized but also easily assimilated.

Learning Outcomes and Performance

The CTML has enabled the development of principles relating to the effective design of multimedia messages. The theory suggests the most effective communication occurs when verbal and visual pathways are utilized simultaneously. Mayer (2001) suggests three outcomes when presenting explanative material to people: 1) no learning, 2) rote learning and 3) meaningful learning. These outcomes are based on measures of retention and problem solving. Retention is the comprehension over time of material being presented. Problem solving is the ability to use knowledge gained to answer related problems not directly answerable from presented material. For example, if presented with an explanation of how a car's brake system works, a retention question might be "*List the components of a braking system*," but a problem solving question would be "*What could be done to make brakes more reliable?*" These problem solving task have been used by Bodart et al. (2001), Burton-Jones and Meso (2006), Gemino (1999; 2004), and Gemino and Wand (2005).

No learning occurs were retention and problem solving are low. Rote learning occurs where retention is high; however, problem solving measures are low. This indicates that although the material has been selected and received, the material has not been well integrated with prior knowledge. Meaningful learning occurs where retention and problem solving are high. This is summarized in Table 1.

This section has described a theory, the CTML, which suggests that combining graphic and textual information can lead to increased learning outcomes. We have argued above that increased learning outcomes are equivalent to higher levels of meaningful learning as measured by the combination of retention/comprehension and problem solving instruments. The CTML has provided a path for a better learning environment, but has yet to be used and tested in the Systems Analysis literature. In the section that follows, we will outline an experiment and hypotheses that suggest that the use of a summary graphic, such as a use case diagram, along with a set of primarily text based use cases will provide model viewers with a significantly higher level of meaningful learning. The level of meaningful learning will be measured by a combination of retention/comprehension and problem solving questions.

Table 1. Describing types of learning outcomes

Type of Learning	Scores on Retention/ Comprehension Tasks	Scores on Problem Solving Tasks
No Learning	Low	Low
Rote Learning	High	Low
Meaningful Learning	High	High

Meaningful learning will occur when comprehension levels are equal to or greater than the group provided with no diagram and where problem solving measures are significantly higher in the group provided with the diagram.

EXPERIMENTAL DESIGN AND HYPOTHESES

In the application of CTML to this study, three dependent variables (comprehension, retention and problem solving) were measured. Other variables measured included prior knowledge of the domain, knowledge of the modeling method, and participant demographics. A treatment group was compared to a control group using a single case.

The control group was provided with a set of five use cases describing a simple bus reservation system. The treatment group was provided the same set of use cases along with a one-page use case diagram. The only difference between the two groups was that the treatment group had the use case diagram. The single page use case diagram shows the interaction between use cases and actors in the system as well as any interactions among use cases in the system.

The following statement provides the underlying logic for conducting this experiment: If a participant is presented with a) a set of use cases and b) a set of use cases and a use case diagram relating these use cases, then the participant will develop a significantly higher level of understanding of the domain being presented with b) than with a). Since use case diagrams in this context are supplements to use cases, the only combination of treatments that makes sense is the use case alone or the use case supplemented by a use case diagram.

Hypotheses

Mayer's (2001) multimedia principle suggests individuals learn better from words and pictures than words alone. When words and graphics are presented together, learners have the opportunity to develop verbal and visual models and build connections between them. When presented with only words, individuals are less likely to develop visual models. As a result, the connections between the verbal and visual models may be lost. The use case diagram may provide a foundation for the selection and integration of information across the use cases. Since cognitive resources are limited, the use case diagram may serve as an effective framework for organizing information and hence allow additional cognitive resources for developing a more sophisticated model of the domain being represented. This suggests a potential for higher levels of understanding derived from use cases supported with a use case diagram as opposed to use cases alone.

The multimedia principle therefore enables us to suggest the following hypotheses. Participants viewing a set use cases with an associated use case diagram will:

H1: *achieve scores on comprehension tasks that are equal to or greater than the group of participants viewing use cases alone.*

H2: *achieve scores on retention tasks that are equal to or greater than the group of participants viewing use cases alone.*

H3: *achieve scores on problem solving tasks that are greater than the group of participants viewing use cases alone.*

METHOD

An empirical procedure was developed to test the hypotheses. The procedure of collecting comprehension, retention, and problem solving measures is based on work by Mayer (1989, 2001). This procedure has been used in the area of system analysis by others including Gemino (1999), Bodart et al. (2001), Burton-Jones and Meso (2006), Gemino (1999, 2004), and Gemino and Wand (2005).

Participants:

Forty-nine upper level business students took part in the study. All students had taken a system analysis course and had basic familiarity with use case models. Females accounted for 20 of the 49 participants (41 percent) of participants. Participation was voluntary. An incentive of $15 was provided for the top four performers. The average time to complete the study was 45 minutes. All participants were at an introductory level in business process design, and had no particular experience with object oriented analysis or the UML. An instrument was provided before the experimental tasks to collect experience with system analysis and the business domain used in the analysis as well as other demographic variables.

Materials:

One case including five use cases and one use case diagram was used in the experiment. The materials are provided in the Appendix. The use cases and use case diagram were created using an approach described in Dennis and Wixom (2000). The text description was provided by the Voyager Bus company case in Bodart et al (2001).

An important note must be made in regard to informational and computational equivalence of the control and treatment groups. The use cases provided to both groups were exactly the same. The level of detail in the uses cases was very general and provided little information in regards to how the system was designed (other than you log onto the system). We argue that with regard to informational equivalence, the control and treatments can be viewed as providing similar information content. The use case diagram provides no additional information that could not be derived from the use cases. The information about the actors involved with the system is available in the use cases. The actors interacting with the each use case are noted. The interaction type (external or temporal) was also noted.

While we argue there are no informational differences between the cases, we recognize that there is likely computational inequivalence. The use case diagram provides no new information; however, the use case diagram does organize the available information differently than the use cases. This organization may help viewers improve their understanding because it provides an understandable graphic. This would represent a computational advantage. The question of whether the computational advantage is significant is what is addressed by the experiment.

Procedure

Participants were randomly assigned into two treatment groups (with or without diagram). An envelope was given to each participant containing a pre-test, five use cases (plus diagrams if necessary), experimental tasks (comprehension, retention and problem solving) and a posttest. Participants worked independently and first completed the pre-test followed by the three experimental tasks and finally the post-test.

The three experimental tasks were completed in a specific order. The first task was a 12-question multiple choice comprehension task (True, False, Uncertain). After the comprehension task, participants were instructed to put away the use cases and diagram (if provided). Participants were then given six minutes to complete a retention task, which asked participants to write down everything they knew about the processes in the use cases. This task was followed by four problem-solving questions used by Bodart et al. (2001). Participants were given two minutes to write as many answers as possible to each problem solving question.

Measures

Learning performance was measured using three variables: comprehension, retention, and problem solving. Comprehension was the number of correct answers out of a possible of 12 (true/false/uncertain) questions. Retention and problem solving scores were coded by two individuals. The retention score was created by giving one mark for each complete and correct idea statement expressed by the participant. There was a maximum of 20 idea statements identified in the use cases. The problem solving score was created by giving participants one point for each acceptable response to the problem solving questions. The Pearson correlation between coders for retention was 0.88 and for problem solving questions 0.90. Differences between independent ratings were then discussed, and a final score for retention and problem solving was established.

RESULTS

Preliminary Tests

Since the sample size was relatively small, it is important to establish the homogeneity of variances before ANOVA analysis. Table 2 below provides the Levene statistics for each of the dependent measures. As shown in the table, the hypothesis of equal variances is not rejected across any of the variables at the 0.05 level.

Domain and modeling experience were collected in the pre-test survey and used as covariates in an ANCOVA analyses. Both domain and modeling method experience were found to have insignificant influences on the dependent variables. This result may be due to the uniformly low levels of experience held by participants. While it seems likely prior domain experience and modeling method experience are related to the dependent measures (Khatri, Vessey, Ramesh, Clay, & Park, 2006), the factors, as measured in this study, had no significant effect in this study and were excluded in further analysis.

Results

The means and standard deviations of the dependent measures (comprehension, retention and problem solving) across the two treatment groups are provided in Table 3 below. The results show little difference across treatment groups for comprehension measures. Note that participants had full access to

Table 2. Test for homogeneity of variances for dependent measures

Measure	Levene Statistic	df1	df2	Sig.
Comprehension	0.349	1	47	0.558
Retention	0.375	1	47	0.543
Problem solving	0.236	1	47	0.630

Table 3. Means and std. dev. across treatments for dependent measures

Dependent Measure	Case: Voyager Bus				
	Treatment Groups				
	Without Diagram n=25 Means (SD)	With Diagram n=24 Means (SD)	Difference between means (With-Without)	Effect Size as % of Without Diagram	Sig.
Comprehension	7.627 (.321)	8.139 (.328)	.512	6.7%	0.271
Retention	7.877 (.541)	9.670 (.552)	1.793	22%	0.025*
Problem solving	12.174 (.824)	14.568 (.841)	2.394	20%	0.045*

* *significant at the 0.05 level*

use cases during the comprehension test. Since the information was available in either treatment, the diagram had little effect in basic comprehension. This result suggests that no systematic information advantage was noted between the two experimental groups.

Retention measures showed differences in the anticipated direction. The size of the effects was approximately 22 percent. This is measured by dividing the difference between the "with" and "without" diagram scores and then dividing the result by the score for the without diagram group.

More importantly, the problem solving measures showed significant differences in the anticipated direction. The size of the effects was approximately 20 percent which was again created by dividing the difference between the "with" and "without" diagram scores and then dividing the result by the score for the without diagram group.

An ANOVA was applied to test the significance of these differences and to test hypotheses H1a, b and c. Results, provided in the final column of Table 2, suggest no significant different in comprehension. This confirms hypothesis H1. The results also imply that there seemed to be no systematic informational bias towards the group provided with the diagram.

In addition, significant differences were observed for both retention and problem solving measures at $\alpha = 0.05$ level. These results support both H2 and H3. These results suggest that although the informational content across treatments was essentially the same, the organization provided by the use case diagram enabled participants with access to the diagram to build a more sophisticated mental model and establish more meaningful learning. This was revealed in higher scores in the problem solving task. Note that although the sample size is relatively small, the effect size is relatively large.

These results suggest that diagrams, even simple diagrams such as the use case diagram provided in this experiment, can have measurable effects on viewer understanding. While use case modeling may be a step forward in requirements determination, it should be noted that text-only use cases may perform significantly better when a corresponding use case diagram is also provided. This occurs because the use case diagram provides a pictorial view of the relations between use cases. The pictorial view provides clues for selecting and integrating important information from text descriptions thereby preserving cognitive processing for increased integration of the material presented in later memory stages.

DISCUSSION

This article used the CTML to hypothesize that the inclusion of a use case diagram can make use case modeling significantly more effective. Use case modeling is a widely used technique to communicate systems models. If the models used to represent systems can be improved to lead to more effective communication and learning about the system being represented, perhaps the failure rate of systems projects could be favorably affected. Charette (2005) states that one of the "most common" reasons for the failure of IT projects is: "Poor communication among customers, developers, and users."

For practitioners who choose to communicate using use cases alone, the results suggest a relatively simple approach of combining text and graphics will improve meaningful learning about the system. These results help to explain the results in Dobing and Parsons (2008) which show the relatively high use of use case diagrams in interaction with clients, second only to the use case narrative. While the use case diagram does not seem to add new information, and hence may not be worth the cost of development, the graphic does seem to provide a cognitive framework that helps users better understand sets of uses cases. This is likely the reason why practitioners continue to use the use case diagrams in interactions with clients. For those practitioners and researchers who suggest combining use cases with UML diagrams, this study provides tangible proof that the combination of text and diagrams can make for significantly improved levels of understanding.

Pictures representing domain constructs are a natural form of communication. The results of this experiment provide evidence that participants developed a higher level of domain understanding when viewing UML use cases with the support of a use case diagram. The hypothesis that a use case diagram has a significant positive effect on level of understanding developed by a person viewing use cases was therefore supported. This result also supports our assertion that use cases augmented with a use case diagram provides a more effective communication of system information than use cases alone. The implications for researchers and practitioners are to include use case diagrams with use cases when possible. Practitioners in particular should note the effectiveness of graphical models when used in conjunction with use case models.

This article provides the first evidence that the authors are aware of showing use case diagrams can effectively support use case modeling. In addition, the article provides support for the Cognitive Theory of Multimedia Learning and the multimedia principle that the theory suggests in the realm of systems analysis and design. This use of theory answers a call for more theory-based approaches to the investigation of analysis methods (Wand & Weber, 2002). The results reaffirm the importance of diagrams and visual information in developing understanding. The results also suggest that practitioners utilizing uses cases without some form of visual overview may not be getting the full effect of use case modeling.

When considering the results, it should also be noted that the use case diagram is not the only graphic model available to researchers and practitioners. While this study has shown a significant effect from including a use case diagram, further improvements may be possible by adding or substituting other UML diagrams (e.g. class model, sequence diagram, activity diagram). This experiment has provided evidence that relevant graphic information improved understanding above that provided by text based uses cases alone. The results of this experiment do not show that the use case diagram is the "best" diagram to be used with uses cases. For example, a class diagram may offer additional insight for model viewers. Future research can be directed more closely on what diagram elements are most effective in supporting use cases. In addition, more empirical evidence is required to understand the effectiveness of use case modeling. While the text based approach has some excellent features and has appealed to practitioners, it is clear that diagrams are an important component for communication. More needs to be understood about this relationship if we are to make use case modeling an even more effective communication tool for stakeholders and developers.

CONCLUSION

This paper has shown that the addition of a graphic representation to the text-based use case can significantly enhance understanding of a system among novice modelers. More research is needed to clarify the effectiveness of use cases and supporting diagrams. The Cognitive Theory of Multimedia Learning has been shown as a potential theory to support and test systems analysis techniques. The results in this paper support the use of a combination of text and visual models to communicate complex systems information.

ACKNOWLEDGMENT

The authors wish to thank Liliana Petrescu for her work in collecting and coding experimental data. This research was supported by the National Science and Engineering Research Council of Canada. We would like to thank the anonymous reviewers for providing comments that improved our paper.

REFERENCES

Agarwal, R., De, P., & Sinha, A.P. (2003). Object oriented modeling with UML: A study of developer perceptions. *Communications of the ACM*, *46*(9), 248-256.

Angioni, M., Carboni, D., Pinna, S., Sanna, R., Serra, N. & Soro, A. (2006). Integrating XP project management in development environments. *Journal of Systems Architecture*, *52*(11), 619-626.

Baddeley, A. D. (1992). Working memory. *Science, 255*, 556-559.

Batra, D. (2008). Unified Modeling Language (UML) topics: The past, the problems, and the prospects. Guest editorial preface. *Journal of Database Management, 19*(1), i-vii.

Bodart, F., Patel, A., Sim, M., & Weber, R. (2001). Should optional properties be used in conceptual modelling? A theory and three empirical tests. *Information Systems Research, 12*(4), 384-405.

Burton-Jones, A. & Meso, P. (2006). Conceptualizing systems for understanding: An empirical test of decomposition principles in object-oriented analysis. *Information Systems Research, 17*(1), 38-60.

Charette, R. (2005). Why software fails. *IEEE Spectrum Online*. Retrieved September 11, 2007 from http://www.spectrum.ieee.org/sep05/1685

Dennis, A., & Wixom, B. (2000). *System analysis and design: An applied approach*. New York: John Wiley and Sons.

Dobing, B. & 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. & Parsons, J. (2008). Dimensions of UML diagram use: A survey of practitioners. *Journal of Database Management, 19*(1), 1-18.

Dori, D. (2002). Why significant UML change is unlikely. *Communications of the ACM*, *45*(11), 82-85.

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

Evermann, J. (2005). Thinking ontologically - Conceptual versus design models in UML. In M. Rosemann & P. Green (Eds.), *Ontologies and business analysis.* Idea Group Publishing, 82-104.

Evermann, J. & Wand, Y. (2005). Ontology-based object-oriented business modelling: Fundamental concepts. *Requirements Engineering Journal*, *10*(2), 146-160.

Fedorowicz, J. & Villeneuve, A. (1999). Surveying object technology usage and benefits: A test of conventional wisdom. *Information & Management*, *35*(6) 331-345.

Gemino, A. (1999). *Empirical comparison of system analysis techniques*, Ph.D. Thesis, University of British Columbia, Vancouver, British Columbia.

Gemino, A. (2004). Empirical comparisons of animation and narration in requirements validation. *Requirements Engineering Journal, 9*(3), 153-168.

Gemino, A. & Wand, Y. (2003). Evaluating modeling techniques based on models of learning. *Communications of the ACM, 46*(10), 79-84.

Gemino, A. & Wand, Y. (2004). Dimensions in experimental evaluation of conceptual modeling techniques. *Requirements Engineering Journal, 9*(4), 248-260.

Gemino, A. & Wand, Y. (2005). Simplicity versus clarity: An empirical comparison of mandatory and optional properties in conceptual modeling. *Data and Knowledge Engineering*, *55*, 301-326.

Halpin, T. & Bloesch, A. (1999). Data modeling in UML and ORM: A comparison. *Journal of Database Management, 10*(4), 4-13.

Jacobson, I., Ericsson, M., & Jacobson, A. (1994). *The object advantage: Business process reengineering with object technology*. Reading, MA: Addison-Wesley.

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The unified software development process.* Reading, MA: Addison Wesley.

Johnson, R. (2002). Object-oriented system development: A review of empirical research. *Communications of the Association for Information Systems*, *8*, 65-81.

Khatri, V., Vessey, I., Ramesh, V., Clay, P., & Park, S.-J. (2006). Understanding conceptual schemas: Exploring the role of application and IS domain knowledge. *Information Systems Research*, *17*(1), 81-99.

Kobryn, C. (1999). A standardization odyssey. *Communications of the ACM*, *42*(10), 29-38.

Mayer, R. (1989). Models for understanding. *Review of Educational Research, 59*(1), 43-64.

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

Mellor, S. (2002). Make models be assets. *Communications of the ACM*, *45*(11), 76-78.

Meso, P. & Jain, R. (2006). Agile software development: Adaptive system principles and best practices. *Information Systems Management*, *23*(5), 19-29.

Paivio, A. (1986). *Mental representations: A dual coding approach.* Oxford, England: Oxford University Press.

Siau, K. & Cao, Q. (2001). Unified Modeling Language (UML) - A complexity analysis. *Journal of Database Management, 12*(1), 26-34.

Siau, K. & Loo, P. (2006). Identifying difficulties in learning UML. *Information Systems Management*, *23*(3), 43-51.

Siau, K., Erickson, J., & Lee, L. (2005). Theoretical vs. practical complexity: The case of UML. *Journal of Database Management, 16*(3), 40-57.

Shoval, P. & Kabeli, J. (2005). Essentials of functional and object-oriented methodology. *Encyclopedia of Information Science and Technology, II*, 1108-1115.

Standish Group. (1994). *Chaos report*. Retrieved September 11, 2007 from http://standishgroup.com/sample_research/chaos_1994_1.php

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. & Weber, R. (2002). Information systems and conceptual modeling: A research agenda. *Information Systems Research*, 203-223.

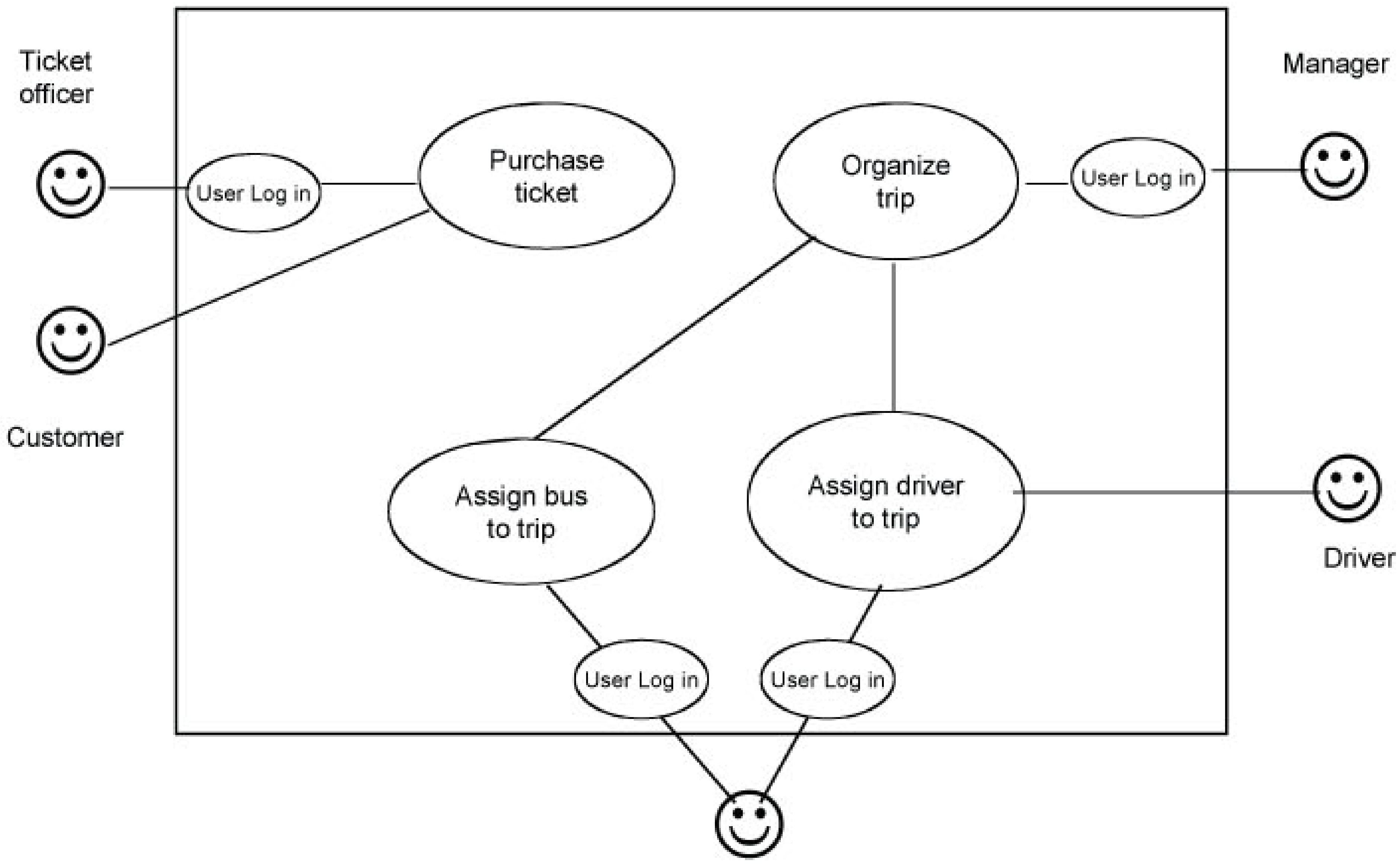
APPENDIX

Five use cases (see below):

1. User Login
2. Purchase Ticket
3. Organizing a Trip
4. Assign Driver to Trip
5. Assign Bus to Trip

Use Case Diagram

Organizing a Bus Trip



User Case Login

Use case Name: User Login			ID Number: 1
Use case Description: This describes how the users accesses the Voyager system			
Trigger: Employee needs to access the system Type: [✓ External] / Temporal			
Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
User Login ID Password	Employee Employee	Employee accesses the system	System functionality
Major Steps Performed			Information for Steps
1. User needs to access the Voyager system and enters login ID, then password in order to perform specific tasks in the system.			User ID User password

User Case Ticket

Use case Name: Purchase ticket			ID Number: 2
Use case Description: This describes the ways that can be used to book travel with Voyager, namely through reservation or direct purchase.			
Trigger: Person decides to go on a trip with Voyager Type: [✓ External] / Temporal			
Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
Destination of trip Desired date of the trip Traveler's name Traveler's address Traveler's phone number	Passenger Passenger Passenger Passenger Passenger	Passenger reservation date Passenger boarding date	Passenger Bus trip attendance Passenger Bus trip attendance
Major Steps Performed			Information for Steps
1. Passenger requests a reservation on a trip, by providing name, address and telephone number and payment method.			Passenger
• Or, passenger directly purchases a ticket at the boarding gate for an unreserved seat, also providing name, address and telephone number.			Bus trip attendance
2. Passengers with a reservation are assigned a reservation date and confirmation number. Passengers without a reservation are assigned a boarding date and a ticket.			Passenger Bus trip attendance

User Case Organizing a Trip

<table>
<tr><td colspan="3">Use case Name: Organizing a trip</td><td>ID Number: 3</td></tr>
<tr><td colspan="4">Use case Description: This describes how a trip is created using route segments.</td></tr>
<tr><td colspan="4">Trigger: Manager receives go ahead to enter a trip into system
Type: [✓ External] / Temporal</td></tr>
<tr><td colspan="2">Major Inputs:</td><td colspan="2">Major Outputs:</td></tr>
<tr><td>Description</td><td>Source</td><td>Description</td><td>Destination</td></tr>
<tr><td>Route segment’s start town</td><td>Route segment</td><td rowspan="6">Trip with associated number, event name if applicable, start town and finish town</td><td rowspan="6">Trip</td></tr>
<tr><td>Route segment’s finish town</td><td>Route segment</td></tr>
<tr><td>Trip start town</td><td>Town list</td></tr>
<tr><td>Trip finish town</td><td>Town list</td></tr>
<tr><td>Trip start time</td><td>Trip schedule</td></tr>
<tr><td>Trip finish time</td><td>Trip schedule</td></tr>
<tr><td colspan="3">Major Steps Performed
1. To enter a new trip, a manager enters a unique trip number, a start and finish town, and a start and finish date.
2. Each trip is made up of route segments. A manager assigns route segments to the trip number. Route segments are defined by a segment number with a start and finish town.
3. Manager assigns maximum and minimum number of passengers for the trip. Trips do not run unless they have a minimum number of passengers.</td><td>Information for Steps
Route segment
Daily route segment
Bus trip</td></tr>
</table>

User Case Assign Driver to Trip

Use case Name: Assign driver to trip			ID Number: 4
Use case Description: This describes the procedure of assigning available bus drivers to trips			
Trigger: Manager enters new trip Type: External / [✓ Temporal]			
Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
Driver's name Driver's address Driver's employee number Driver's absence status	Drivers records Drivers records Drivers records Drivers records	Driver(s) choice for bus trip is made	Driver's schedule
Major Steps Performed			Information for Steps
1. When the daily bus trip has been defined, scheduler receives trip details.			Drivers records
2. Drivers' profiles (including availability) are viewed in order to assign one or more drivers to the trip.			Drivers records
3. If a driver has a record of frequent absences, then the scheduler must verify availability with driver before scheduling.			Bus trips
4. At the end of each week, a report of the drivers' schedules is created for the coming week.			Drivers records
5. The scheduler posts this schedule for the coming week.			

User Case Assign Bus to Trip

Use case Name: Assign bus to trip	ID Number: 5
Use case Description: This describes the process of assigning a bus to a daily trip, after checking the maintenance status of the bus	
Trigger: Driver is assigned by scheduler Type: External / [✓ Temporal]	

Major Inputs:		Major Outputs:	
Description	Source	Description	Destination
Make of bus	Bus records	Bus choice is made	Bus trip
Model of bus	Bus records		
Registration number of bus	Bus records		
Date of last maintenance	Bus records		
Average daily kilometers	Bus records		

Major Steps Performed	Information for Steps
1. When the bus trip has been defined and a driver has been selected, buses' records are reviewed in order to assign a bus to the daily trip.	Bus records
2. Maintenance status for each bus is verified. If maintenance status is up-to-date and in good standing, the bus is considered for the trip.	Bus records
3. Once maintenance records have been approved, the bus with the lowest kilometers that meets the maximum number of passengers required for the trip is selected and assigned to the trip.	Bus records
4. Once a vehicle is assigned to a particular trip number and route segment, it cannot be assigned to another trip number.	

Andrew Gemino is an associate professor in the management information systems area in the Faculty of Business Administration at Simon Fraser University (SFU). His degrees include an MA in economics and an MBA from SFU and a PhD in MIS from the University of British Columbia. His research interests include information technology project management and the effective communication of information system requirements. His research is currently funded by the Natural Sciences Research Council (NSERC) and the Social Sciences and Humanities Research Council (SSHRC) of Canada. Gemino is vice president of the AIS Special Interest Group in System Analysis and Design (SIGSAND) and a member of the Surgeon Information System Working Group for the Provincial Surgical Oncology Council affiliated with the British Columbia Cancer Agency.

Drew Parker is an associate professor in the management information systems area in the Faculty of Business Administration at Simon Fraser University. He holds a BComm and a MBA degree from the University of Calgary, and a PhD from the Ivey School of Business at the University of Western Ontario. He also holds the Information Systems Professional (I.S.P.) designation from the Canadian Information Processing Society. His current research interests originated to a large extent from work developing and delivering system analysis courses online internationally, and working on using telecommunications and the Internet to communicate and collaborate effectively.