COMPUTATIONAL ANALYSIS OF KNOWLEDGE SHARING IN COLLABORATIVE DISTANCE LEARNING

by

Amy L. Soller

B.S., Boston University, 1996

M.S., University of Pittsburgh, 2000

Submitted to the Graduate Faculty of

the Intelligent Systems Program in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2002

UNIVERSITY OF PITTSBURGH

INTELLIGENT SYSTEMS PROGRAM

This dissertation was presented

by

Amy L. Soller

It was defended on

August 30, 2002

And approved by

Dr. Janyce Wiebe
Professor of Computer Science and Intelligent Systems, University of Pittsburgh

Dr. Diane Litman
Professor of Computer Science and Intelligent Systems, University of Pittsburgh

Dr. Daniel Suthers
Professor of Computer Science, University of Hawaii

Dr. Sandra Katz
Research Associate, Learning Research and Development Center, University of Pittsburgh

Dissertation Director: Dr. Alan M. Lesgold
Dean, School of Education; Professor of Intelligent Systems, University of Pittsburgh

In loving memory of Denise Danyelle Moffatt-Pascale

Dissertation Advisor: Dr. Alan M. Lesgold ____________

COMPUTATIONAL ANALYSIS OF KNOWLEDGE SHARING IN COLLABORATIVE DISTANCE LEARNING

Amy L. Soller, Ph.D.

University of Pittsburgh, 2002

The rapid advance of distance learning and networking technology has enabled universities and corporations to reach out and educate students across time and space barriers. This technology supports structured, on-line learning activities, and provides facilities for assessment and collaboration. Structured collaboration, in the classroom, has proven itself a successful and uniquely powerful learning method. Online collaborative learners, however, do not enjoy the same benefits as face-to-face learners because the technology provides no guidance or direction during online discussion sessions. Integrating intelligent facilitation agents into collaborative distance learning environments may help bring the benefits of the supportive classroom closer to distance learners.

In this dissertation, I describe a new approach to analyzing and supporting online peer interaction. The approach applies Hidden Markov Models, and Multidimensional Scaling with a threshold-based clustering method, to analyze and assess sequences of coded on-line student interaction. These analysis techniques were used to train a system to dynamically recognize when and why students may be experiencing breakdowns while sharing knowledge and learning from each other. I focus on knowledge sharing interaction because students bring a great deal of specialized knowledge and experiences to the group, and how they share and assimilate this knowledge shapes the collaboration and learning processes. The results of this research could be used to dynamically inform and assist an intelligent instructional agent in facilitating knowledge sharing interaction, and helping to improve the quality of online learning interaction.

Acknowledgments

This dissertation is dedicated to the memory of my dearest friend, Denise Danyelle Moffatt-Pascale. Through Denise's battle with breast cancer, I have learned to embrace life and the opportunities it affords us, and I have learned to appreciate the support and love of my family and friends. Not every woman is given the opportunity to attain a PhD, and I deeply thank those people who helped to make this possible for me.

Over the past four years, I have had the privilege to work closely with one of the most extraordinary men I know. Alan Lesgold has been not just my advisor, but also my colleague and friend. He has helped me see the challenges and rewards of interdisciplinary research, and the meaning of true mutual respect.

Dan Suthers was also involved in both my masters and dissertation research, and played a major role in helping me evolve and focus this work. Through conversations we have had around the world, from Colorado to France to Japan, he has helped me shape many of these ideas.

Working with my committee members, Diane Litman, Dan Suthers, Jan Wiebe, and Sandy Katz, has been especially rewarding. They posed challenges to help me improve my work, and encouraged me to view my research from different perspectives. Special thanks also to my friend and colleague, Patrick Jermann, who has shown me that great ideas really do come on cocktail napkins!

My family has provided stability, and unfaltering love, support, and encouragement throughout my dissertation studies. To my parents, Carol and Herb, and to Jeff and Debbie, and Jon and Robin, thanks so much for always being there. I love you.

Thanks to all my friends for making sure that I continued to experience life while writing my dissertation. Thanks especially to Lisa Scott, Carol "fun!" Uminski, Alida Skogsholm, John "you'dn't've" Connelly, Reid Loper, Ken Koedinger, Steffi Bruninghaus, Saeeda Hafiz, Todd Kuhl, Jeff Pascale, and the Moffatt family.

My friends and colleagues from MITRE, Brad Goodman and Frank Linton, served as invaluable resources, and continually provided motivation during my dissertation studies. They helped to run the experiments reported in this dissertation, and shared their data and research ideas. The roots of many of my dissertation ideas grew out of my work as a member of Brad's MSR team, before beginning graduate school.

My colleagues around the world have both motivated this work, and contributed to raising global interest in analyzing collaborative learning interaction. Special thanks to Kwang-Su Cho, Martin Muehlenbrock, Angeles Constantino, Pierre Dillenbourg, Ulrich Hoppe, Helen Pain, and Martha Pollack.

This research was funded by an Andrew Mellon Predoctoral Fellowship, and by the NetLearn project at LRDC. Thanks to all the folks at LRDC who made my experience at the University of Pittsburgh so memorable.

Table of Contents

List of Tables

List of Figures

1 Introduction

The rapid advance of networking technology has enabled universities and corporate training programs to reach out and educate students who, because of schedule or location constraints, would not otherwise be able to take advantage of many educational opportunities. Although distance learning programs have begun to revolutionize the face of education, many still struggle to provide a supportive environment in which students can interact with and learn from other students in their class. A popular response to this issue has been the deployment of Computer-Supported Collaborative Learning (CSCL) systems (Guzdial et al., 1997; Jermann & Dillenbourg, 1999; Scardamalia & Bereiter, 1994; Singley, Fairweather, & Swerling, 1999; Suthers, Weiner, Connelly, & Paolucci, 1995). CSCL systems offer software replicas of many of the classic classroom resources and activities. They may provide shared workspaces, on-line presentations, lecture notes, reference material, quizzes, student evaluation scores, and facilities for chat or on-line discussions. Successful distance learning programs around the globe have proven almost all of these tools successful. All but one – the support for on-line learning communication. Chat tools and bulletin boards (Blackboard, Inc., 1999; Bruckman & Bonte, 1997; O'Day, Bobrow, Bobrow, Shirley, Hughes, & Walters, 1998) enable students to participate in on-line discussions, but provide no guidance or direction to students during or after these dialogue sessions.

In the classroom, effective collaboration with peers has proven itself a successful and uniquely powerful learning method (Brown & Palincsar, 1989; Doise, Mugny, & Perret-Clermont, 1975). Learning and working with peers may benefit the overall team performance by increasing the quality of the team product, and it may also enhance individual performance. Students learning effectively in groups encourage each other to ask questions, explain and justify their opinions, articulate their reasoning, and elaborate and reflect upon their knowledge, thereby motivating and improving learning. These benefits, however, are only achieved by active and well-functioning learning teams. Placing students in a group and assigning them a task does not guarantee that the students will engage in effective collaborative learning behavior (King, Staffieri, & Adelgais, 1998; Salomon & Globerson, 1989). While some peer groups seem to interact naturally, others struggle to maintain a balance of participation, leadership, understanding, and encouragement.

Classroom instructors are often responsible for teaching students not only the cognitive skills necessary to learn the subject matter, but also the social skills they need to communicate well in a team. Just as students learning in the classroom need support from their instructor, students learning via CSCL technology need guidance and support on-line. The state of the art in CSCL technology today (Jermann, Soller, & Muehlenbrock, 2001) still falls behind in providing the sort of support that a teacher might provide in the classroom. Supporting group learning requires an understanding of the collaborative learning process, which is shaped by not only the individual's ability, learning style, and motivation, but also the group members' individual behaviors, and the dynamics of their interaction (described further in chapter 2). Although CSCL researchers have made great strides in analyzing and assessing group interaction online, they still struggle to develop computational methods for supporting the sort of rich student interaction that is found in the classroom. Providing a supportive environment means accounting for the spectrum of activities that groups engage in while learning. These activities include the exchange and negotiation of knowledge (Baker, 1994), possibly leading to conflict and cognitive change (described in section 2.1), the construction of new knowledge (section 2.2), and the development of social skills (section 2.3).

In chapter 3, we will see how a group's ability to share, understand, and construct new knowledge is an important predictor of the value of the group learning experience. The effectiveness of knowledge construction depends on the participants' evolving knowledge bases and the group's ability to share and assimilate the bits of knowledge necessary to construct new knowledge. As the shared bits of knowledge are assimilated into the group thinking process, group members evolve and develop a shared understanding. Understanding how knowledge is shared and assimilated in groups, and supporting this process, is critical to ensuring that students can establish the common ground necessary to co-construct knowledge. From an intuitive standpoint, the knowledge that group members bring to bear on the problem, and how this knowledge is shared, understood, and further developed (or not) ultimately shapes both the process and the product of the collaboration. This dissertation will therefore focus on analyzing the process of knowledge sharing during collaborative distance learning activities.

Chapter 4 discusses a number of ways other researchers have addressed the issue of analyzing and assessing online group interaction in the interest of providing a supportive CSCL environment. For example, researchers have studied aspects of group learning such as communication and trust (McManus & Aiken, 1995), creativity and conformity (Barros & Verdejo, 1999), and coordination and conflict (Muehlenbrock, 2001). To my knowledge, this dissertation is the first attempt to address the issue of computationally assessing online knowledge sharing.

Group members that do not effectively share the knowledge they bring to the group learning situation will have a difficult time establishing a shared understanding, and co-constructing knowledge with their teammates – both of which ultimately lead to poor learning outcomes (Jeong, 1998; Winquist, & Larson, 1998). A research effort to understand and support students' knowledge sharing behavior is a complex endeavor, involving analysis of student learning, understanding, conversation, and physical actions. But the results of such an effort can be applied to analyzing and supporting other complex aspects of collaborative learning, such as the joint construction of shared knowledge and cognitive conflict. Furthermore, this research may help to define guidelines about the limits on the kinds of support a collaborative learning system, in general, might offer.

This dissertation describes a novel approach to assessing the effectiveness of knowledge sharing dialog during collaborative learning activities. Knowledge sharing dialogs are naturally occurring conversation subsets in which group members attempt to understand some new conceptual knowledge that one group member presents, explains, or illustrates to them. My approach involved applying a machine learning technique, Hidden Markov Modeling, to differentiate between instances of effective knowledge sharing interaction, and instances in which students are having trouble sharing or understanding new knowledge (termed knowledge sharing "breakdowns").

In chapter 5, I propose a design for a CSCL system that could dynamically assess online knowledge sharing dialogs, and mediate situations in which students are having trouble understanding the new knowledge their teammates share. As part of this dissertation work, I have developed two key reusable components of the proposed system: the networked-based CSCL interface, and the knowledge sharing analyzer. The knowledge sharing analyzer was trained to classify sequences of knowledge sharing interaction as effective or ineffective, and then explain why the students did or did not succeed. The training sequences were obtained through a series of experiments, outlined in chapter 6, designed specifically to collect instances of knowledge sharing during collaborative learning. These instances were automatically coded to reflect both task (e.g. student A created object 6) and conversational events (e.g. student A requested elaboration).

Chapter 7 explains how the coded knowledge sharing sequences were used to train two Hidden Markov Models: one representing effective knowledge sharing interaction, and another describing breakdowns in knowledge sharing interaction. The models, when tasked to determine the effectiveness of new sequences of knowledge sharing interaction, correctly classified 74% of these sequences, a 24% improvement over the baseline (chance). Chapter 7 then takes a closer look at the differences between the effective and ineffective sequences in order to understand why students may be having trouble.

An approach that combines Hidden Markov Models, and Multidimensional Scaling with a threshold-based clustering method, was used to find groups of generalized models that represent the various ways students may effectively share knowledge, or the various ways that students may have trouble sharing new knowledge with each other. For example, effective knowledge sharing discussions were marked by questioning, explanation, agreement, and motivation, whereas discussions in which the students experienced breakdowns in knowledge sharing were marked by poor explanations, instructions for action, doubt, and acknowledgement. The results of this analysis could serve to inform and advise an online instructor or computer-based coach in selecting an appropriate facilitation strategy.

Chapter 8 further discusses the data analysis performed for this dissertation, and the implications of the analysis for the proposed CSCL system. Chapter 9 describes directions for future research along these lines.

The overall goal of this dissertation research is to illustrate the computational analysis of the knowledge sharing process, and show how we might identify situations in which facilitation might increase the effectiveness of the group interaction. Studying the interaction that provokes and follows knowledge sharing events may help us assess the ability of the group to assimilate new information that group members naturally bring to bear on the problem.

2 The Collaborative Learning "Effect"

In the United States educational system, students are traditionally encouraged to work independently, and grades are assigned based on individual performance. Yet, in our work force, most jobs require the ability to work in a team. Working with others often increases task efficiency and accuracy (Thagard, 1997), while giving each team member a valued role to play based on his or her individual area of expertise. Although it is not always the case, as we will see in the next chapter, groups sometimes outperform the best individual in the group. For example, Schwartz (1999) asked individuals and pairs of 7th graders to construct visualizations given descriptions of fictitious fish and their habitat requirements. Only 6% of the individuals who worked alone constructed abstract visualizations (e.g. charts and graphs instead of pictures of fish), whereas 67% of the pairs developed abstract visualizations. Working in pairs encouraged the students to generate new ideas that they probably would not have come up with alone.

In a similar study, Ellis, Klahr, & Siegler (1994) found that fifth graders who collaborated with a partner were able to come up with new mathematical rules that neither partner knew at the beginning of the learning session. During the study, they asked individual and paired students to choose the larger of two decimal fractions, and explain their rationale. They found that 75% of the paired students generated a new, correct explanation during the problem solving session, compared to only 32% of the students working individually.

These studies suggest that the ability of a group may somehow transcend the abilities of its individual collaborators. Wegner (1987) proposes that this collaborative learning "effect" may aided by a transactive memory system – a sort of distributed group "brain" that performs encoding, storage, and retrieval of information. Social psychologists use this concept of a transactive memory to explain the results of experiments in which group members recall more information together than the sum of what they can remember alone. When group members have knowledge of each other's areas of expertise, they can selectively attend to the information that they understand best, and simply remember which group member they can probe for other information. Transactive memory systems are most efficient when group members have established roles, and their areas of expertise are known to the group.

Learning and working with peers may benefit not only the overall team performance by increasing the quality of the team product; it may also enhance individual performance. Vygotsky (1978) explains that learning in collaboration with others is necessary for the development of one's own cognitive processes. As early as 1937, Bos (1937) showed that students could learn to formulate ideas and opinions more effectively by communicating them to other students. In his study, 11-13 year olds worked individually or in pairs to identify the artist in a set of paintings, or to place pictures in a sequence that would tell a story. Bos found that students who worked in pairs produced more accurate responses than students who worked alone. He also maintained that children are disadvantaged in a group if they have correct conceptions, but cannot communicate them effectively to their teammates.

The next three subsections summarize a few of the prominent viewpoints on how collaboration might enhance individual learning, and describe their supporting studies. Underlying each of these viewpoints is the importance of supporting effective knowledge sharing and transfer, which is the focus of this dissertation. In this chapter and the next, I provide a summary of research describing the various circumstances under which collaboration might enhance or hinder individual learning. I argue that collaborative learners, regardless of whether they are classroom or distance learners, need support and guidance. Chapter 4 concludes my literature review by describing the approaches researchers have taken in the past to address the issue of support. This review of literature will lay the foundation for my claims and analyses, which begin formally in chapter 5. The reader already familiar with collaborative learning and CSCL research may choose to skip ahead to this chapter.

2.1 Collaboration as a Means for Inducing Cognitive Change

Determining the benefits of collaboration requires a clear set of metrics that help to assess the effect collaboration has on individual learning. One set of metrics was developed by drawing upon constructivist views (Groen, 1978), which see cognition and learning as resulting from social processes. Theories of sociocognitive conflict and cognitive change prescribe socially motivated learning metrics that assess the degree to which the student's knowledge base changes during the course of the collaborative learning session, as a result of the conflict in the interaction. Conflict, in this view, provides the trigger for conceptual change (Damon, 1984).

A classic piece of work by Doise, Mugny, & Perret-Clermont (1975) shows how collaboration facilitates the transfer of knowledge that can later be used during individual activities. They performed two experiments, the first of which had 6 and 7 year olds examine a physical model of a village, and try to reconstruct it (using physical objects, such as houses) on a table perpendicular to the model. The students worked either individually or in groups. In some cases, the students needed to mentally rotate the model and/or the locations of the buildings. Doise, Mugny, and Perret-Clermont found that the children that worked in groups performed superior to students working alone. Performance measures included (1) a measure of how much each component of the solution differed from the model, and (2) the number of houses correctly placed on grid. Further analysis showed that social interaction did not improve performance for the simple items, but it did improve performance for the complex items. Doise and his colleagues speculated that the groups produced more points of view, increasing the chances that conflict would occur and provoke discussion, and hence learning.

Doise, Mugny, and Perret-Clermont's second, and more often cited experiment, involved a popular Piagetian conservation task. Students were asked to pour equal amounts of juice into three containers: one short and wide, one tall and thin, and one regular. From their performance on this task, the children were classified as either a conserver (C), a non-conserver (NC) , or an intermediate (I). Conservers are students who understand clearly that quantities are conserved, regardless of shape of container. The non-conservers were then grouped with 2 other conservers. After working with conservers, 24 of the 37 children (65%) progressed from NC → I → C. Furthermore, 13 of those students who progressed, produced good arguments on the post test which were not generated during the social interaction. The interaction, for these children, helped them elaborate their internal operational structures. It is also interesting to note that the most effective discussions were those in which the conserving (C) children defended their own viewpoints in a coherent and consistent manner. This seminal work legitimized the concept that cognitive structures which are formed during social interaction are reactivated while working individually.

2.2 Collaboration as a Method for Stimulating Knowledge Construction

Research that explained the benefits of collaborative learning as resulting from cognitive conflict failed to account for the process of collaborative knowledge construction that group members actively engage in. Heisawn Jeong (1998) explains, in her dissertation, how "[conflicts must be] explained and justified to lead to cognitive change, and why people can even *regress* as a result of experiencing socio-cognitive conflicts." She goes on to explain that, "if people fail to construct the right knowledge in the following interaction, the experience of conflict itself is not likely to induce cognitive change or learning (p.6)."

The era of co-construction of shared knowledge was born in the early 1990s, with the introduction of Resnick, Levine, and Teasley's 1991 text, "Perspectives on a Socially Shared Cognition." In this book, Schegloff talks about a shift in thinking from "common culture/shared knowledge" to "shared cognition". Shared cognition refers to not just shared knowledge, but rather a shared process or set of practices by which actions can be predicted, confirmed, modified, or expanded. In Rogoff's (1991) task, 9 year old children either individually or collaboratively planned an errand-running route to gather specific items. The results of their study showed that children do not benefit from collaborating with a partner unless both partners are engaged in shared, skilled problem solving.

From the shared cognition viewpoint, the degree of shared understanding is used to predict success in the process of knowledge construction, which in turn can be used to predict learning success. Heisawn Jeong (1998) tested this idea in her dissertation. She showed pairs of students 73 sentences about the human circulatory system, and then asked them to individually draw a model of the circulatory system. They were also asked to recall terms, and to answer questions requiring inferencing, and higher-level conceptual knowledge (44 questions in all). Pre-and-Post tests were used to measure learning gains. Jeong found that there was an overall improvement in the students' individual mental models as a result of the collaboration. The students possessed more shared knowledge on the post test (31%) than on the pre-test (23%), and this difference was not a mere reflection of knowing more having participated in the study, since the percentage of shared knowledge increased (not just the number). Perhaps most interesting is that the amount of shared knowledge was positively correlated with individual student performance, especially on questions that required inferencing. This research justifies the concept that collaboration leads to co-construction of knowledge, and that co-construction of knowledge, in turn, is important to learning.

Okada and Simon (1997) suggest that the processes of co-construction and cognitive conflict may not be inseparable. The co-construction of new knowledge is often aided by mismatches, or conflicts, in students' knowledge bases. When students share their knowledge with each other, these conflicts become evident, encouraging the students to explore new options. In their study, Okada and Simon (1997) had individuals and pairs of students answer scientific discovery questions by performing experiments using a computer simulation of a molecular genetics laboratory. They found that the dyads engaged in more explanatory activities, and entertained twice as many hypotheses as the individuals. This result makes sense if the processes of idea generation and explanation play major roles in the co-construction of new knowledge. Idea generation and explanation are two fundamental ways that students share knowledge with each other. I will revisit this concept again in the next chapter.

2.3 Collaboration as a Mechanism for Supporting Social Skill Development

Learning with peers may not only enhance students' learning of the subject matter, but may also give students an opportunity to practice the social skills they need in their everyday lives. Brown and Palincsar's (1989) seminal work (described in the next chapter) is often cited for its impact on individual performance and improvement, however their interests were more socially oriented:

> "Improved retention of the content of a particular set of materials, although desirable, may not be the primary benefit of group participation. Practice discussing, defending, and evaluating one's opinions and those of others may result in improved ability to learn about future text content, a learning to learn effect that would be far more beneficial than gains on any one set of factual material. (p. 402)"

Barnes and Todd (1977) performed an in-depth qualitative analysis of 13 year olds' group discussions in the classroom. Their book, which provides excellent examples of students learning cognitive strategies (i.e. proposing a cause, evaluating), also provides convincing accounts of students developing social skills such as the management of cooperation and competition, the giving of mutual support, and the ability to manage the group process. From Barnes and Todd's perspective, collaborative learning enhances individual learning by providing students with the opportunity to learn cognitive skills, while at the same time practicing the social skills that are so essential in our social environment.

The advantages that group work offers individuals is often measured by examining the individual performance before, and after, the group session. This sort of procedure, however, does not take into account the socially embedded outcomes of the group process that cannot be assessed though individual performance measures. Socially oriented outcomes, such as mutual concern, and the ability to share, cooperate, and lead, are difficult variables to measure. At this point, we can only speculate on the social benefits the collaborative learning experience offers.

In this chapter, we saw how both individual learners and learning groups may benefit from collaboration. By stimulating cognitive change and knowledge construction, collaboration facilitates the generation, transfer, and understanding of knowledge, all of which are important for learning. As an added benefit, working with peers may also help students develop the social skills that are so essential in today's society. In the next chapter, we will see why these benefits are not always realized, and in the following chapter (chapter 4), we take a look at what might be done when the collaborative learning process fails.

3 When Group Interaction Fails to Improve Individual Performance

In the previous chapter, we saw how the collaborative learning experience has the potential to motivate students to seek new insights and perspectives, ask questions openly, and practice explaining difficult concepts, thereby gaining a better understanding of the domain (Doise, Mugny, & Perret-Clermont, 1975). The extent to which these benefits are realized depend largely on the effectiveness of the group interaction.

Salomon and Globerson's (1989) research gives evidence that teams do not always "function the way they ought to." In one of their studies, individuals and pairs of students were asked to read and recall information from text passages. The paired students recalled less of the content, and showed less improvement than the individual learners. From their experiences, Salomon and Globerson (1989) provide an explanation of the sorts of debilitating effects that can occur in groups. For example, the "free rider" effect describes the activity that students are engaging in (i.e. taking a "free ride") when their teammates do all the work. The "ganging up on the task" effect happens when groups collaboratively look for ways to complete the task with minimal effort.

Davidson (1985, as cited by Cohen, 1994) surveyed research in mathematics education that compared cooperative to individual learning, and found that in 2/3 of the cases, cooperative learning produced no significant advantages. When students do not collaborate effectively, the social and cognitive advantages of group learning are lost. In this chapter, I discuss a few of the variables that interact to determine whether or not a group will learn effectively.

3.1 Revisiting Conflict in Group Learning

Recall (from the previous chapter) that according to cognitive conflict theory, students learn by experiencing an internal equilibrium imbalance, and this conflict provokes cognitive change. Perret-Clermont, Perret, and Bell (1991) later elaborated on the results of Doise et al.'s Piagetian conservation experiment (described earlier), explaining that nonconservers do not need to be paired with a conserver in order to learn the correct response. The nonconserver need only be paired with another student with a different opinion, so as to challenge the original (incorrect) opinion. But they added that sociocognitive conflict does not always result in developmental progress. Two conditions must be fulfilled: (1) students must have the necessary cognitive prerequisites, and (2) social confrontation is only fruitful if the cognitive gap between the partners is not too wide.

Bearison, Magzamen, and Filardo (1986, as cited by Webb & Palincsar, 1996) had students perform a task similar to Doise et al.'s first experiment, and found that students who engaged in very infrequent or very frequent disagreements performed less well than students who engaged in a moderate amount of disagreement. It seems that some conflict is important for promoting cognitive change, but too much can prevent group progress.

Conflict is just one variable that may affect collaborative learners' performance. In the next section, I provide a glimpse of the many other interacting variables that help determine whether or not a group will succeed.

3.2 The Intricate Web of Collaborative Learning Variables

Since collaboration does not always enhance individual learning, the next logical question is, *when* does collaboration enhance individual learning, and *how* can we prevent the sorts of effects that Salomon and Globerson (1989) describe? Researchers have addressed these questions by analyzing the many different factors that naturally affect group learning, and by evaluating a number of experimental methods that are expected to enhance learning in a collaborative setting, taking into consideration the changing state and quality of interaction in some cases.

Understanding the interdependencies between the many factors that influence group learning, and using this knowledge to support learning teams continues to be a challenge, both on line and in the classroom. As Webb and Palincsar (1996) explain, studying group learning involves much more than studying a synthesis of individual behaviors:

> "Consider the numerous intraindividual factors (e.g. prior knowledge, motivation, language) that influence the learning of one child in "individualistic" activity. Place this learner in a group context, and not only does one have to contend with all the issues that attend this interaction among the group members (from the very mundane resource issues to the more lofty issues of attaining intersubjectivity), but in addition, other intraindividual factors that may have receded into the background when considering

> individualistic activity now emerge as salient, indeed critical (e.g. the learner's gender and social status)." (p. 867)

Just as supporting individual learning requires an understanding of individual thought processes, supporting group learning requires an understanding of the processes of collaborative learning. The outcome of the collaborative learning process is shaped by not only the individual's ability, learning style, and motivation, but also the group members' individual behaviors, and the dynamics of their interaction. This makes it excessively difficult to predict learning outcomes. Even tried and true methods that distinguish effective individual learners will have similar effects, but different manifestations in the face of collaboration. This dissertation research has attempted to bring together cognitive, social, and computational perspectives to develop advanced methods for analyzing, modeling, and supporting online collaborative learning activities.

The long-term vision for this research is to help distance learning students interact effectively, so that they may maximize their potential learning gain. Many different factors may influence group dynamics, which in turn influence student learning. Some of these factors include group composition and cohesion, group members' status and gender, group size, task structure, student and teacher roles, discourse styles, nature of facilitation, rewards or incentives, training in communication skills, group processing, and the learning environment (Levine & Moreland, 1998; Webb & Palincsar, 1996). Some of these factors will prove to be more or less powerful online. For example, online users are able to maintain a high level of anonymity, essentially masking their status level (and, in some cases, even their gender), and suppressing the effects of status differentiation. A student's status (and even their perception of their and others' status) can impact individual performance. Cohen (1985, as cited by Cohen, 1994) showed that student status in a group is correlated with interaction, and that the frequency of task-related interaction, in turn, is correlated with learning gains. This result intuitively explains why a student of lower status may be less reluctant to explain, justify, and promote their opinions online than in a face-to-face situation, where their status may be more apparent.

3.3 The Importance of Effective Knowledge Sharing

In the previous chapter, we saw how a group's ability to co-construct knowledge is an important predictor of the value of the group learning experience (Jeong, 1998; Okada & Simon, 1997).

The effectiveness of knowledge co-construction depends on the participants' evolving knowledge bases and the group's ability to share and assimilate the bits of knowledge necessary to construct new knowledge. As the shared bits of knowledge are assimilated into the group thinking process, group members evolve and develop a shared understanding.

Hatano & Inagaki (1991) explain that learners are not always ready to assimilate the knowledge bits offered by their peers. For example, the way the new knowledge is represented may prevent the receivers of the knowledge from easily incorporating it into their own mental representations. Sometimes, the new information must be offered in a persuasive manner, or presented by a highly respected person (i.e. teacher, mentor). Especially when the receivers of the new information form the majority (with a different opinion than the information givers), it may be particularly difficult for the receivers to accept the new information, unless assured of its plausibility.

Understanding how knowledge is shared and assimilated in groups, and supporting this process, is critical to ensuring that students can establish the common ground necessary to co-construct knowledge. From an intuitive standpoint, the knowledge that group members bring to bear on the problem, and how this knowledge is shared, understood, and further developed (or not) ultimately shapes the effectiveness of the group learning process.

Experiments designed to study how new knowledge is assimilated by group members are not new to social psychologists. *Hidden Profile* studies (Lavery, Franz, Winquist, & Larson, 1999; Mennecke, 1997), designed to evaluate the effect of knowledge sharing on group performance, require that the knowledge needed to perform the task be divided among group members such that each member's knowledge is incomplete before the group session begins. The group task is designed such that it cannot be successfully completed until all members share their unique knowledge. Group performance is typically measured by counting the number of individual knowledge elements that surface during group discussion, and evaluating the group's solution, which is dependent on these elements.

Surprisingly, studying the process of knowledge sharing has been much more difficult than one might imagine. Stasser (1999) and Lavery et al. (1999) have consistently shown that group members are not likely to discover their teammates' hidden profiles. They explain that group members tend to focus on discussing information that they share in common, and tend not to share and discuss information they uniquely possess. Moreover, it has been shown that when group members do share information, the quality of the group decision does not improve (Lavery et al., 1999; Mennecke, 1997). There are several explanations for this. First, group members tend to rely on common knowledge for their final decisions, even though other knowledge may have surfaced during the conversation. Second, “if subjects do not cognitively process the information they surface, even groups that have superior information sharing performance will not make superior decisions (Mennecke, 1997).” Team members must be motivated to understand and apply the new knowledge.

At least one study (Winquist & Larson, 1998) confirms that the amount of unique information shared by group members is a significant predictor of the quality of the group decision. More research is necessary to determine exactly what factors influence effective group knowledge sharing. One important factor may be the complexity of the task. Mennecke (1997) and Lavery et al.’s (1999) tasks were straightforward, short-term tasks that subjects may have perceived as artificial. Tasks that require subjects to cognitively process the knowledge that their teammates bring to bear may reveal the importance of effective knowledge sharing in group activities. This dissertation employs such a task.

4 Supporting Collaborative Learning Activities

Collaboration may enhance individual learning by influencing the way people think and learn in the presence of others. Or, it may enhance learning by providing an environment that naturally supports a number of methods that are thought to improve individual performance. For example, Lesgold et al. (1992) showed how individual performance is enhanced when students are encouraged to pose problems to each other during a reflective follow-up session. King, Staffieri, and Adelgais (1998) showed that student performance on knowledge construction tasks, which involve inferencing and integrating presented material, is enhanced when they are trained to ask thought provoking questions, and explain and elaborate their responses. In this section, I survey a number of other methods that researchers have used to foster effective collaborative learning interaction in the classroom and online.

4.1 Fostering Effective Collaborative Learning in the Classroom

Brown and Palincsar's (1989) Reciprocal Teaching method produced significant and impressive results involving transfer of cognitive skills. In this method, the teacher and group members take turns playing the role of group leader, and give guidance and feedback to the leader about the needs of the group. Then the teacher's presence gradually fades away as the students become more competent to control their own learning. The students are instructed to take turns questioning, summarizing, clarifying, and predicting in order to construct a shared understanding of text passages. Brown and Palincsar (1989) observed student participation in the discussions, and tested their reading and retention of novel passages daily. They found that 71% of their junior high school students who were taught using the reciprocal teaching method reached a stable level of over 70% on the tests, while less than 20% of the controls reached this criterion. Students in the reciprocal tutoring groups progressed from passive participants to competent discussion leaders over the course of about 20 days.

Webb and Farivar (1994) explain that collaborative learning enhances individual learning when peers are able to address each other's needs. Students in a group who are all learning the material for the first time may have a better sense (than the teacher) of what their peers do not understand, and may be able to provide more relevant explanations. Webb and Farivar (1994) studied 166 students (55% Latino, 26% White, 15% African American) in six 7^{th} grade math classes. Groups of 4 were constructed, and the experimental condition received training in communication and helping skills, while the comparison condition received training in only communication skills. Although the differences between the two samples (experimental and comparison) on pretests and posttests were not significant, the minority students gave and received more elaboration in the experimental condition than in the comparison condition. Also, the minority students in the experimental group performed better on 3 of the 4 post-tests than the minority students in the comparison group. This study also showed that giving and receiving elaborated explanations was related to achievement on post-tests, and receiving answers without elaboration was not related to posttest achievement. This result is consistent with Webb's previous research (see Webb & Palincsar, 1996 for a summary).

Slavin (1980) studied the effects of imposing various reward and task structures on learning groups. For example, reward interdependence describes the degree to which group members share the rewards (or penalties) for performance. If one group grade is assigned to all group members, then the task is said to have high reward interdependence. Task interdependence describes the degree to which students must rely on each other to complete the task.

The Jigsaw method uses high task interdependence, but low reward interdependence. In this method, the task is broken into distinct pieces, and each group member is responsible for becoming an expert in his or her piece. The students study with the members of other groups who have the same piece, and then return to their teams to share what they have learned. Students are graded on individually, and there is no formal group goal. Although Slavin (1980) found mixed results for achievement using Jigsaw, he did find that this method has a positive effect on students' self-esteem. It is possible that giving each student unique information to learn helps them feel like a special resource to the team. The experimental procedure carried out for this dissertation was similar to the Jigsaw procedure, in that each student was given a unique piece of knowledge to share with the group. The procedure here differs from the Jigsaw method in that it was intended to model the more common situation in which team members bring their knowledge and experiences to the group, and are not necessarily singled out and identified by the knowledge they hold.

The Team-Games-Tournament method employs high reward interdependence, but low task interdependence. In this method, teams of students study together to help each other learn the academic material. The students then represent their team at tournament tables in which they compete against members of other teams who are of the same ability. During the tournament, the teacher asks questions covering academic content, and the score each student earns is added to his team's overall score. This method has been shown to produce significant academic achievement according to curriculum specific and standardized test results (Slavin, 1980). It has also been shown to improve race relations and mutual concern among students.

The cooperative task and reward structure that will be most effective in the classroom depends partly on the nature of the task. Slavin (1980) argues that students learning low-level material (e.g. calculation, application of principles) are best supported through a structured, focused instruction schedule, a method in which students are held accountable for their individual contributions, and a reward structure in which recognition is given to successful groups. Students learning high-level material (analyzing problems, evaluating), are best supported through less structured cooperative learning techniques that foster high autonomy, participation and decision making.

But we must be careful in our use of group rewards and praise. Dweck (1989) explains that frequently praising students for their success on easy tasks may result in unexpected outcomes for more difficult tasks in which students need to develop and validate their own competence. When external judgments and rewards are imposed, students might focus on obtaining favorable judgment (or avoiding unfavorable judgments), instead of developing their ability to understand and figure things out. Group rewards may also cause high performers to blame low performers for the group's failure (Webb & Palincsar, 1996).

Johnson, Johnson, and Stanne (1985) studied the effects of three different goal structures. In the cooperative goal structure, individual goal achievements are positively correlated, so that when one group member succeeds, his teammates also benefit. In the competitive case, individual goal achievements are negatively correlated, so that one group member's success necessitates his teammates' failure. In the individualistic case, individual goal achievements are not correlated. Seventy-one 8th graders participated in Johnson, Johnson, and Stanne's (1985) study. Either alone, or in pairs, the students used the computer to sail an ancient ship around the new world and back in search of gold, using the sun, stars, & trade wind to navigate. Johnson and his colleagues found that the students in the cooperative condition completed more worksheet items, and got more items correct than the students in the other 2 conditions. Students in the cooperative condition also made more task and management statements, and talked less to the teacher, than the students that worked individually. Also, the competitive case seemed to have a particularly debilitating effect on the female students.

4.2 Fostering Effective Collaborative Learning Online

The rapid advance of networking technology has enabled universities and corporate training programs to reach out and educate students who, because of schedule or location constraints, would not otherwise be able to take advantage of many educational opportunities. This new technological capability demands software that can support structured, on-line learning activities; thus we have recently seen the rapid development of computer-supported collaborative learning (CSCL) systems (Guzdial et al., 1997; Jermann & Dillenbourg, 1999; Scardamalia & Bereiter, 1994; Singley, Fairweather, & Swerling, 1999; Suthers et al., 2001). CSCL systems offer software replicas of many of the classic classroom resources and activities. They may provide shared workspaces, on-line presentations, lecture notes, reference material, quizzes, student evaluation scores, and facilities for chat or on-line discussions. These tools enable students to interact and engage in learning dialogs with other students in real time, but they provide no guidance or direction to students during or after these dialogue sessions.

In chapter 2, we saw how effective collaboration with peers has the potential to create a successful and uniquely powerful learning experience for students (Brown & Palincsar, 1989; Doise, Mugny, & Perret-Clermont, 1975). Students learning effectively in groups encourage each other to ask questions, explain and justify their opinions, articulate their reasoning, and elaborate and reflect upon their knowledge, thereby motivating and improving learning. In chapter 3 we saw how these benefits do not necessarily apply to all learning teams; they seem to favour active and well-functioning learning teams (Brown & Palincsar, 1989; Jarboe, 1996; Salomon & Globerson, 1989). In the classroom, placing students in a group and assigning them a task does not guarantee that the students will engage in effective collaborative learning behavior. Students learning online via CSCL technology need guidance and support, just as students learning in the classroom need support from their instructor.

Should we use our computational resources to promote cognitive conflict and conceptual change (Teasley & Roschelle, 1993), offer group rewards (Slavin, 1980), support social aspects of interaction (Cohen, 1994), or assign students roles (Brown & Palincsar, 1989) such as peer helper or group leader? Supporting collaborative learning means accounting for the spectrum of activities that groups engage in while learning. Separating a student's participation from the quality of his contributions, or studying discourse and action separately, may produce an inadequate understanding of the group activity. The many factors that influence collaboration should be viewed as corresponding to pieces of a pie that represents a comprehensive model of group interaction and learning. We do not yet have an adequate understanding of what a computer-based collaborative learning pie looks like, although we have begun to discover some of the contributing pieces. For example, Muehlenbrock and Hoppe (1999) have made progress in analyzing student actions on a shared workspace (but they do not attempt to analyze dialog), whereas Constantino-Gonzalez and Suthers (2001) study specifically the interaction between student participation, opinions, and differences in structured representations.

The next 3 subsections describe a few approaches taken by CSCL developers to analyze and support on line collaborative learning. More research, however, is needed to understand the various factors that influence distance learning groups, and recognize and support their activities. The research presented in this section should be viewed as a set of tools that are designed to support various aspects of collaborative learning. Many of these tools may be applied in conjunction with one another, depending on the interests of the instructors and researchers, and the needs of the group. This dissertation contributes to the implementation of this toolbox by exploring a new machine learning approach, described in the next chapter, to assess and support knowledge sharing behavior.

4.2.1 Finite State Machines

The Coordinator (Flores, Graves, Hartfield, & Winograd, 1988) was one of the first systems to adopt a finite state machine approach. A finite state machine defines a set of possible states, the actions that can be taken in each state, and the transitions that lead from one state to another. In Flores et al.'s view, conversations represented intentions to take actions in an organization. Users sent messages to each other by choosing conversational acts (such as *Request* or *Promise*) from menus set up by the system. The system dynamically generated these menus based on a state transition matrix of "sensible next states", displaying only those actions that would direct the conversation toward completion of action. The Coordinator was intended to create organizational change by making the structure of conversation explicit. Consequently, the first versions were often regarded as overly coercive.

McManus and Aiken's (1995) Group Leader system compared sequences of students' conversation acts to those allowable in four different finite state machines developed specifically to monitor discussions about comments, requests, promises, and debates. The Group Leader was able to analyze sequences of conversation acts, and provide feedback on the students' trust, leadership, creative controversy, and communication skills[1]. Conversation acts were derived from the phrases, or *sentence openers*, that students were required to select (from a menu) to begin each contribution. For example, a student who begins his sentence with the opener "You think that" indicates to the system that he is performing a Paraphrasing act. The Group Leader might note a student's limited use of sentence openers from the creative controversy category, and recommend that the student, "use the attribute of preparing a pro-position by choosing the opener of 'The advantages of this idea are'". The Group Leader received a positive response by the students, and paved the way for further research along these lines.

Inaba and Okamoto (1997) describe a model that draws upon the ideas of finite state machines and utility functions. Utility functions are used to numerically describe the desirability of a state (Russell & Norvig, 1995). Inaba and Okamoto (1997) used a finite state machine to control the flow of conversation and to identify proposals, while applying utility functions to measure participants' beliefs with regard to the group conversation. For example, the utility function for evaluating a student's attitude took into account the degree to which his teammates agreed with his proposals. Hybrid approaches such as this are key, as they broaden our ability to analyze interaction in new ways.

Barros and Verdejo's (1999) asynchronous newsgroup-style environment enables students to have structured, computer-mediated discussions on-line. Users must select the type of contribution (e.g. proposal, question, or comment) from a list each time they add to the discussion. The list is determined by the possible next actions given by a state transition graph, which the teacher may specify before the interaction begins. In this case, the state transition graph provides a mechanism to structure, rather than to understand, the conversation. Evaluating the interaction involves analyzing the conversation to compute values for the following four attributes: initiative, creativity, elaboration, and conformity. For example, making a proposal positively influences initiative and negatively influences conformity. These four attributes, along

---

[1] These four categories were proposed by Johnson and Johnson (1991), and are intended to define the skills involved in small group learning.

with others such as the mean number of contributions by team members and the length of contributions, factor into a fuzzy inference procedure that rates student collaboration on a scale from "awful" to "very good". This work is seminal in combining a finite state approach with fuzzy rubrics to structure and understand the group interaction. A closer look at interaction sequences containing both task and conversational elements may help in composing rubrics for dynamically evaluating learning activity, enabling a facilitator agent to provide direction at the most appropriate instances.

4.2.2 Rule Learners

Katz, Aronis, and Creitz (1999) developed two rule learning systems, String Rule Learner and Grammar Learner, that learn patterns of conversation acts from dialog segments that target particular pedagogical goals. The rule learners were challenged to find patterns in the hand-coded dialogs between avionics students learning electronics troubleshooting skills and expert technicians. The conversations took place within the SHERLOCK 2 Intelligent Tutoring System for electronics troubleshooting (Katz et al., 1998).

The String Rule Learner, which searches for string patterns common to a training set, discovered that explanations of system functionality often begin with an *Identify* or *Inform* Act. The Grammar Learner, which develops a probabilistic context-free grammar for specified conversation types, learned that explanations of system functionality not only begin with an *Inform* statement, but may go on to include a causal description, or another *Inform* Act followed by a *Predict* Act[2].

Katz, Aronis, and Creitz's (1999) research focused on analyzing the interaction between a tutor and two students. Their approach, however, could be used to analyze group learning dialog (S. Katz, personal communication, November 9, 2001). Rule learning algorithms such as this hold promise for classification and recognition tasks in general, and may prove useful tools for assisting in the sequential analysis of peer learning conversations.

4.2.3 Decision Trees and Plan Recognition

In his dissertation, Martin Muehlenbrock (2001) introduces *activity recognition*, a plan recognition approach that analyzes patterns of student actions in context, and identifies those that

[2] The coding terminology used here has been altered from the original for brevity and clarity.

indicate coordination or conflict. Interaction patterns are analyzed by mapping actions taken on a shared workspace to steps in a partially ordered, hierarchical plan. The hierarchical nature of the plan allows users' individual actions to be generalized to problem solving activities (e.g. conflict creation or revision). Using this method, Muehlenbrock shows that it is possible for a system to differentiate between phases of constructive and non-constructive problem solving activity.

Constantino-Gonzalez and Suthers (in press) system, COLER, coaches students as they collaboratively learn Entity-Relationship modeling, a formalism for conceptual database design. Decision trees that account for both task-based and conversational interaction are used to dynamically advise the group. COLER's private and shared workspaces help students develop their ideas independently and facilitate the construction of shared knowledge, while decision trees that account for both task-based and conversational interaction are used to dynamically advise the group. Students are required to express their agreement or disagreement (by clicking on "Agree", "Disagree", or "Not Sure" buttons) each time an item is added or changed on the group's shared workspace. This information, along with student participation statistics, and differences between students' private and group workspaces, is used by COLER's personal coaches to dynamically facilitate the group. For example, Jim's COLER coach might observe his teammate adding a node to the group's shared diagram, and might notice that this node is missing in Jim's private diagram. If Jim disagreed with his teammate's new addition, his coach might then recommend that the two students discuss a few alternatives so that they may learn from each other, and perhaps come to consensus.

Both Constantino-Gonzalez and Suthers (2001), and Muehlenbrock and Hoppe (2001) have implemented novel ways to analyze group members' actions on shared workspaces, and have successfully inferred domain independent behaviors from information based on the frequency and types of domain related actions. A computer tutor that furthers this research by attempting to understand the rich conversation between peers as they discuss their problems, ask questions, and probe their teammates for explanations, might be even better suited to address the pedagogical and social needs of the learning group. More work is needed to understand how students communicate, and to apply this knowledge in developing computational methods for determining how to best support and assist the process of collaboration.

4.2.4 Discussion

Each of the approaches described in this chapter targets a different type of interaction (e.g. action sequences, dialog sequences), or a different analysis perspective (e.g. conflict, participation). The selection of an analysis method should be driven by the desired outcome: to better understand the interaction, or to provide advice or support to the students. In the interest of impacting the group learning process, the result of the analysis should reveal occurrences of events that the system knows how to target.

These researchers' findings should not be viewed independently, but rather as a toolbox of methods and strategies for understanding and supporting various aspects of online collaborative learning behavior. This toolbox reflects the perspectives of both the software designer and the educational practitioner, enabling the marriage of theory and implementation. Modeling collaborative learning activities means modeling both verbal and nonverbal interactions, and both task and social aspects of group learning. Studying these aspects separately allows researchers to deal with difficult issues (such as natural language understanding), while controlling for the variability inherent in collaborative learning. Future research along these lines should help to develop a more complete toolbox of methods for computationally analyzing collaborative learning activities. With a more complete toolbox at hand, researchers may be better suited to adopt holistic views of supporting collaborative learning communities. This dissertation assists in the development of this toolbox by introducing and testing a new tool that is intended to aid in the analysis of a critical element of collaborative learning – knowledge sharing.

5 Supporting Knowledge Sharing Online

In the previous two chapters, we discussed the various factors that contribute toward effective collaborative learning, and saw the importance of effective knowledge sharing. We reviewed an array of tools designed to support various aspects of collaborative learning, but found a lack of tools appropriate for computationally assessing and supporting knowledge sharing. This chapter begins our formal discussion of how to support online knowledge sharing activities. A proof of concept framework for computationally supporting knowledge sharing will be presented in the first three sections, followed by a formal introduction to Hidden Markov Models, and a summary of my claims.

Imagine that a group of students gathers around a table to solve a problem, and begins to exchange the knowledge that each brings to bear on the problem. Each group member brings to the table a unique pool of knowledge, grounded in his or her individual experiences. The combination of these experiences, and the group members' personalities and behaviors will determine how the collaboration proceeds, and whether or not the group members will effectively learn from and with each other (Brown & Palincsar, 1989; Dillenbourg, 1999; Webb & Palincsar, 1996).

If we take a closer look at the interaction in such a group, we might see that the way in which a student shares new knowledge with the group, and the way in which the group responds, determines to a large extent how well this new knowledge is assimilated into the group, in particular whether or not the group members learn the new concept. It is reasonable to assume that, in effective knowledge sharing conversation, the presentation (sharing) of new concepts and ideas might initiate questioning, explaining, and critical discussion. Studying the interaction that provokes and follows knowledge sharing events may help us assess the ability of the group to assimilate new information that group members naturally bring to bear on the problem.

Group members that do not effectively share the knowledge they bring to the group learning situation will have a difficult time establishing a shared understanding, and co-constructing new knowledge. These difficulties are then reflected as poor learning outcomes (Jeong, 1998; Winquist & Larson, 1998), making research efforts to understand and support student knowledge sharing activities essential. But such research efforts are complex endeavors, involving analysis of student learning, understanding, conversation, and physical actions. It is also probable that the results of these efforts may be applied in analyzing and supporting other complex aspects of collaborative learning, such as the joint construction of shared knowledge and cognitive conflict. Furthermore, research along these lines may help to define guidelines about the limits on the kinds of support a computer-mediated networked communication environment, in general, might offer.

This research focused on interaction between students learning in groups of three. Groups of three were chosen because, although research in analyzing online collaborative learning is rich with examples of dyads working together to solve problems (e.g. Joiner, 1995; Baker & Lund, 1997), no comparable body of research exists for groups of three or more (but see Constantino-Gonzalez & Suthers, 2000, and Inaba & Okamoto, 1997). Research involving groups of three or more students may scale up more easily to groups of 4, 5 and 6, while research involving groups of only two may be applied only to other groups of 2. For example, a question asked by a student working in a dyad is understood to be directed toward his partner. In a group of 3, a student can put a question "on the table". Depending on the level of activity in the group, and the students' abilities to answer the question, it may or may not get answered, whereas in the dyad case, the question recipient is expected to respond. The interaction structure in groups larger than two tends to be less rigid, with non-contiguous and overlapping conversational turns, and multiple agendas being discussed concurrently. Also, determining which students understand the concepts that are being discussed is more straightforward in the dyad situation, because of the rigid turn-taking structure, and more difficult when more participants weaken the link between participation and understanding.

The domain chosen for this work was Object-Oriented Analysis and Design. Software engineers use Object-Oriented Analysis and Design methodologies to construct graphical models for optimizing their designs before implementation, and to communicate design decisions. These models are also useful for preparing documentation, or designing databases. Object-Oriented Analysis and Design was chosen because it is usually done in industry by teams of engineers with various expertise, so it is an inherently collaborative domain that lends itself well to the study of knowledge sharing. This research may be applied to other, similar domains that require teams of people learning to solve complex problems that requiring inference and idea generation.

In the next section, I sketch a hypothetical proof of concept system for supporting knowledge sharing activities online, and in the remainder of this chapter, I describe in detail my contribution to this effort.

5.1 The Vision: A Proof of Concept Framework

Figure 1 shows a visionary system detailing the components needed to diagnose and support knowledge sharing behavior online. Any collaborative distance learning system requires a networked communication interface, so that the participants can interact via text chat, voice, or some other channel. The proof of concept framework should also include a shared workspace, where the students can jointly construct a diagram. As the students are collaborating, the analysis engine should be running in the background, dynamically assessing the interaction and recommending actions to the coaching agent, who is ideally online monitoring the students while they are learning.

The proof of concept analysis engine should contain three modules. The first module (the dialog segmenter) should segment the dialog as it unfolds, indicating when changes of topic occur. The dialog segmenter should be responsible for determining when students begin sharing their knowledge on a particular topic (where topics are perhaps predetermined by the task requirements), and when discussion of this topic ceases. Each of these discrete segments, or knowledge sharing episodes, should then be sent to the second module (the knowledge sharing analyzer), which would analyze the episode to determine its effectiveness. The knowledge sharing analyzer may use information from the instructional module about how students may be effective or ineffective at sharing knowledge, or it might generalize from representative examples of effective and ineffective knowledge sharing. It would be tasked to determine whether or not the episode is effective, and if not, what might have gone wrong. It would send this information to the instructional module, which would determine how to best support the learning group. The instructional module might control and drive a computer-based coach, or it might inform a human distance learning instructor.

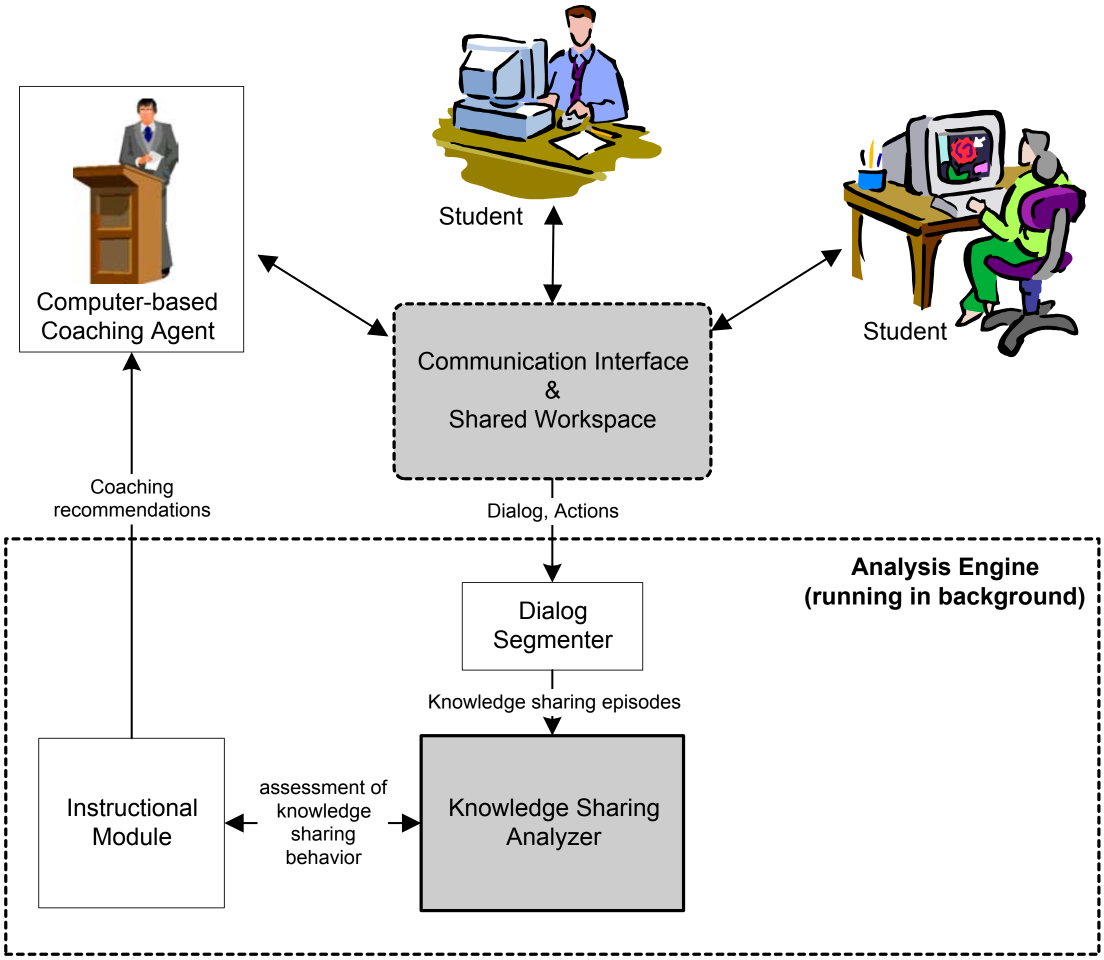


Figure 1: A proof of concept framework for supporting online knowledge sharing during collaborative learning activities

For example[3], suppose the knowledge sharing analyzer has determined that John has just shared a bit of knowledge with his peers, but that this attempt was ineffective because the new knowledge was not discussed, or used by his peers in any sort of constructive activity (Webb, Troper, & Fall, 1995). In this case, the instructional module might recommend that John's peers summarize the new knowledge, or try to apply it to the problem at hand. If the knowledge sharing analyzer had determined that the explanation was insufficient, the instructional module might recommend a different strategy, such as asking John to elaborate, or use an analogy or example in his explanation.

[3] This example is provided for illustrative purposes only, since the instructional module was not developed as part of this dissertation work.

The gray components in Figure 1 show the components I have developed as part of this dissertation work. The dialog segmentation was done by hand, however in chapter 6, I provide guidelines on how to automate this process, and insight from the natural language processing literature on how this process might be facilitated.

Since the segmentation is done offline, the knowledge sharing analysis is also performed post-hoc. No changes are necessary to move the knowledge sharing analyzer to a dynamic, online component, provided that the segmentation is real-time.

The instructional module need not contain any artificial intelligence, and can be developed using lessons learned from the field of educational psychology. In chapter **9**, I provide direction on what sorts of actions might be productive under the various situations recognized by the knowledge sharing analyzer. Since the instructional module, and the computer-based coaching agent are straightforward, they may be developed at a later point, and were not part of this dissertation work.

The next two sections describe the two components that were developed as part of this dissertation work – the interface and the knowledge sharing analyzer.

5.2 The Interface

Distance learning students use two primary communication channels to share knowledge with each other. They discuss and explain their ideas using language, and they show each other representations of their knowledge by illustrating or sketching. The interface designed for this dissertation research supports these two main communication channels by providing a synchronous chat-style (textual) interface, and a shared (what you see is what I see) graphical workspace (Figure 2).

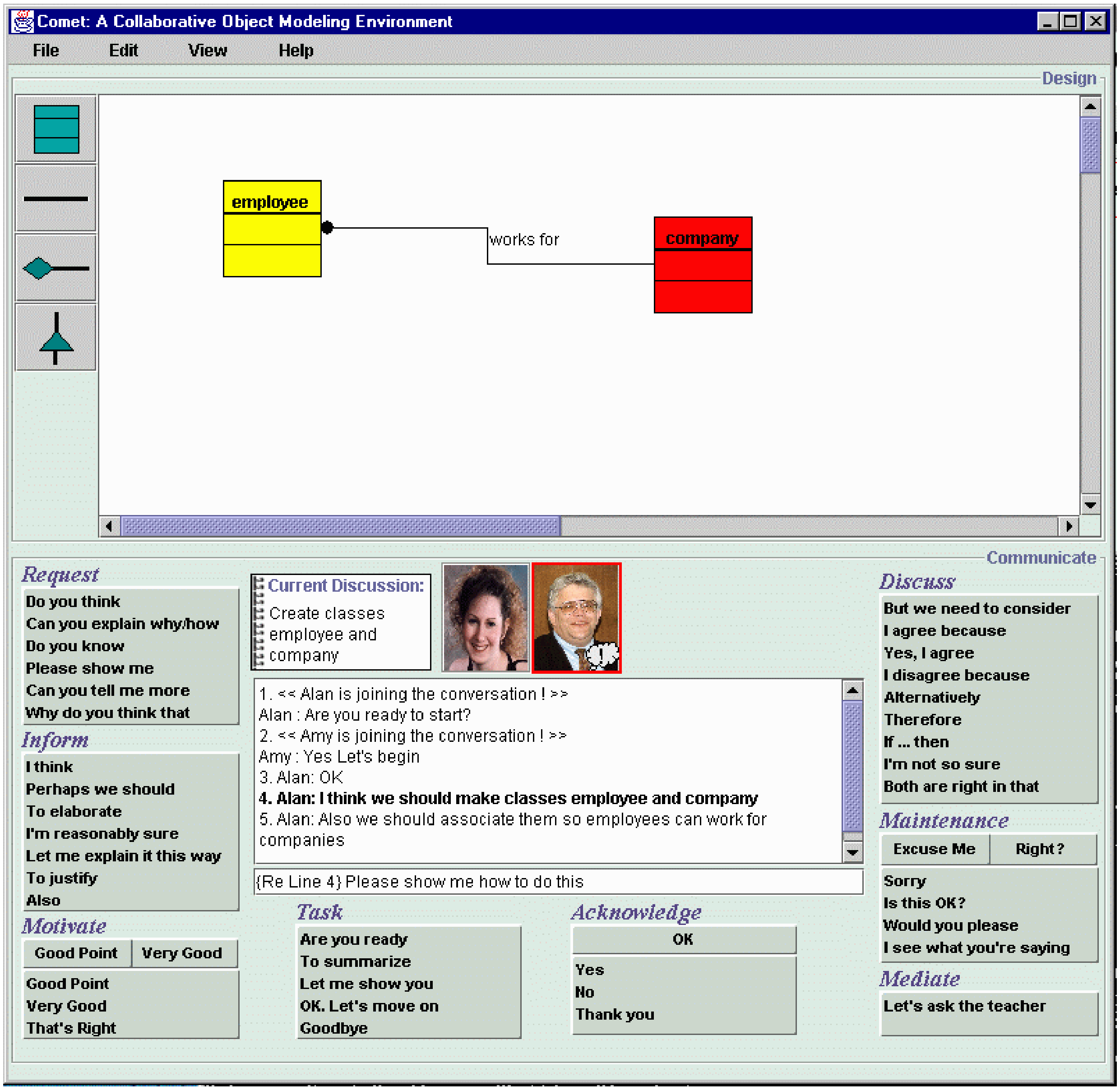


Figure 2: The shared OMT workspace (top), and sentence opener interface (bottom)

Exercise: Prepare a class diagram using the Object Modeling Technique (OMT) showing relationships among the following object classes: school, playground, classroom, book, cafeteria, desk, chair, ruler, student, teacher, door, swing. Show multiplicity balls in your diagram.

The shared graphical workspace is shown on the top half of Figure 2. The workspace allows students to collaboratively solve object-oriented design problems using Object Modeling

Technique (OMT) (Rumbaugh, Blaha, Premerlani, Eddy, & Lorensen, 1991). Object Modeling Technique is a well known Object-Oriented Analysis and Design methodology. Earlier in this chapter, we saw how the knowledge sharing, idea generation, and inferencing required to perform Object-Oriented Analysis and Design made it a good candidate for this research. An example of an OMT design problem is shown below the interface.

The shared OMT workspace provides a palette of buttons down the left-hand side of the window that students use to construct and link objects in different ways depending on how they are related. Objects on the shared workspace can be selected, dragged, and modified, and changes are reflected on the workspaces of all group members.

The interface plays two major roles in supporting knowledge sharing online. It serves as both a medium through which the students communicate, and a source of information about the interaction that the system can draw upon in supporting the group. The amount of structure imposed upon the communication at the interface level has a profound impact on the computation required to analyze and support the interaction at the system level (Jermann, Soller, & Muehlenbrock, 2001). This is especially the case for interfaces that support both shared workspace actions and natural language communication. A seamless, unstructured chat or voice style interface places the burden of understanding the conversation in the system's hands – a place where our current state of the art speech processing and natural language technology, although rapidly advancing, continues to be error-prone, computationally intensive, and time consuming (Stolke et al., 2000). Furthermore, introducing natural language understanding technology means introducing its underlying issues of ambiguity in language, increasing the complexity of the problem substantially.

There have been a few different approaches to dealing with this issue. Muehlenbrock (2001) builds upon the historical concept that, in general, a student's understanding of a concept is reflected in his actions, and his explanations of these actions. In a one-on-one tutoring environment, this information is available and, in most cases, straightforward to analyze. The system is able to watch the student solve the problem, perhaps ask pointed questions to evaluate the student's understanding of key concepts, and once in a while, interrupt him if remediation is necessary. Evaluating the learning of a group of students solving the same problem, however, presents a few new challenges. If one student solves the problem successfully while explaining his actions, and his teammates acknowledge and agree with his actions, to what degree should we assume his teammates understand how to solve the problem themselves? If a student is continually telling her partner what to do, and her partner is simply following her instructions without questioning her, who should get credit for solving the problem? Especially in cases like these, determining which group members understand which material requires some knowledge about how the group conversation relates to the student actions.

Another approach involves the software restricting the students' natural language to a formal language (e.g Tedesco & Self, 2000), or structuring the students' language by having them indicate the conversational intentions underlying their contributions. Cahn and Brennan (1999) explain that a system can represent or model a dialog using only the "gist" of successive contributions; a full account of each contribution, verbatim, is not necessary. The gist of a contribution can often be determined by the first few words, or the *sentence opener*. Sentence openers such as "Do you know", "In other words", and "I agree because", suggest the underlying intention of a statement. Associating these sentence openers with conversational acts such as Request Information, Rephrase, or Agree, and requiring students to use a given set of sentence openers, allows a system to automatically code dialog without having to rely on Natural Language parsers. Sentence openers provide a natural way for users to identify the intention of their conversational contribution without fully understanding the significance of the underlying communicative acts. Previous work has established promising research directions based on approaches that adopt this idea. Most approaches make use of a structured interface, comprised of organized sets of sentence openers. Students typically select a sentence opener from the interface to begin each contribution.

The sentence opener approach was introduced by McManus and Aiken in 1995 and has since shown potential in a few systems that use the phrases primarily as a mechanism for structuring the interaction. For example, Baker and Lund (1997) compared the problem solving behavior of student pairs as they communicated through both a sentence opener interface and an unstructured chat interface. They found that the dialogue between students who used the sentence opener interface was more task focused. Jermann and Schneider's (1997) subjects could choose, for each contribution, to type freely in a text area, or to select one of four short cut buttons, or four sentence openers. Jermann and Schneider discovered that, in fact, it is possible to direct the group interaction by structuring the interface, as Baker and Lund suggest. Furthermore, they found that the use of the sentence openers was more frequent overall than that of the free text zone (58% vs. 42%).

Because of the dialogical constraints of sentence openers, students may not always use them as expected. For example, it is possible to use the sentence opener, “I think”, to say, “I think I disagree”. In order to determine to what degree the students used the openers as they were intended, Soller (2002) had 2 researchers recode 3 of her 5 dialogs (selected at random). The raters were asked to assign each conversational contribution a tag from the coding scheme that best indicates the speaker’s intention. The new tags were then compared to the system’s codes, which describe the sentence openers that the students chose during the dialog. The agreement between the raters and the system was high at the subskill, or category, level (kappa = 0.86), and reasonable at the sentence opener level (kappa = 0.66) (Carletta et al., 1997). This suggested that the students were, in fact, choosing appropriate sentence openers for their contributions.

A combination of approaches that include sentence openers, structuring of student activity, and analysis of student actions may also be used. For example, this dissertation applied an approach that assesses group interaction by analyzing students’ communication patterns, in the form of speech act sequences, and performs a coarser grained analysis of student workspace actions than Muehlenbrock (2001).

The communication interface developed as part of this dissertation[4] is shown on the bottom half of Figure 2. It contains sets of sentence openers (e.g. “I think”, “I agree because”) organized in intuitive categories (such as Inform or Discuss). To contribute to the group conversation, a student first selects a sentence opener. The selected phrase appears in the text box below the group dialog window, where the student may type in the rest of the sentence. Each sentence opener is associated with a particular conversational intention, given by a subskill and attribute. For example, the opener, “I think” corresponds to the subskill (or category) “Inform”, and the more specific attribute, “Suggest”. The categories and corresponding phrases on the interface represent the conversation acts that were most often exhibited during collaborative learning and problem solving in a previous study (Soller, 2001). The full taxonomy of sentence openers, attributes, and subskills is shown in Figure 3, and described in detail in my masters thesis (see Soller, 2001 for a published version).

---

[4] The first version of this interface was developed in 1997, while I was working at the MITRE Corporation on Dr. Brad Goodman’s MSR project entitled, “Intelligent Collaborative Learning Environments.”

In addition to the structured chat-style facility, the communication interface features the following functionality:

– Students can refer back to statements in the dialogue history by selecting the appropriate line (shown in bold in the figure).
– Students may direct their comments to a particular team member by clicking on his or her picture
– A separate agenda window (not shown) enables students to keep track of their work. The current discussion item from the agenda is shown in the "current discussion" notepad above the discussion window.
– While students are typing a contribution, before they hit <Enter> to send their comment to the group discussion window, a small cloud appears on their pictures to let their teammates know they are about to make a contribution
– The picture of the last person to contribute to the discussion appears in a red box.

The student action log (shown in Figure 4) is an integral part of the interface, although it is usually not seen by the students. It records information about what was said or done on the interface, when, and by whom. Each conversation contribution is coded according to the Collaborative Conversation Skills Taxonomy (Figure 3), using the following format:

Line number) Date & Time : Student : Subskill : Attribute : <actual contribution>

Actions are logged in a similar fashion, with variables enclosed in “$” signs. The next section discusses how this log is segmented and processed by the knowledge sharing analyzer.

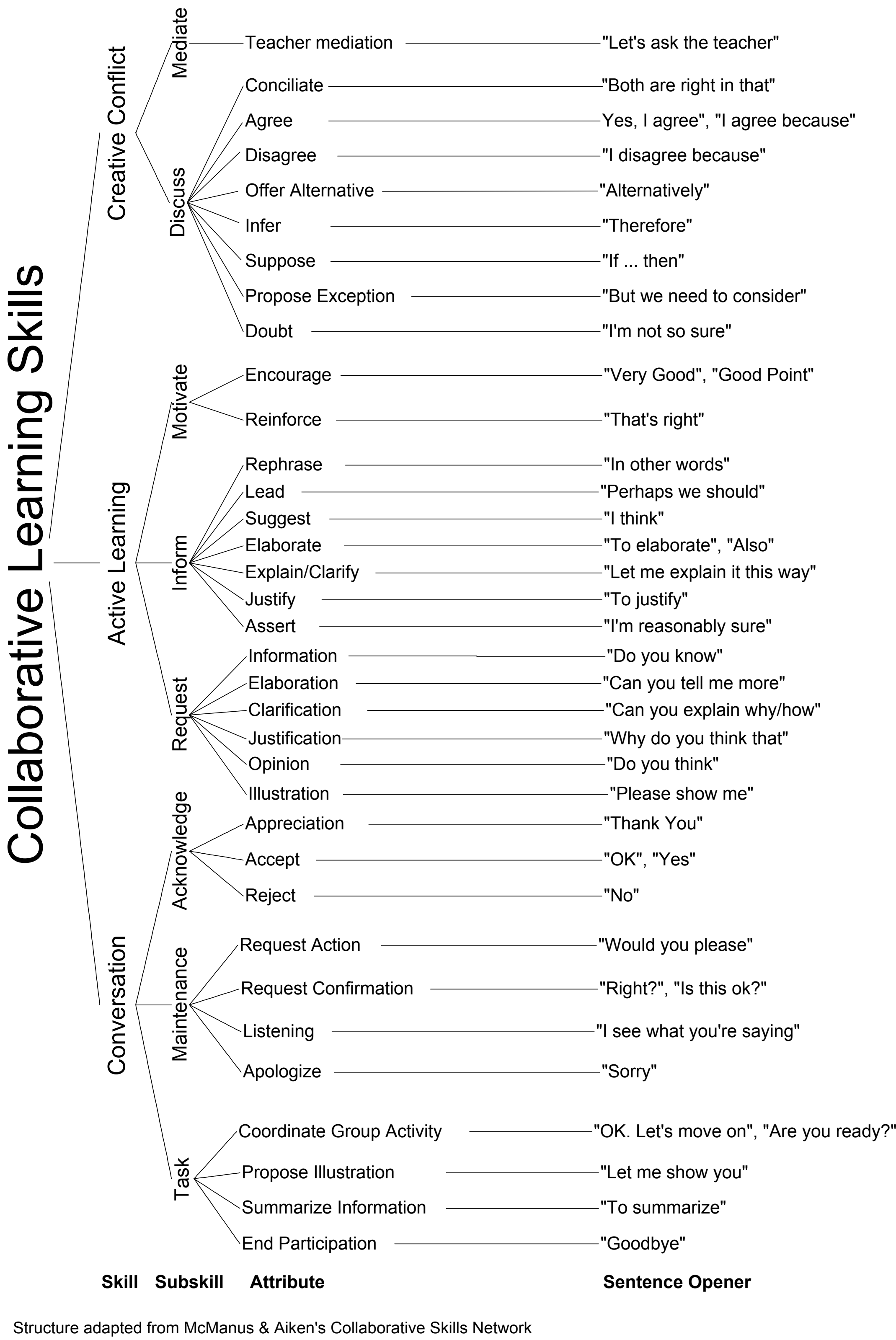


Figure 3: The Collaborative Learning Conversation Skills Taxonomy (Soller, 2001)

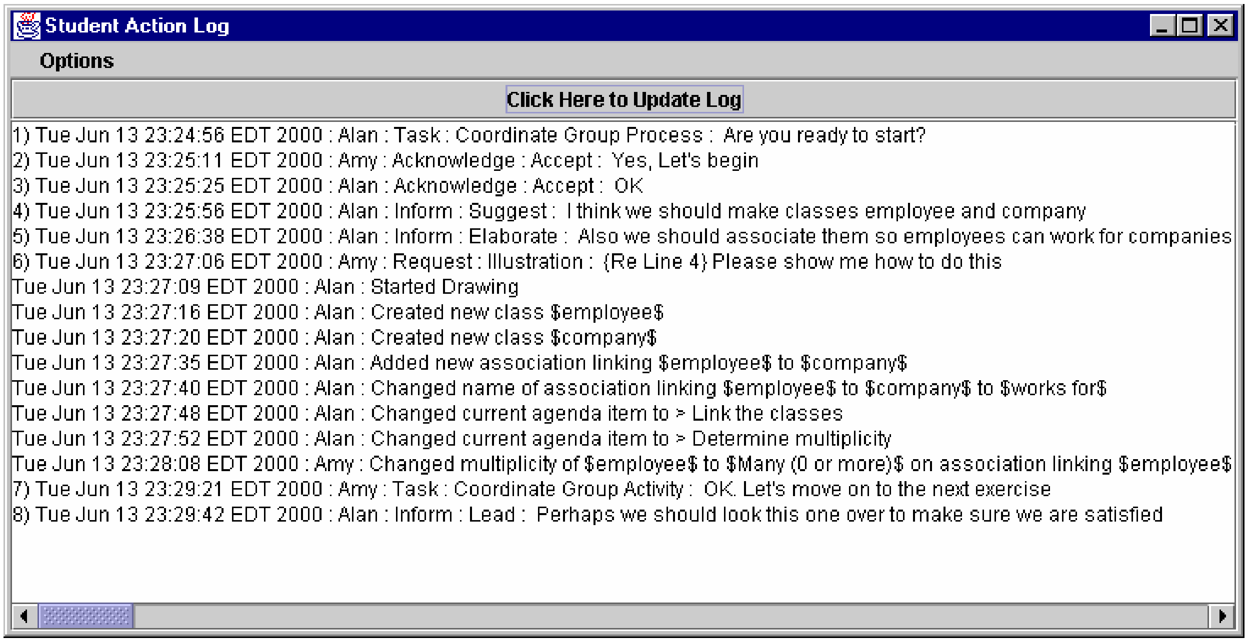


Figure 4: The student action log dynamically records all student actions and conversation

5.3 The Knowledge Sharing Analyzer

The knowledge sharing analyzer is responsible for assessing the coded sequences of interaction to determine their effectiveness. The analyzer assumes that the input is a *knowledge sharing episode* – a sequence in which one group member attempts to share new knowledge with her teammates. Formally, I define a *knowledge sharing episode* as a series of conversational contributions (utterances) and actions (e.g. on a shared workspace) that begins when one group member introduces new knowledge into the group conversation, and ends when discussion of the new knowledge ceases. New knowledge is defined as knowledge that is unknown to at least one group member other than the knowledge sharer.

The dialog segmenter (Figure 1) is responsible for ensuring that all sequences to be analyzed are, in fact, knowledge sharing episodes. To ensure that high-quality knowledge sharing opportunities exist, each group member in my experiments was provided with a unique piece of knowledge that the team needs to solve the problem. This *knowledge element* was designed to mirror the sort of unique knowledge that students might naturally bring to the problem from their own experiences. By artificially constructing situations in which students are expected to share knowledge, I was able to single out interesting episodes to study, and more concretely define situations that can be compared and assessed. I analyzed the team knowledge sharing process by manually selecting the dialog segments in which students share new knowledge with the group (see chapter 6 for more detail), and comparing these segments to the group members' performance on pre and post tests. These tests targeted the specific knowledge elements to be shared and learned during the experiment.

In order to conclude that a knowledge element is shared "effectively", three requirements must be satisfied (F. Linton, personal communication, May 8, 2001):

(1) the individual sharing the new knowledge (the "sharer") must show that she understands it by correctly answering the corresponding pre and post test questions

(2) the concept must come up during the conversation, and

(3) at least one group member who did not know the concept before the collaborative session started (as shown by his pre-test) must show that he learned it during the session by correctly answering the corresponding post-test question.

In this dissertation, I focus on situations in which criteria (1) and (2) are satisfied, since these criteria are necessary for studying how new knowledge is assimilated by collaborative learning groups. Other research has addressed how students individually acquire new knowledge (criterion 1, Gott & Lesgold, 2000), and how to motivate students to share their ideas (criterion 2, Webb & Palincsar, 1996).

In general, analyzing knowledge sharing episodes involves the following three steps:

1. Determining which student played the role of knowledge sharer, and which the role(s) of receiver
2. Analyzing how well the knowledge sharer explained the new knowledge
3. Observing and evaluating how the knowledge receivers assimilated the new knowledge

The knowledge sharing analyzer is responsible for performing these steps, and sending the results of this analysis to the instructional module. If the students are having difficulty explaining new knowledge to each other, information about how and why the breakdown occurred would assist the instructional module in selecting an appropriate support strategy. The process of determining whether or not students are effectively sharing new knowledge requires an assessment of the student-student and student-computer interaction. Studying this interaction over time should provide important insight into the group process. But sequences of student interaction are scattered with noise and interruptions. To complicate matters, counts of marginal statistics of events found in the sequences do not differentiate between effective and ineffective knowledge sharing.

For example, Tricia Chirumbole determined that there were no significant correlations between individual learning and participation counts, including workspace actions, speech, and overall participation (T. Chirumbole, personal communication, October 11, 2000)[5]. Dr. Frank Linton did find some correlation ($r = 0.54$) between counts of student actions (including actions on the workspace, and chat contributions) and the group's success on the task (F. Linton, personal communication, June 28, 2001). Further analysis showed that individual pre/post test score differences are poorly correlated with group success ($r = 0.2$). This meant that looking at individual participation, or assessing the progression of the group diagram, was not enough to determine if the students were learning from each other during this task. This strengthens the argument that some analysis of the peer to peer conversation is essential. In studying the conversation, the only positive and significant correlation that existed was that between the proportion of *Motivation* acts within an individual's dialogue and the individual's learning ($r = 0.54$). This makes sense, because a high proportion of motivation acts may suggest a willingness to listen and learn from others. This trait is particularly important in a situation where each subject's post-test success is dependent upon his learning a piece of knowledge from each of his teammates (T. Chirumbole, personal communication, October 11, 2000). It is interesting, however, to note that viewed as an element in the overall group dialogue, motivation speech contributions did not predict individual learning ($r = 0.16335$).

Further analysis was run to determine if a neural network could be trained to distinguish between effective and ineffective knowledge sharing behavior. A neural network with 5 hidden nodes was trained by constructing feature vectors containing the percentage of time each group member used each subskill during each classified knowledge sharing episode. Since there were 8 subskill categories, and 3 students in each group, this resulted in a feature vector of length 24. A cross validation study showed that the network was unable to distinguish between the effective and ineffective sequences using this information (it achieved less than 50% accuracy).

The neural network may have performed poorly because it did not take into account the progression of events in the knowledge sharing sequences. A look at the pairwise sequences within the knowledge sharing episodes revealed that this ordering is important. The pairwise analysis showed that two of the most common transitions in the effective knowledge sharing sequences were *Discuss-Discuss* (a *Discuss* act followed by another *Discuss* act), and *Discuss-Maintenance*, whereas the common transitions in the ineffective sequences included *Inform-Acknowledge* and *Inform-Inform*. Although the pairwise approach provided insight into the sorts of transitions that differ in effective and ineffective sequences, it was not able to reliably differentiate between these two cases because many of the most common pairs did not occur at all in some of the knowledge sharing sequences. Both a principal components analysis and a factor analysis confirmed this. These methods were tasked to classify the pairwise sequences of knowledge sharing interaction, and neither was able to differentiate between the effective and ineffective knowledge sharing episodes. Although these methods did account for the some ordering in the sequences, they were not able to account for the unpredictable nature of human conversation that makes pattern identification so difficult. In other domains (e.g. Schrodt, 1999), such “noise” has been successfully accounted for by introducing stochastic methods that are designed to model natural random processes.

The marginal and pairwise analyses triggered the search for a more intelligent method that could deal with the ordered, but unpredictable and somewhat noisy nature of human dialog. This is why the stochastic and sequential Hidden Markov Models (HMMs) made them a strong candidate for this research. HMMs were also chosen because of their flexibility in evaluating sequences of indefinite length, their ability to deal with a limited amount of training data, and their recent success in speech recognition tasks (Rabiner, 1989).

In previous work, I obtained promising results in applying Hidden Markov Models (HMMs) to accomplish step (1), and a generalization of step (2) (Soller & Lesgold, in press; Soller, Wiebe, & Lesgold, 2002). My pre-dissertation research suggested that Hidden Markov Models might be able to distinguish between effective and ineffective knowledge sharing episodes. This dissertation confirmed these results with further data, and explored the implications of using HMMs to assess the effectiveness of online collaborative learning. I revisit these theses again in section 5.5, after a brief introduction to HMMs.

---

[5] The analysis presented in this section was done using data from the first five groups run at LRDC.

5.4 An Introduction to Hidden Markov Models

Hidden Markov Models were used to model the sequences of three person interaction that make up knowledge sharing episodes. Before introducing the HMM algorithm, let's take a look at the simple case of the Markov chain. Markov chains are essentially probabilistic finite state machines, used to model processes that move stochastically through a series of predefined states. For example, imagine a robot whom we will call Hal, that wanders through a hospital from room to room performing various duties. Hal's virtual map might include a doctor's office, a ward, and the coffee room (see Figure 5). The probability of Hal entering the ward after visiting the doctor's office might be 0.2, the probability of entering an the coffee room 0.3, and the probability staying in the doctor's office 0.5. In other words, if Hal is in the doctor's office, there is a 20% chance that he will move to the ward, a 30% chance that he will wander to the coffee room, and a 50% chance that he will just stay put. In Markov chains, the arcs describe the probability of moving between states. The sum of the probabilities on the arcs leaving a state must sum to one, and the probability of a sequence of states is the product of the probabilities along the arcs. So, if Hal is in the doctor's office, then the probability that he will move to the ward, and then to the coffee room is (0.2)(0.3) = 0.06.

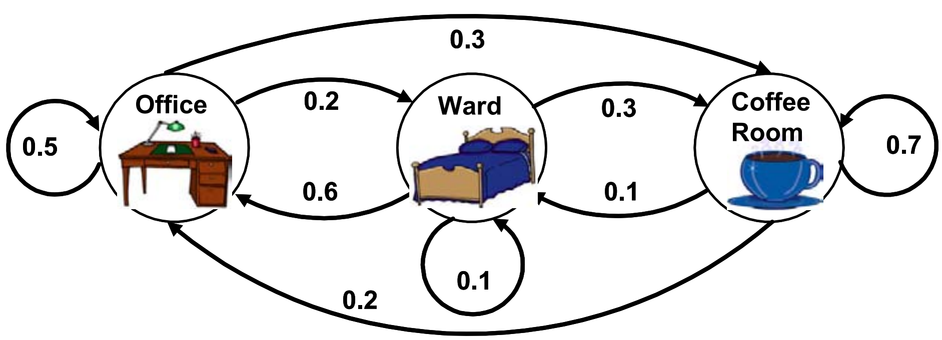


Figure 5: A Markov chain showing the probability that Hal (the robot) will enter various rooms

Hidden Markov Models generalize Markov Chains in that they allow several different paths through the model to produce the same output. Consequently, it is not possible to determine the state the model is in simply by observing the output (it is "hidden"). An HMM defines several possible output variables for each state, and the selection of an output at a given state is stochastic.

Markov models observe the Markov assumption, which states that the probability of the next state is dependent *only* upon the previous state. This assumption seems limiting, however efficient algorithms have been developed that perform remarkably well on problems similar to that described in this dissertation. Hidden Markov Models allow us to ask questions such as, "How well does a new (test) sequence match a given model?", or, "How can we optimize a model's parameters to best describe a given observation (training) sequence?" (Rabiner, 1989). Answering the first question involves computing the most likely path through the model for a given output sequence; this can be efficiently computed by the Viterbi (1967) algorithm. Answering the second question requires training an HMM given sets of example data. This involves estimating the (initially guessed) parameters of an arbitrary model repetitively, until the most likely parameters for the training examples are discovered. In the remainder of this section, I introduce a few of the formal concepts involved in answering the first question, and illustrate these concepts with a simple example. The level of detail provided here should be sufficient for understanding the analysis in chapter 7. For further information on HMMs, see Rabiner (1989) or Charniak (1993).

The state transition matrix for an HMM is denoted by A = $\{a_{ij}\}$, where

$$a_{ij} = \Pr[q_{t+1} = S_j \mid q_t = S_i]$$

The variable q_t denotes the state at time t. The equation above explains that the transition matrix describes the probabilities of the states q_t in the model, given that the previous state was q_{t-1}. The transition matrix for the Hal example, described earlier in this section, is shown in Figure 6.

	q_{t+1}		
q_t	**Office**	**Ward**	**Coffee Room**
Office	0.5	0.2	0.3
Ward	0.6	0.1	0.3
Coffee Room	0.2	0.1	0.7

Figure 6: State transition matrix, $\{a_{ij}\}$, for weather example

Let O define an observation sequence. For example, O = {Ward, Office, Office, Coffee Room}. An HMM differs from a Markov chain in that we might observe several different events in a particular state. For example, we might be chatting in the coffee room, and observe that Hal has re-entered the room but has changed the message displayed on his chest. The observation symbol probability distribution in state *j* is given by $B = \{b_j(k)\}$, where

$$b_j(k) = \Pr[v_k \text{ at } t \mid q_t = S_i]$$

and *V* describes the set of all possible observation symbols. The observation symbol probability distribution describes the probabilities of each of the observation symbols, v_k, for each of the states, *q*, at each time *t*.

If we let π describe the initial state distribution, $\pi = \Pr[q_1 = S_i]$, then an HMM can be fully described as

$$\lambda = (A, B, \pi)$$

We are now ready to look at the forward-backward procedure for finding the likelihood of a given observation sequence, given an HMM. This likelihood is denoted $\Pr(O \mid \lambda)$. Let $\alpha_t(i) = \Pr(O_1, O_2, ..., O_t, q_t = S_i \mid \lambda)$. The variable, $\alpha_t(i)$, is called the forward variable, and describes the probability of a partial observation sequence (up until time *t*), given model λ. In the first step, we initialize $\alpha_t(i)$: $\alpha_t(i) = \pi_i b_i(O_1)$. This initializes the forward variable as the joint probability of state S_i and the initial observation O_1. The second step is given by the following equation, in which *N* denotes the number of states in the HMM:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1})$$

The sum, $\sum_{i=1}^{N} \alpha_t(i) a_{ij}$, describes the probability of the joint event in which $O_1, O_2, ..., O_t$ are observed, the state at time *t* is S_i, and the state S_j is reached at time *t+1*. In other words, it is the probability of being in state S_j at time *t+1* , accounting for all the accompanying previous partial observations. Then $\alpha_{t+1}(j)$ can be determined by multiplying this value by $b_j(O_{t+1})$.

To find $\Pr(O \mid \lambda)$, we need only take the sum over the terminal values:

$$\Pr(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Let's take a look at how these equations can be used to determine the probability of an observation sequence, given an HMM. Suppose that our robot, Hal, is wandering through the hospital, delivering patient records to the doctors, and flowers to the patients. To simplify matters, we will assume that Hal only wanders between doctors' offices and hospital wards, and our HMM describes the probability that Hal will encounter a doctor or a patient at any given time. Our HMM, illustrated in Figure 7, shows that the probability of Hal being in an office, and seeing a patient there is 0.2. The probability of Hal being in an office, and seeing a doctor is 0.4. The probability of Hal wandering from an office to a ward, and seeing a patient in the ward is 0.1. The probability of wandering from an office to a ward, and seeing a doctor there is 0.3, and so on. Now, suppose that we know who Hal has seen, and we would like to know where he has been.

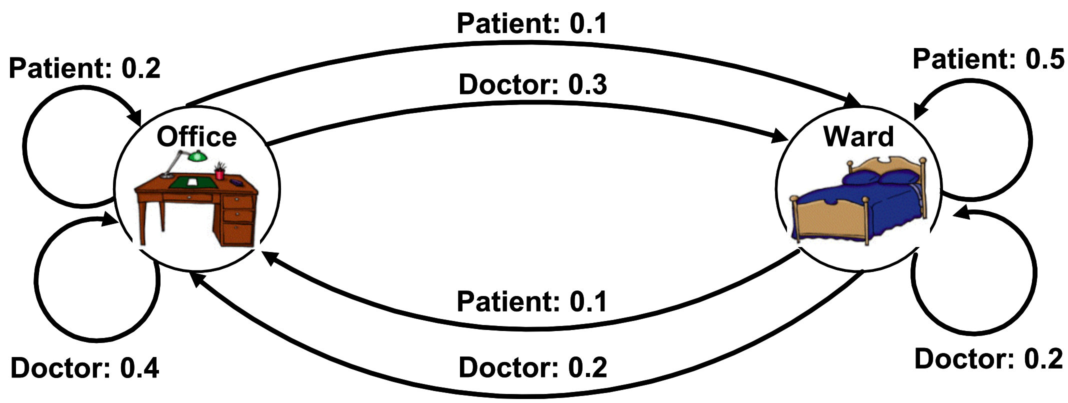


Figure 7: An example HMM showing a scenario for a hypothetical robot

Consider the observation sequence is O = {patient$_1$, patient$_2$}, stating that Hal has seen two patients, but no doctors. Figure 8 shows the possible paths we could take through the HMM, and the probabilities of these paths. The symbols on the arrows describe Hal's observations, and the nodes show the possible states (locations). We will assume that Hal starts out, at $t = 0$, in an office, and his first observation is the empty string, ϵ.

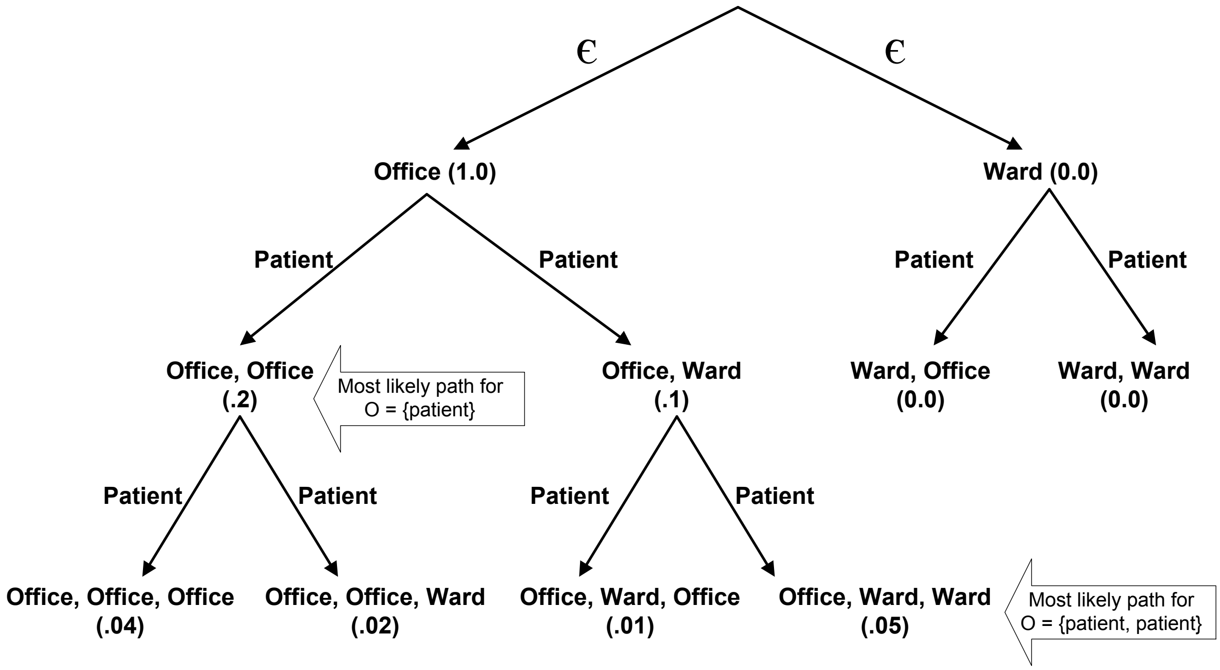


Figure 8: Possible paths through the HMM shown in Figure 7, and their probabilities (adapted from Charniak, 1993)

At $t = 1$, Hal sees a patient, and the HMM considers the first input, O_1 = {patient$_1$}. The most likely sequence of states that would produce this observation is Office$_0$, Office$_1$. The probability of this path is 0.2 (given directly by the model shown in Figure 7). We then receive a message that Hal has just seen another patient. If we were to continue with the assumption that Hal saw the first patient in an office, we would find a less than optimal state sequence. The probability of Hal seeing a patient in an office, and then another patient in an office is (0.2)(0.2) = 0.04. Figure 8 shows that, with this additional information, the state sequence, Office$_0$, Ward$_1$, Ward$_2$, is more likely the case: (0.1)(0.5) = 0.05.

This example shows why it may not be possible to determine the state of an HMM, given its observations. Note also that HMMs can easily handle observation sequences of indefinite lengths, although longer sequences will have smaller probabilities.

The iterative training and testing algorithms for HMMs do not perform an exhaustive search of the state space. Such a search, for V possible observation symbols, N states, and a test sequence of length T, would require $O(VT*N^T)$ calculations. In the analysis done for this dissertation, V = 112, N = 5, and T ≈ 15. An exhaustive search would require about $112*15*5^{15}$ = 5.12×10^{13} calculations! HMMs improve upon this by storing the best path for each state, at each iteration. As we saw from the Hal example, storing the overall best state is not sufficient for optimal performance.

5.5 Summary of Claims

The long-term goal of this research is to support groups of students who are learning at a distance over a computer network. In section 5.1, this long-term goal was realized as a hypothetical proof of concept system, in which an online computer-based coaching agent might dynamically mediate situations in which new knowledge is not effectively assimilated by the group. This coaching agent should be driven by an analysis engine that can (1) recognize when students are having trouble learning the new concepts they share with each other, and (2) determine why they are having trouble. This section explains how this dissertation attempts to address both of these issues, and summarizes my contributions to the development of the proof of concept system.

The way in which students interact with their peers and the computer artifacts provides clues about whether or not they are having trouble sharing and learning new knowledge. Studying this interaction over time, therefore, should provide important insight into the group process. We saw in section 5.3 how sequences of student interaction are scattered with noise and interruptions, and performing a statistical analysis of events found in these sequences does not reliably differentiate between effective and ineffective knowledge sharing. We also saw the importance of preserving and attending to the progression of events in the sequences.

Modeling the somewhat unpredictable and noisy nature of human interaction calls for a stochastic method. And, modeling the sequential nature of conversation calls for a state-based, or similar sequential analysis method. In previous work, the stochastic, state-based Hidden Markov Modeling approach performed well (achieving over 92% accuracy) when tasked to assess the effectiveness of sequences of knowledge sharing interaction (Soller & Lesgold, in press; Soller, Wiebe, & Lesgold, 2002). This analysis was done using fourteen knowledge sharing sequences obtained from five groups. Although these results were encouraging, the dataset was too small to confirm the results. This dissertation attempted to confirm my previous results by testing Claim 1 (stated below).

CLAIM 1: *A Hidden Markov Modeling approach can be used to distinguish between effective and ineffective student knowledge sharing interaction.*

Although this claim does not preclude other methods from performing well at this classification task, the preliminary analysis (described in section 5.3) showed that some methods that do not account for sequential and stochastic nature of the data perform less well than the HMM approach. To test Claim 1, I have analyzed data from 7 additional groups of three students each, run using the same experimental method as the first five groups. The results of this analysis are described in chapter **7**.

Previous research, which looked at groups of two and three, showed that student interaction may be diagnosed by analyzing sequences of simple speech acts (e.g. "OK", "I Agree") and student actions (e.g. Muehlenbrock, 2001; Constantino-Gonzalez & Suthers, 2001; also see chapter 4). This previous research, however, did not attempt to computationally understand or diagnose sequences of coded group conversation. By testing Claim 1, I will also, in effect, be attempting to confirm that online student interaction can be assessed by analyzing sequences of conversational acts and student actions.

CLAIM 2: *Online knowledge sharing behavior can be assessed by analyzing sequences of conversational acts and student actions (on a shared workspace), with limited consideration of the dialog content.*

If the HMM approach proves viable, then I will be able to claim that segmented online knowledge sharing behavior can be assessed without the use of complex and error-prone natural language understanding technology. This claim would strengthen the argument that collaborative learning, in general, might be assessed and supported with only limited consideration of the dialog content. Some level of natural language understanding, however, would probably be needed to perform the dialog segmentation, as described in section 5.1.

Understanding *why* a knowledge sharing episode is ineffective is critical to selecting a proper mediation strategy. For example, a student sharing new knowledge with his teammate may have trouble formulating sufficiently elaborated explanations, and may need help in using analogies or multiple representations. Or, a knowledge receiver may need encouragement to speak up and articulate why he does not understand a new knowledge element.

In the second phase of this dissertation work, I attempted to develop generalized models of knowledge sharing. These models represent both effective knowledge sharing, and breakdowns in knowledge sharing. For example, models of breakdowns include scenarios in which new knowledge is not effectively conveyed by the sharer, and models in which new knowledge is not effectively assimilated by the receivers. From a system's point of view, it is necessary to determine why students are having trouble in order to recommend strategies for supporting the process of knowledge sharing during collaborative learning activities.

In attempting to model the various ways in which group members may fail or succeed at effectively assimilating new information, this research considered an HMM clustering approach (Juang & Rabiner, 1985; Smyth, 1997). In this approach, one HMM is trained for each knowledge sharing sequence, S_i, where $1 \leq i \leq N$, and the N HMMs are clustered, using a log-likelihood algorithm, into K groups. Then, one HMM is fit to each cluster.

CLAIM 3: *A Hidden Markov Modeling clustering approach can help us discover and automatically classify the various ways group members are successful or unsuccessful in sharing new knowledge with each other*

The HMM clustering approach was attractive because, for the unsuccessful interaction sequences, each of the K HMMs that the system learns should describe a different way in which group members may have trouble sharing new knowledge with each other. For example, one HMM might represent a situation in which the knowledge sharer does not explain the new concept well, and another might represent a situation in which the knowledge receivers do not understand the new knowledge and do not ask for clarification. If Claim 3 is confirmed, the knowledge sharing analyzer (Figure 1) need only find which of the K HMMs best match any new sequence of knowledge sharing interaction, in order to determine why students are having trouble. This information would enable the instructional module and online knowledge sharing coach (section 5.1) to make an informed decision about what strategy might best support the group knowledge sharing process.

Likewise, for the successful interaction sequences, each of the *K* HMMS that the system learns should describe a different way in which the group members successfully share new knowledge with each other. A more detailed description of the HMM clustering approach is included in chapter 7. This approach, if successful, should help to explain what students are doing when they succeed or fail. Understanding this is key to the development and selection of appropriate strategies to support those students who may experience knowledge sharing breakdowns.

6 Experimental Method

Over the course of three years (from June 2000 to April 2002), a series of experiments was run at both the University of Pittsburgh in Pittsburgh, Pennsylvania, and the MITRE Corporation in Bedford, Massachusetts. During these experiments, 12 groups of 3 participants each were asked to solve one Object-Oriented Analysis and Design problem using the specialized shared OMT tool shown in the top part of Figure 2, while communicating through the structured, sentence opener interface shown in the bottom part of the same figure. This chapter discusses the experimental procedure, designed specifically to obtain examples of knowledge sharing behavior, and the data pre-processing that was performed to prepare the data for the Hidden Markov Modeling procedure, described in the next chapter.

6.1 Subjects

A total of 36 students (12 groups of 3 students each) participated in the study. Each participant was assigned to a group of three based on his or her availability. Eight groups were run at LRDC, and four groups were run at the MITRE Corporation in Bedford, MA. Two of the four MITRE groups were comprised of summer student interns, and two were comprised of MITRE technical staff members. All of the subjects (except the MITRE staff members) were undergraduates or first-year graduate students majoring in the physical sciences or engineering, none of whom had expertise in Object Modeling Technique. The subjects participating at LRDC received pizza halfway through the four hour study, and were paid at the completion of the study. The experiments at MITRE were split into 2 two hour parts, run over the course of a week, and the subjects were compensated for their time. All twelve groups were run using the same protocol, given the same information, and asked to do the same exercise. One of the LRDC groups was omitted from the study because the group members did not discuss any of the knowledge elements during the problem solving session. The remainder of this dissertation refers to the remaining 11 groups (33 students).

6.2 Experimental Procedure.

The subjects in each group were first asked to introduce themselves to their teammates by answering a few personal questions. Each experiment (Refer to Table 1) began with a half hour interactive lecture on OMT basic concepts and notation, during which the subjects practiced solving a realistic problem. The subjects then participated in a half hour hands-on software tutorial. During the tutorial, the subjects were introduced to all 36 sentence openers on the interface (see Appendix A). The subjects were then assigned to separate rooms, received their individual knowledge elements, and took a pre-test.

Table 1: Sample research protocol

Elapsed Time	Step in Experiment
0:00	Sign Consent form
0:15	Participants introduce themselves: *Questions:* *Name, College & Major, Hometown* *Favorite undergraduate class* *Favorite pizza toppings* *Favorite wintertime activity* *Best way to get a date for this weekend*
0:25	Introductory lesson
0:50	Tool tutorial
1:20	Pizza Break (LRDC subjects)
1:40	Participants are assigned to separate rooms
1:45	Pretest & Review
2:10	MITRE Subjects: End of Session I and Beginning of Session II (with a brief review)
2:10	Exercise
3:35	Break (after completing exercise)
3:40	Posttest
3:50	Questionnaire
4:00	Participants are paid & sign receipt form

The individual knowledge elements were all different conceptual elements that addressed key OMT concepts (for example, “Attributes common to a group of subclasses should be attached to the superclass because the subclasses inherit the features of the superclass”). Each knowledge element was explained on a separate sheet of paper with a worked-out example (see Appendix B). The pre-test (see Appendix C) included one problem for each of the three knowledge elements. It was expected that the student given knowledge element #1 would get only pre-test question #1 right, the student given knowledge element #2 would get only pre-test question #2 right, and likewise for the third student. To ensure that each student understood his or her unique knowledge element, an experimenter reviewed the pre-test problem pertaining to the student's knowledge element before the group began the main exercise. Students who missed the pre-test problem on their knowledge element were asked to reread their knowledge element sheet and rework the missed pre-test problem, while explaining their work out loud (Chi et al., 1989).

The subjects were not told specifically that they hold different knowledge elements, however they were reminded that their teammates may have different backgrounds and knowledge, and that sharing and explaining ideas, and listening to others' ideas is important in group learning. For the LRDC students, it was also possible to offer an external group incentive to encourage the students to do their best. The students were told that if their group solution satisfied the requirements of the exercise better than the other groups, they would each receive an extra cash bonus[6]. The group exercise is shown in Appendix D. All groups completed the exercise on-line within about an hour and fifteen minutes.

During the on-line session, the software automatically logged the students' conversation and actions (see Appendix E). After the problem solving session, the subjects completed a post-test, and filled out a questionnaire (Appendix F). The post-test, like the pre-test, addressed the three knowledge elements. It was expected that the members of effective knowledge sharing groups would perform well on all post-test questions.

6.3 Data Collection and Pre-processing

The groups showed both instances of effective knowledge sharing and instances of ineffective knowledge sharing. Recall that in order for a knowledge element to be considered as effectively shared, three requirements must be satisfied: (1) the individual sharing the new knowledge (the "sharer") must show that she understands it by correctly answering the corresponding pre and post test questions, (2) the concept must come up during the conversation, and (3) at least one group member who did not know the concept before the collaborative session started (as shown by his pre-test) must show that he learned it during the session by correctly answering the corresponding post-test question (F. Linton, personal communication, May 8, 2001).

Because there were 33 subjects, there were a maximum of 66 possible opportunities for effective knowledge sharing: 2 opportunities for each student to learn the other 2 students' elements. The students took advantage of 23 of these opportunities (i.e. they met all 3 criteria). Three students did not meet criterion (1), eliminating 6 opportunities. In 12 of the cases, the student already knew the knowledge element (as given by the pre-tests), so there was no opportunity for learning in those cases. The experimental results described in the next chapter will help to explain why the students did not take advantage of the remaining 25 opportunities. The remainder of this chapter will present the sequences of knowledge sharing interaction that were logged while students were learning collaboratively.

The student action logs (e.g. Figure 4) from the 11 experiments were parsed by hand to extract the segments in which the students shared their unique knowledge elements. A total of 29 of these *knowledge sharing episodes* were manually identified and classified. In the future, it should be possible to automate the parsing and identification of these episodes. Appendix G gives an example of how one of the logs was parsed in order to identify and extract a knowledge sharing episode. The procedure was based on identifying the main topic of conversation by considering both the student dialog and workspace actions. Only four topics of conversation were possible: one of the three knowledge elements, or anything else (the catch all category). Once a conversation topic is determined, the next topic is assumed to be the same as the previous topic, unless a change is detected. Dr. Frank Linton has had some success in using this procedure (F. Linton, personal communication, January 24, 2002).

The portion of the log shown in Appendix G begins with Alex, Sheldon, and Jim discussing whether or not there should be a link between the object classes *car loan* and *car*. This topic of discussion closes when Sheldon suggests that they be linked indirectly, through some of the other classes. He says, "I think car is linked to car loan through company/bank/person." Jim then suggests that they move on to the next topic of discussion, and recommends discussing the discriminator, even though Alex seems a little confused about the previous discussion. The concept of a discriminator, and how to apply one to an object model was Sheldon's knowledge element. It is not unusual that Jim was the student who suggested the discriminator because the exercise hinted that a discriminator might be useful. Up until Jim suggests that the group discuss the discriminator, the topic of discussion for this dialog falls into the catch all category. The mention of the keyword, discriminator, and the following progression of dialog on this topic, singles out this episode as a knowledge sharing episode. The episode ends

[6] The MITRE subjects were not offered a group incentive because they were not paid for their participation.

when Jim changes the discussion topic by saying, “Are you ready to finish?” and then Sheldon clicks on the next agenda item, “Verify Diagram”.

Because automatic segmentation procedures rely on natural language understanding technology, they are error-prone, and may not parse the student action logs the way that they were parsed manually for this research. If we assume that the dialog segmenter can produce results close to that of the human parser, a more important question, then, concerns the degree to which the knowledge sharing analyzer is sensitive to the particular boundaries of the 29 manually identified knowledge sharing episodes. In Chapter 7, I show that changing the boundaries of the knowledge sharing episodes slightly does not affect the knowledge sharing analyzer’s performance.

A sequence was considered effective if at least one of the students receiving the new knowledge did not know it before the session (as shown by his pre-test) and demonstrated that he learned it during the session (as shown by his post-test). Recall from section 6.2 in this chapter, that the pre and post tests directly targeted the three knowledge elements that the students were expected to share during the group problem solving session. A sequence was considered ineffective if a knowledge element was discussed during the episode, but none of the receiving students demonstrated mastery of the concept on the post test.

The extracted episode from Appendix G (which is also LRDC 15 in Appendix E) was coded as ineffective because neither Jim nor Alex answered the post test question on discriminators correctly. Chapter 7 will take a closer look at why some students learn their peers’ knowledge elements, and others do not. The qualitative analysis in that chapter is driven by the computational analysis – the focus of this dissertation.

As shown in Table 2, 10 of the extracted knowledge sharing episodes were classified as effective (given the pre and post test scores) and 19 were determined to be ineffective. These sequences do not directly correspond to the 52 opportunities mentioned earlier in this section, because one episode may result in 2 students learning, or one student may learn across several episodes.

Table 2: Number of Knowledge Sharing Episodes obtained

	Effective	**Ineffective**	**Total**
LRDC	7	11	18
MITRE	3	8	11
	10	**19**	29

The 29 knowledge sharing episodes varied in length from 5 to 49 contributions, and contained both conversational elements and action events. The top part of Figure 9 shows an example of one such episode. The sentence openers, which indicate the system-coded subskills and attributes, are italicized. The bottom part of Figure 9 shows an example of an actual sequence, based on the episode, that was used to train HMMs to analyze and classify new instances of knowledge sharing (described in the next chapter). Appendix E includes all 29 coded knowledge sharing episodes.

Student	Subskill	Attribute	Actual Contribution (Not seen by HMM)
A	Request	Opinion	*Do you think* we need a discriminator for the car owner-ship
C	Discuss	Doubt	*I'm not so sure*
B	Request	Elaboration	*Can you tell me more* about what a discriminator is
C	Discuss	Agree	*Yes, I agree* because I myself am not so sure as to what its function is
A	Inform	Explain/Clarify	*Let me explain it this way* - A car can be owned by a person , a company or a bank. I think ownership type is the discrinator.
A	Maintenance	Apologize	*Sorry* I mean discriminator.

Actual HMM Training Sequence
A-Request-Opinion C-Discuss-Doubt B-Request-Elaboration C-Discuss-Agree A-Inform-Explain A-Maintenance-Apologize Sequence-Termination

Figure 9: An actual logged knowledge sharing episode (above), showing system coded subskills and attributes, and its corresponding HMM training sequence (below)

In a preliminary analysis, a prototype knowledge sharing analyzer was tasked to select one of the three participants as the knowledge sharer. This analysis was performed because, if successful, it would allow the system to assign a special set of tags to the contributions of the knowledge sharer. In Figure 9, for example, the tags reserved for the knowledge sharer's contributions begin with the code "A-". Differentiating the roles of the knowledge sharer and recipients was thought to facilitate the system's assessment of the episode's effectiveness. Because Hidden Markov Models are designed to output the probability that a particular sequence matches a trained model, the knowledge sharer's contributions within each of the training episodes were given a reserved set of labels, and these episodes were used to train one HMM. The knowledge sharer role was therefore held consistent throughout the training data, and each test set was reproduced twice such that three test sequences were obtained, each featuring a different participant playing the role of knowledge sharer. Figure 10 shows schematically how this was done using the sequence from Figure 9.

For this analysis, I used a 14-fold cross validation approach, in which I trained and tested data from the first 5 groups (14 knowledge sharing sequences). The cross validation approach required that each of the 14 test sets be tested against the other 13 (training) sets. This involved determining the likelihood of each of the three test sequences in each test set, given an HMM trained using the other 13 training sets. Only the sequences that correctly indicated the knowledge sharer's role were included in the training sets.

Given the choice of three possible knowledge sharers, the five node HMMs chose the correct student as knowledge sharer for all 14 experiments, achieving a 100% accuracy. The baseline comparison is chance, or 33.3%, since there is a 1/3 chance of arbitrarily choosing the right student as knowledge sharer. The next best comparison is to count the number of *Inform* conversation acts each participant uses during the knowledge sharing episode, and select the student with the highest number in each test set. This strategy produces a 64.3% accuracy. Because of the success of this preliminary analysis, the analysis that follows adopts a set of specialized tags (described later in this chapter) for the knowledge sharer.

Some of the extracted sequences included actions that students took on the workspace. These actions were matched to a list of predetermined "productive" actions – those that were expected to lead students to a model solution. The model solution is intended to represent the successful application of all three knowledge elements. Productive actions in the extracted sequences were given special labels, and included in the sequence with the name of the student who took the action (e.g. A-KE-Action). Table 3 shows an example of the actions a student might take that lead to the completion of Knowledge Element #1: "Moving attributes common to subclasses into the superclass".

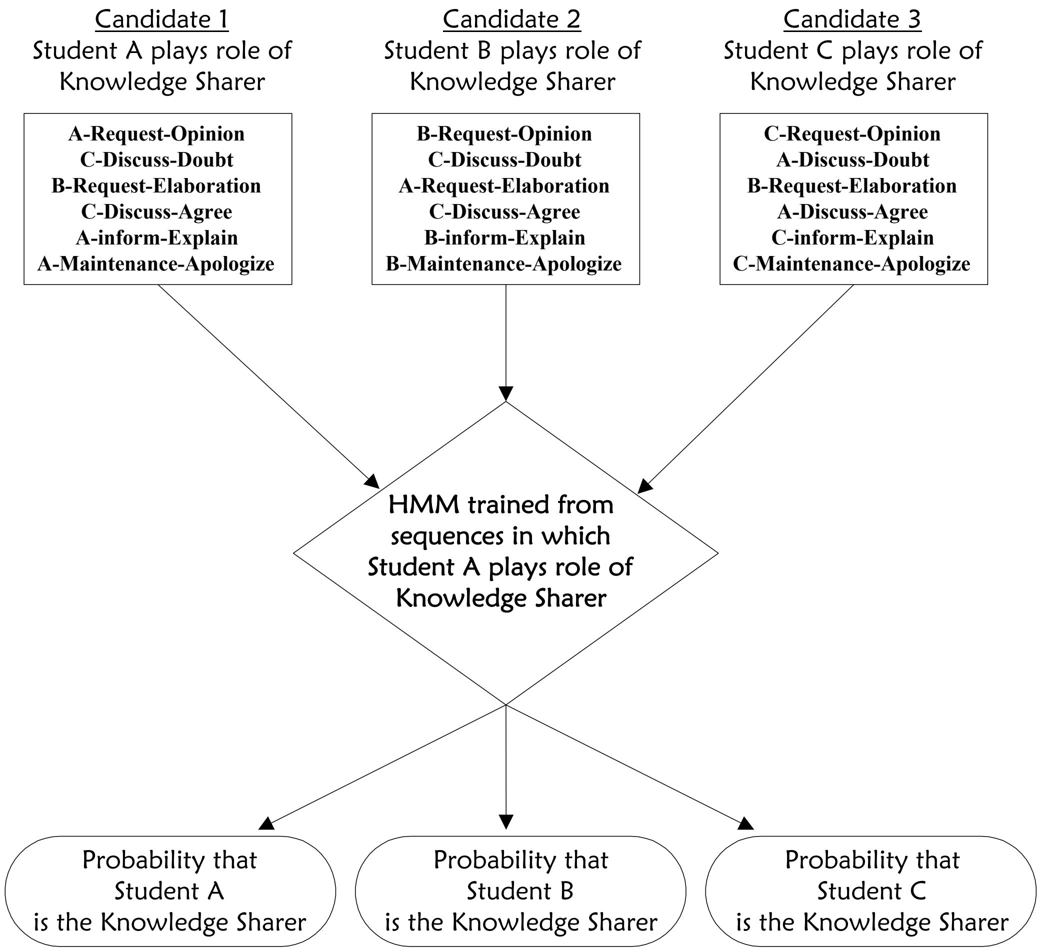


Figure 10: Schematic showing procedure for determining the Knowledge Sharer role

The knowledge sharing sequences were coded using a numerical scheme before they were used to train the HMMs. The numerical scheme consists of 36 codes for each of the 36 possible sentence openers, plus one for when a student takes a KE-Action on the workspace. The student sharing his or her knowledge element (student A) was assigned codes 1 – 37, and the other two students were arbitrarily assigned codes 38 – 74 (student B), and 75 – 111 (student C). Codes 37, 74, and 111 were reserved for situations in which students A, B, or C, respectively, took a KE-

Action (i.e. Table 3). Code 112 was reserved as the sequence terminator. Each coded episode was replicated with actors B and C swapped so that the analysis would not reflect idiosyncrasies in the random determination of participants B and C. This resulted in a total of 58 episodes (or 29 pairs of episodes).

Table 3: An example of the productive workspace actions that lead to the completion of a knowledge element

KE-Actions for Knowledge Element #1	Log Entry Corresponding to KE-Action
Delete Attribute *Name* in Class *Person*	Edited attributes for class $Person$ - $<+address +age -Name>$
Delete Attribute *Name* in Class *Company*	Edited attributes for class $Company$ - $<-Name>$
Delete Attribute *Name* in Class *Bank*	Edited attributes for class $Bank$ - $<-Name>$
Add Attribute *Name* to Class *Owner*	Edited attributes for class $Owner$ - $<+Name>$

After the knowledge sharing analyzer showed that it could successfully identify the knowledge sharer for each episode (described earlier in this section), it was challenged to assess the effectiveness of the episode. A preliminary analysis was performed using the fourteen sequences of knowledge sharing interaction from the first five LRDC groups, run in the Spring of 2000. This preliminary analysis showed that the Hidden Markov Modeling approach performed well (achieving over 92% accuracy) when tasked to assess the effectiveness of these sequences. Assessing the effectiveness of the sequences involved classifying them as effective or ineffective (50% baseline). The encouraging results of this preliminary study motivated the collection of data from 7 more groups, and a re-analysis and closer look at the entire dataset. The next chapter describes the full analysis in detail.

7 Results of Evaluating Knowledge Sharing Conversation

This chapter presents the results of the computational analysis outlined in chapter **5**. This analysis is intended to test Claims 1, 2, and 3, repeated below for your convenience. The data analyzed here was obtained through the procedure described in chapter **6**.

CLAIM 1: *A Hidden Markov Modeling approach can be used to distinguish between effective and ineffective student knowledge sharing interaction.*

CLAIM 2: *Online knowledge sharing behavior can be assessed by analyzing sequences of conversational acts and student actions (on a shared workspace), with limited consideration of the dialog content.*

CLAIM 3: *A Hidden Markov Modeling clustering approach can help us discover and automatically classify the various ways group members are successful or unsuccessful in sharing new knowledge with each other*

This chapter is organized as follows. First, the student responses to the questionnaires are presented. Second, I present the results of applying Hidden Markov Models to classify the knowledge sharing episodes obtained from the transcripts. Third, the episodes are analyzed using a combination of HMM clustering, multidimensional scaling, and qualitative analysis.

7.1 Questionnaire Results

This section discusses the students' responses to the questionnaires, and compares them to the results of the pre and post tests. The results of the questionnaires are shown in Figure 11. In general, the participants felt a high degree of engagement in the task, and felt they did learn OMT during the four hour study. A statistical analysis, however, showed that actual individual learning (as measured by students' pre and post tests) was not correlated with perceived learning (as measured by the questionnaires). The pre and post tests showed that the subjects learned an average of 0.65 Knowledge Elements during the study, and the groups learned an average of 1.96 Knowledge Elements. Perceived learning, but not actual learning, was positively influenced by the students' degree of engagement in the task, and their feeling of control during the study (r = 0.51 and r = 0.52 respectively). As an observer, I noted that most students were very engaged during the group exercise portion of the study, and genuinely enjoyed the experience.

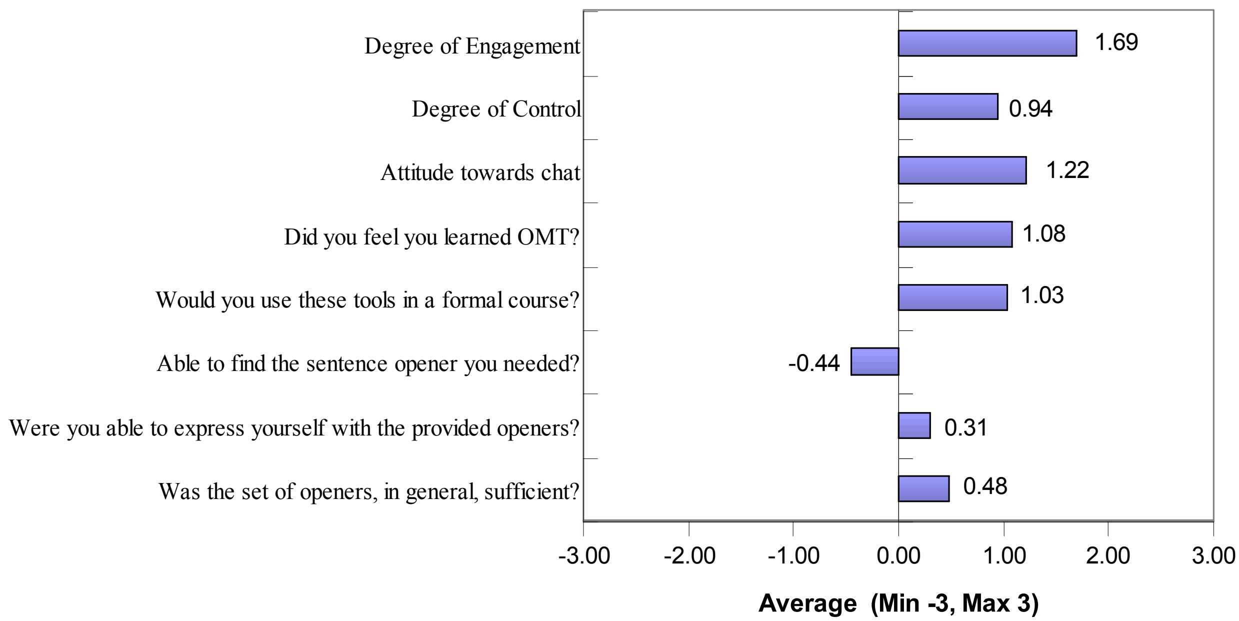


Figure 11: Averages of questionnaire responses

Although the questionnaire responses overall were positive, they were fairly neutral with respect to the sentence openers[7]. This was expected because at this stage of software development, the students did not receive feedback and guidance in return for their efforts to communicate in a restricted language. One factor that did seem to impact students' ability to find the sentence opener they needed was their attitude towards chat tools in general (r = 0.52). Although students reported that the sentence opener tutorial was helpful, many still spent time during the problem solving session reading through the list of sentence openers to find the one that was most appropriate for their contribution.

[7] The average from last question was 3.48 on a [0 6] scale, but was converted to a [-3 3] scale for purposes of visual comparison.

7.2 Assessment of Knowledge Sharing Effectiveness using HMMs

For this analysis, two 5 state Hidden Markov Models were trained[8]. The first was trained using only sequences of effective knowledge sharing interaction (I call this the effective HMM), and the second using only sequences of ineffective knowledge sharing (the ineffective HMM). Testing the models involved running a new knowledge sharing sequence – one that is not used for training – through both models. The output from the effective HMM described the probability that the new test sequence was effective (as defined by the training examples), and the output from the ineffective HMM described the probability that the test sequence is ineffective. The test sequence was then classified as effective if it had a higher path probability through the effective HMM, or ineffective if its path probability through the ineffective HMM was higher. Since the probabilities in these models can be quite small, it is common to take the log of the path probability, which results in a negative number. The largest path probability is then given by the smallest absolute value. Procedures similar to this have been used successfully in other domains, such as gesture recognition (Yang, Xu, & Chen, 1997), and the classification of international events (Schrodt, 1999).

It is not necessarily intuitive that two probabilities, generated by models trained from different data sets, are comparable or even indicative of the effectiveness of a test sequence. The procedure discussed in chapter 5 described how to obtain $\Pr(S \mid \lambda)$ – the probability of a test sequence given an HMM. If we would like to test the effectiveness of a sequence, we need to compare $\Pr(S \mid \lambda_{eff})$ to $\Pr(S \mid \lambda_{ineff})$. As long as the models are initially seeded using the same constraints, we can obtain the same result by comparing $\Pr(\lambda_{eff} \mid S)$ to $\Pr(\lambda_{ineff} \mid S)$. Formally, we can compute $\Pr(\lambda \mid S)$ by Bayes Rule:

$$\Pr(\lambda \mid S) = \frac{\Pr(S \mid \lambda)\Pr(\lambda)}{\Pr(S)}.$$

In comparing $\Pr(\lambda_{eff} \mid S)$ to $\Pr(\lambda_{ineff} \mid S)$, the probability of the test sequence, Pr(*S*), is a constant because the same test sequence is run through both models; therefore, it can be eliminated. This leaves us with the comparison: $[\Pr(S \mid \lambda_{eff})\Pr(\lambda_{eff})] : [\Pr(S \mid \lambda_{ineff})\Pr(\lambda_{ineff})]$.

[8] Before choosing the 5 node HMM, I experimented with 3 and 4 node HMMs, obtaining similar (but not optimal) results. Performance seemed to decline with 7 or more states. Similar observations were made for the HMMs described earlier.

Because the models λ_{eff} and λ_{ineff} are also constants across all of the test cases, and do not differ significantly (p = 0.65), they too can be eliminated, leaving us with $\Pr(\lambda \mid S) = \Pr(S \mid \lambda)$. The p statistic, obtained through a Kolmogorov-Smirnov test, tells us that the distributions of transition probabilities in the two models do not differ significantly (Fisher & van Belle, 1993). Since the HMMs remain constant for all of the test cases, it is reasonable to perform relative comparisons between $\Pr(\lambda_{eff} \mid S)$ and $\Pr(\lambda_{ineff} \mid S)$, although the absolute magnitudes of the differences between the models may not be significant. In summary, it may be more computationally intuitive to think of the analysis that follows as a process of comparing two HMMs – one effective and one ineffective, and determining which model best matches a given test sequence. But because this is essentially the same as the more intuitive and conventional terminology in which we calculate the likelihood of a sequence, given a model, this dissertation has adopted the latter form.

Of the 29 knowledge sharing sequences identified, 10 were found to be effective and 19 were found to be ineffective. As described in chapter 6, each coded episode was replicated with actors B and C swapped so that the analysis would not reflect idiosyncrasies in the random determination of participants B and C. This resulted in a total of 58 episodes (or 29 pairs of episodes). Because of the small dataset, I used a modified 58-fold cross-validation approach, in which each test sequence and its B-C swapped pair was removed from the training set and tested against the two HMMs (representing effective and ineffective interaction) which were trained using the other 56 episodes.

Table 4 shows the path probabilities of each test sequence through both the effective and ineffective HMMs. Columns 2 and 3 of the table show the logs of the Viterbi path probabilities (Rabiner, 1989). This value is highly dependent on the length of the test sequence (longer sequences will produce smaller probabilities), and so will vary for each sequence. The first half of the table shows the results for the data before actors B and C were swapped. For this data, notice that the path probabilities for 8 of the 10 effective test sequences (labeled L1 through L7, and M1 through M3) were higher through the effective HMM, and the path probabilities for 13 of the 19 ineffective test sequences (labeled L8 through M11) were higher through the ineffective HMM. The first letter of the labels correspond to the location where the data was obtained ("L" for LRDC, or "M" for MITRE). These labels also directly correspond to those in APPENDIX E. The results obtained for the B-C swapped data was comparable to that shown in the first half of the table. The baseline comparison for this analysis is chance, or 50%, since there is a 1/2 chance of arbitrarily classifying a given test sequence as effective or ineffective. The HMM approach produced an accuracy of 74.14%, performing at almost 25% above the baseline.

Table 4: Path probabilities of each test sequence through both the effective and ineffective HMMs

	Test Set	Log(Prob) through Effective HMM	Log(Prob) through Ineffective HMM	Difference	Classified Correctly?
Effective	L1	-129.648	-156.453	26.805	+
	L2	-21.380	-44.409	23.029	+
	L3	-42.853	-55.364	12.510	+
	L4	-238.851	-249.835	10.984	+
	L5	-71.927	-88.491	16.563	+
	L6	-32.732	-34.388	1.655	+
	L7	-22.075	-24.445	2.370	+
	M1	-106.072	-129.389	23.317	+
	M2	-42.366	-41.145	-1.221	-
	M3	-77.336	-73.593	-3.743	-
Ineffective	L8	-34.463	-33.3785	-1.085	+
	L9	-32.710	-31.3626	-1.348	+
	L10	-36.626	-34.8753	-1.750	+
	L11	-49.515	-46.9704	-2.545	+
	L12	-26.065	-21.5107	-4.555	+
	L13	-32.208	-38.4808	6.273	-
	L14	-130.764	-148.63	17.866	-
	L15	-44.502	-50.6354	6.134	-
	L16	-27.217	-30.4924	3.276	-
	L17	-34.332	-26.2631	-8.069	+
	L18	-20.475	-18.6691	-1.806	+
	M4	-40.567	-44.5923	4.026	-
	M5	-17.440	-16.1282	-1.312	+
	M6	-27.552	-23.5049	-4.047	+
	M7	-46.039	-41.5058	-4.533	+
	M8	-18.379	-15.5094	-2.870	+
	M9	-36.888	-36.718	-0.170	+
	M10	-21.753	-21.9527	0.200	-
	M11	-26.715	-26.4327	-0.283	+

DATA WITH ACTORS B & C SWAPPED					
Effective	L1	-129.390	-157.318	27.928	+
	L2	-20.377	-44.139	23.762	+
	L3	-42.805	-55.307	12.502	+
	L4	-241.656	-249.926	8.270	+
	L5	-71.369	-87.779	16.410	+
	L6	-32.173	-34.141	1.968	+
	L7	-22.647	-24.138	1.491	+
	M1	-104.856	-129.084	24.228	+
	M2	-44.304	-41.646	-2.658	-
	M3	-76.340	-74.776	-1.565	-
Ineffective	L8	-34.533	-33.1999	-1.333	+
	L9	-32.524	-31.8998	-0.624	+
	L10	-36.797	-36.2113	-0.586	+
	L11	-49.523	-47.5637	-1.959	+
	L12	-25.817	-21.0018	-4.815	+
	L13	-32.430	-37.8814	5.452	-
	L14	-130.943	-147.8511	16.909	-
	L15	-44.960	-50.4655	5.505	-
	L16	-26.832	-30.9112	4.079	-
	L17	-33.971	-26.5918	-7.379	+
	L18	-20.319	-18.4904	-1.829	+
	M4	-40.160	-45.421	5.261	-
	M5	-17.402	-16.027	-1.375	+
	M6	-27.673	-22.1351	-5.538	+
	M7	-46.427	-41.8255	-4.602	+
	M8	-18.344	-15.8313	-2.513	+
	M9	-36.752	-35.7105	-1.041	+
	M10	-21.908	-21.2401	-0.668	+
	M11	-27.240	-26.075	-1.165	+
					Accuracy: 0.7413

This result represents a decrease in accuracy from the previously reported result. This decrease may have occurred when the data from the latter seven groups was added because, (1) the number of new interaction patterns that appeared in the latter seven groups (that were not present in the first five groups) may exceed the amount of data available to distinguish these new patterns, and (2) the algorithms used to train and test the HMMS from the first five groups (HMM Software Version 1.02, written by Tapas Kanungo, 1998) are slightly different from those used in the final analysis (MATLAB HMM Toolbox written by Kevin Murphy, 1998)[9].

[9] The MATLAB HMM Toolbox was used for the final analysis because the data pre-processing and post-processing algorithms were also written in MATLAB.

Recall from Chapter 5 that the knowledge sharing analyzer assumes the dialog is already segmented; it accepts only knowledge sharing episodes from the dialog segmenter as input. Although the dialogs for this research were segmented by hand, we might like to know that if a dialog segmentation component were plugged into our framework, that it would not alter the effectiveness of the knowledge sharing analyzer, even if it does not segment the dialog exactly as it was segmented here. In order to ensure that the knowledge sharing analyzer was not sensitive to the particular boundaries of the 29 manually identified knowledge sharing episodes, the HMM effective/ineffective classification procedure was rerun using knowledge sharing episodes with "fuzzy" boundaries. Both the upper and lower boundaries of every other numbered episode were moved up, or down, by one contribution, unless one of two special cases was recognized. For 12 of the 58 boundaries, deletion would have meant removing a contribution in which students mentioned keywords directly from the formal knowledge element descriptions. And, for 4 of the 58 boundaries, addition would have meant including a contribution that was already part of another episode. The HMM approach performed at 77.59% accuracy when trained and tested by these episodes with fuzzy boundaries. Based on this analysis, the knowledge sharing analyzer's performance seems to be stable, even if the dialog segmentation is not perfect.

The analysis in this section shows that Hidden Markov Models are useful for identifying when students are effectively sharing the new knowledge they bring to bear on the problem, and when they are experiencing knowledge sharing breakdowns. A system based on this analysis alone could offer support and guidance about 74% of the time the students need it, which is better than guessing when students are having trouble, or basing intervention on students' requests for help. The 25% error rate still means that the instructor or the computer-based coaching agent might miss the opportunity to offer advice to the group when it is needed. In this case, however, the data suggests that there is a good chance the system will pick up on the breakdown the next time it occurs.

The next step is determining why students may be having trouble so that appropriate facilitation methods can be identified and tested. The next section takes a closer look at the differences between the effective and ineffective sequences in order understand the qualitative differences. For example, we might expect to see more questioning and critical discussion in effective knowledge sharing episodes, and more acknowledgement in less effective episodes (Soller, 2001).

7.3 *Analysis of Knowledge Sharing Interaction using HMM Clustering and Multidimensional Scaling*

Understanding *why* a knowledge sharing episode is ineffective is critical to selecting a proper mediation strategy. For example, a student sharing new knowledge with his teammate may have trouble formulating sufficiently elaborated explanations, and may need help in using analogies or multiple representations. Or, a knowledge receiver may need encouragement to speak up and articulate why he does not understand a new knowledge element.

In the second phase of this dissertation work, I attempted to develop generalized models of effective knowledge sharing, and breakdowns in knowledge sharing. A system must be able to differentiate between these cases if it is to understand knowledge sharing interaction, and recommend strategies for supporting this process. In attempting to model the various ways in which group members may fail to effectively assimilate new information, I first applied an HMM clustering approach (Juang & Rabiner, 1985; Smyth, 1997).

In the first step of the HMM clustering approach, 20 effective, and 38 ineffective HMMs were trained (in the traditional manner) from each of the 10 paired effective, and 19 paired ineffective knowledge sharing sequences. Recall from the previous section that each sequence was replicated with actors B and C swapped, resulting in a B-C swapped pair.

Formally, each sequence, S_j, $1 \leq j \leq N$, was used to train one HMM, M_i, $1 \leq i \leq N$, $i = j$. For the effective case, $N_e = 20$, and for the ineffective case, $N_i = 38$. Then, the log-likelihood of each sequence, S_j, given each of the HMMs, M_i was calculated via the standard HMM testing procedure. This resulted in 2 matrices, one describing the likelihoods of the effective sequences given the effective models, $loglik_e(S_j \mid M_i)$, and one describing the likelihoods of the ineffective sequences given the ineffective models, $loglik_i(S_j \mid M_i)$. Figure 12 shows the format of these matrices.

By transposing a matrix such as the one in Figure 12, and comparing the row vectors, one can compare the similarities of the HMMs. Each row vector of the transposed matrix describes the likelihood of each of the sequences given a particular model, M_i ; hence similar models should produce similar likelihood vectors. Given this observation, it would make sense to cluster these row vectors to see which models were most similar. Clustering, in the traditional sense, means calculating the distance (or similarity) between vectors, and grouping similar vectors together in an iterative fashion. The data, however, did not lend itself well to a traditional hierarchical clustering approach, because there were several outlier data points that caused the generation of single clusters from singleton data points. One way of dealing with this problem is to represent the data in a multidimensional space that can easily be divided into regions describing groups of HMMs. Multidimensional Scaling procedures were designed for this purpose.

	$M_1,\ldots,M_i,\ldots,M_N$
S_1 $\vdots$ S_j $\vdots$ S_N	$loglik_{ji} = \log L(S_j \mid M_i),\ 1 \le i,j \le N$

Figure 12: The format of the matrices describing the likelihoods of each of the knowledge sharing sequences given each of the HMMs

The origins of the Multidimensional Scaling (MDS) approach reside in the field of psychometrics, where scientists model and numerically characterize measures that are subjective or not clearly defined (Shepard, 1980). The MDS approach is based on the concept that a spatial representation of these perceived measures might help to explain their similarities or differences. A main drive in the development of MDS was that scientists believed that the popular clustering approaches, such as hierarchical clustering, imposed unnatural constraints on the organization of the data. For example, given a set of genealogical data, clustering methods might group aunt and uncle in one cluster, and niece and nephew in another. Once this occurs, it is no longer possible to form the gender-based clusters, aunt and niece, or uncle and nephew (Shepard & Arabie, 1979).

The Multidimensional Scaling (MDS) approach was attractive for this research because each of the groupings found in the multidimensional space might describe a particular way in which group members effectively share new knowledge with each other, or experience breakdowns while attempting to share new knowledge with each other. The full algorithm to perform the Multidimensional Scaling of HMM likelihoods is described in Figure 13.

In step 4 of the algorithm, the Multidimensional Scaling procedure was applied to the HMM log-likelihood matrices, such that $loglik(S_j \mid M_i) \rightarrow D_{ji}$, where D_{ji} describes the distance between the N HMMs in a 3 dimensional space. Let's briefly take a look at this procedure.

We begin by transposing the matrix shown in Figure 12, so that we have a set of vectors describing the likelihood of each of the sequences given each of the HMMs.

Algorithm: Multidimensional Scaling of HMM Likelihoods

1. Train N HMMs ($M_i, 1 \le i \le N$) from N sequences ($S_j, 1 \le j \le N$), where N = 58 (20 effective and 38 ineffective episodes)
2. Test each S_j with each of the N HMMs, producing an N^2 Log Likelihood matrix showing the log likelihood of each of the N sequences given model M_i
3. Transpose, then normalize the matrix. This step is important because transposing the matrix allows us to compare the models, M_i, to each other, and normalizing it allows us to factor out the relative log-likelihood of a sequence (which is directly proportional to the sequence length) as a discriminating parameter.
4. Perform multidimensional scaling: $L_{ji} \rightarrow D_{ji}$
5. Document groups of data points according to the following criteria:

   a. Distance from the center of a cluster to any member < θ (θ = 1.6 for effective groups, θ = 1.2 for ineffective groups)

   b. Distance from any member of a group to the center of the group is less than the distance from that member to the center of any other group

Figure 13: Algorithm for Multidimensional Scaling of HMM Likelihoods

$$L_{M1} = (l_{11}, l_{12}, ..., l_{1N})$$
$$L_{M2} = (l_{21}, l_{22}, ..., l_{2N})$$
$$L_{M3} = (l_{31}, l_{32}, ..., l_{3N})$$

We then calculate the Euclidean distance between the vectors: $D_{ji} = d(L_{Mj}, L_{Mi})$. To obtain the configuration of points in a multidimensional space, the point positions are adjusted via gradient descent, within the constraints of the matrix D_{ji}. The *stress* measure, *S*, is used to determine how well the configuration matches the data:

$$S = \left\{ \frac{\sum_{j,i} [(f(\delta_{ji}) - d_{ji}]^2}{\sum_{j,i} d_{ji}^2} \right\}^{\frac{1}{2}}$$

The idea here is to minimize *S*. The function, $f(\delta_{ji})$, in the numerator of this equation is an objective function (i.e. least squares linear regression) designed for this purpose (Kruskal & Wish, 1978). There are no hard and fast rules for determining how small S should be, or how many dimensions are optimal for modeling the data. Increasing the number of dimensions, however, does generally reduce the stress measure, at the sacrifice of the configuration's visual comprehension. Kruskal and Wish (1978) recommend that the number of stimuli minus one be at least four times the dimensionality.

Using 3 dimensions ($S \approx 0.2$), the MDS procedure was applied to the distance matrix generated in step 4 of the algorithm in Figure 5. The HMMs were then assigned to groups based on the closeness of the data points in the configurations. Figure 14 through Figure 16 show the left and right side, and top views for the effective matrices, D_{ji}. Figure 17 through Figure 19 show the left and right side, and top views for the ineffective matrices, D_{ji}. In these figures, the distance from any point (with label *i*) to each of the other point in the figure, represents the relative distance from the model M_i to each of the other models.

The groupings are also shown in the figures. These groupings are based on the closeness of points along all three dimensions, and therefore may not be obvious from looking only at one dimension. They were verified using an iterative, self-organizing data analysis technique (ISODATA) along with a maximum distance threshold criteria (Looney, 1997). The ISODATA algorithm is similar to a k-means clustering algorithm, except that it is able to split clusters containing feature vectors that exceed a maximum standard deviation threshold, σ_{split}, and it is also able to lump together clusters whose centers fall within a minimum distance threshold, d_L. The ISODATA algorithm was modified slightly, to include a maximum distance threshold, θ, as described in step 5 of the general algorithm shown in Figure 13. The maximum distance threshold enabled the algorithm to ignore those points that were too far away from any of the established clusters. The dataset that was analyzed was small compared to the number of different ways students may share new knowledge with each other. Even though some of the models in the dataset may represent single examples of certain types of interaction, only those models for which several examples exist can be reliably classified and analyzed. The additional maximum distance threshold criteria ensured that those models represented by only a single example would not be forced into a cluster.

The modified ISODATA procedure, applied to the scaled distance matrix, D_{ji}, discovered three groups of effective HMMs (A_e , B_e , and C_e), and four groups of ineffective HMMs (A_i , B_i , C_i, and D_i). Table 5 and Table 6 show the actual knowledge sharing episodes that correspond to the groups shown in Figure 14 through Figure 19. It is important to remember that the episodes listed in the third column of the tables are those sequences that were used to train the HMMs, whose relative distances were scaled and clustered to produce the groups listed in the second column of these tables.

In some cases, the HMM trained from the B-C swapped pair of a clustered sequence may not be included in the cluster. For example, cluster B_e contains episode L4 (row 2, column 3 of Table 5), which is represented by the sequence 14, and its B-C paired sequence 4. Note however, in column 2 of Table 5, that only the model trained from sequence 14 was found to be in cluster B_e. This is probably a result of not having enough training data. When actors B and C are swapped in a sequence, some patterns may result that are not present in other sequences. This was in fact the reason that the B-C swapped pairs were included in this analysis. A cluster that contains one or both paired sequences of an episode is considered to include that episode.

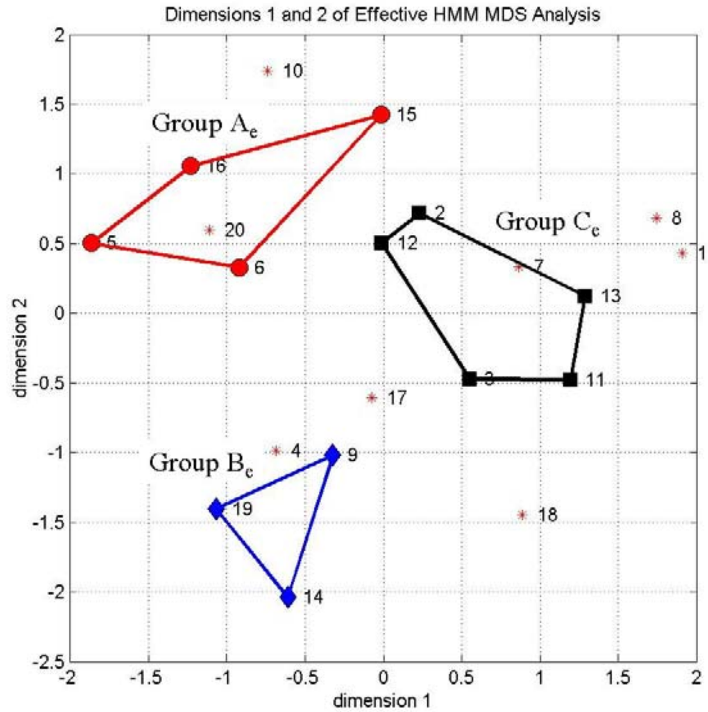


Figure 14: Side view of 3 dimensional space showing groups of effective HMMs

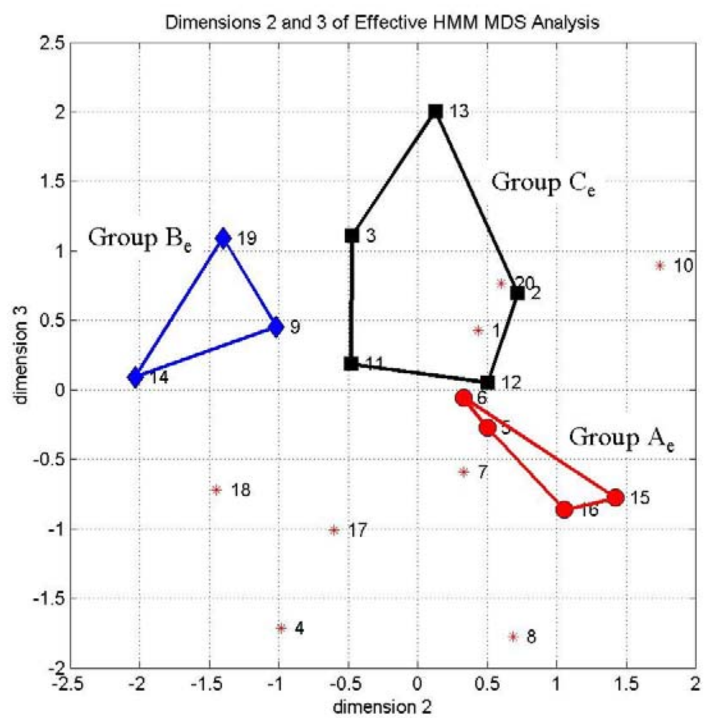


Figure 15: Side view of 3 dimensional space showing groups of effective HMMs

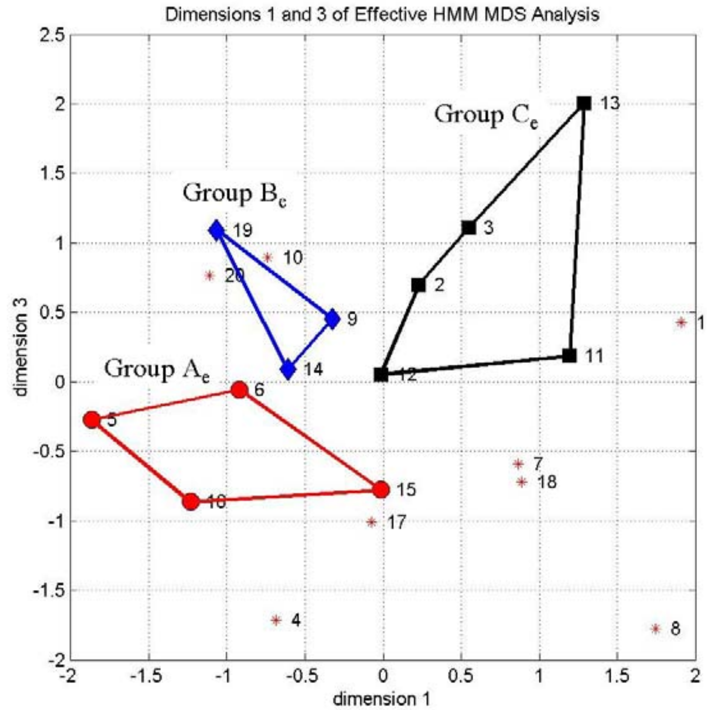


Figure 16: Top view of 3 dimensional space showing groups of effective HMMs

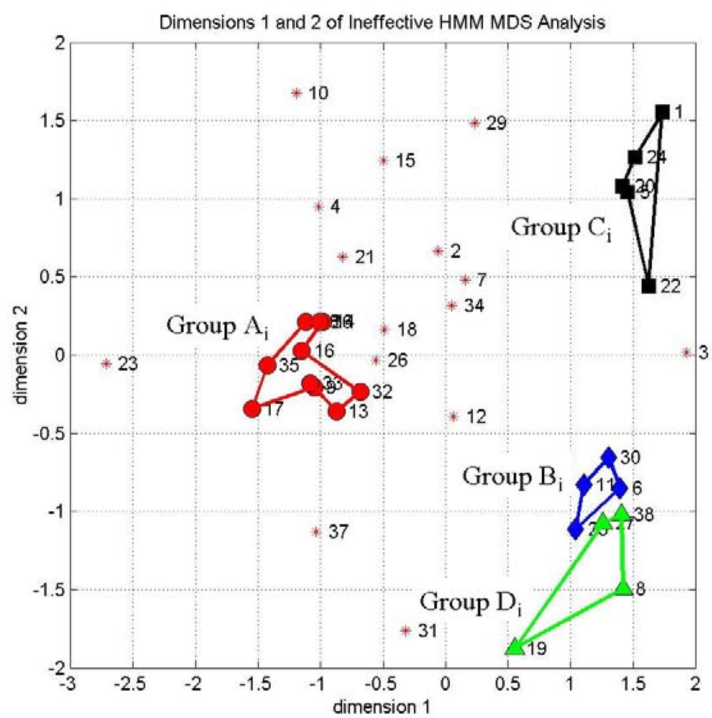


Figure 17: Side view of 3 dimensional space showing groups of ineffective HMMs

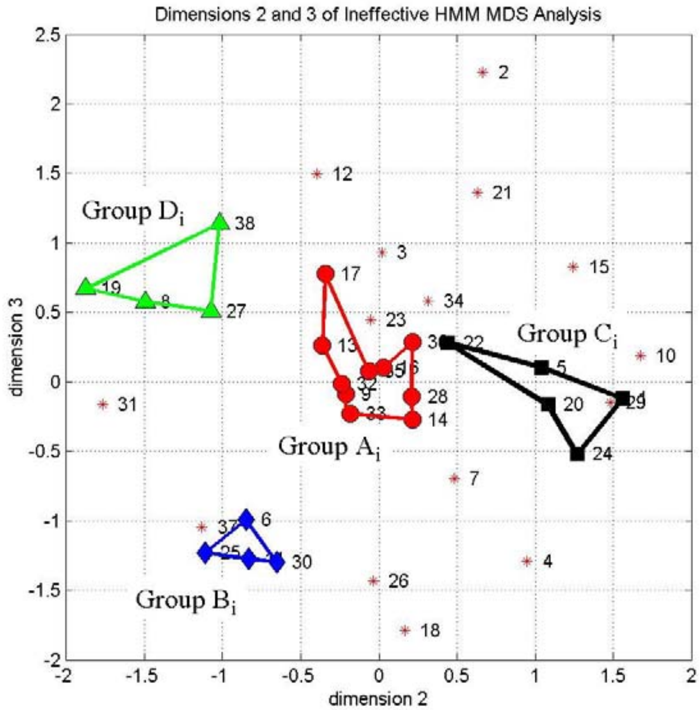


Figure 18: Side view of 3 dimensional space showing groups of ineffective HMMs

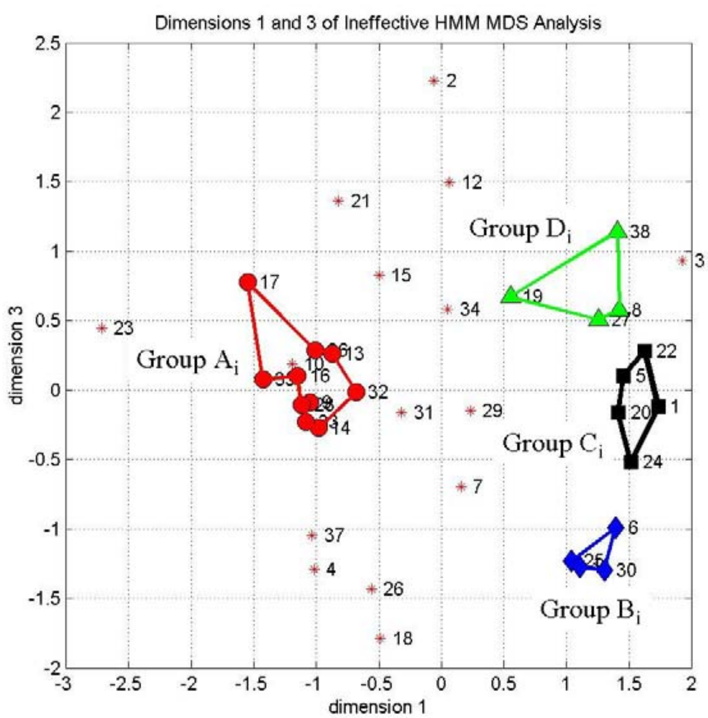


Figure 19: Top view of 3 dimensional space showing groups of ineffective HMMs

Table 5: Clusters of Effective HMMs from the MDS and Clustering Procedures

Clusters from Figure 14 through Figure 16	Labels (i) from Figure 14 through Figure 16	Corresponding Episode in **APPENDIX E**
A_e (Red Circle)	5, 15, 6, 16	L5, L6
B_e (Blue Diamond)	14, 9, 19	L4, M2
C_e (Black Rectangle)	11, 2, 12, 13	L1, L2, L3

Table 6: Clusters of Ineffective HMMs from the MDS and Clustering Procedures

Cluster from Figure 17 through Figure 19	Labels (i) from Figure 17 through Figure 19	Corresponding Episode in **APPENDIX E**
A_i (Red Circle)	9, 28, 13, 32, 14, 33, 16, 35, 17, 36	L16, M5, M6, M8, M9
B_i (Blue Diamond)	6, 25, 11, 30	L13, L18
C_i (Black Rectangle)	1, 20, 22, 5, 24	L8, L10, L12
D_i (Green Triangle)	8, 27, 19, 38	L15, M11

The remainder of the analysis presented in this chapter attempts to work backwards from the computational analysis to glean some notion of how students share new knowledge. This analysis is not expected to describe all the factors at work when students share new knowledge. Because the computational procedure is able to account for the inherent elements of randomness and noise in human communication, some patterns that the procedure finds and uses for classification may be difficult to discern qualitatively. The qualitative analysis that follows is intended to help explain the results of the computational procedure, but it may not explain all the results. For this reason, the qualitative analysis alone may not be sufficient for distinguishing the various mechanisms at work during knowledge sharing interaction, and the computational approach provides an invaluable tool in this process. This combination of computational and qualitative analyses should provide some insight as to why the students may be experiencing knowledge sharing breakdowns, so that the instructional module is better able to guide and support the group interaction.

Each grouping that was found through the MDS analysis and modified ISODATA procedure was compared to a qualitative analysis of the student activity in each of the groups. The episodes were first summarized blindly, without knowledge of the groupings. Then, the summarized episodes were compared to the clusters that were found computationally. The next few paragraphs describe, as suggested by these analyses, the sort of interaction that occurs when students attempt to share new knowledge with each other.

Table 7 shows the three generalized models that were found from the groups of effective examples (A_e , B_e , and C_e). In the first group (A_e), one of the receivers (student B or C) first requests information about one of the Knowledge Elements. This act is followed by an act in which the sharer (student A) provides an explanation, and then the receiver(s) acknowledge the help. In the summarized episodes shown in the table, the receivers are denoted by R1 or R2, and the sharer is denoted by S. In the second group (B_e), one of the receivers requests information about a Knowledge Element, the sharer provides some explanation, and then the receiver requests further clarification, after which the sharer provides further clarification. In the third group, (C_e), the sharer explains or illustrates his Knowledge Element, and the receiver(s) indicate they understand the new information by motivating or encouraging the sharer.

Table 8 shows the four generalized models that were found from the groups of ineffective examples (A_i , B_i , C_i , and D_i). In the first group (A_i), the sharer (student A) first proposes that the group discuss his Knowledge Element. The sharer then proceeds to either explain the Knowledge Element, or give instructions to one of the receivers (students B or C) for applying the Knowledge Element concept to the exercise. The episode closes with the receiver(s) simply acknowledging, or requesting confirmation of his actions. In the second group, (B_i), the sharer first attempts to explain his Knowledge Element. This act is followed by acknowledgement, and no further explanation. In the third group, (C_i), the sharer proposes his Knowledge Element, and this act is followed by doubt on the part of the receivers. In the fourth group, one of the receivers first requests an explanation of one of the Knowledge Elements, after which the sharer explains his Knowledge Element poorly, ending the discussion on the Knowledge Element.

In general, the discussions in which students effectively shared and learned each other's Knowledge Elements were marked by questioning, explanation, agreement, and motivation, whereas the discussions in which the students experienced breakdowns in knowledge sharing were marked by poor (inaccurate or incomplete) explanations, instructions for action, doubt, and acknowledgement. These elements alone, however, do not suffice to distinguish effective from ineffective sequences. This was confirmed through a Kolmogorov-Smirnov test comparing the distribution of the proportions of acts from the effective and ineffective sequences (Fisher & van Belle, 1993). The difference between the effective and ineffective distributions was not significant at the 5% level (p = 0.2607)[10]. As described in chapter 5, both noise and sequencing of the interaction play important roles in the characterization of knowledge sharing interaction, and a full reverse engineering of the computational analysis by qualitative means may be difficult.

Table 7: Summarized analysis of episodes corresponding to clustered effective HMMs

Group A_e	Group B_e	Group C_e
LRDC 5, 6	**LRDC 4 MITRE 2**	**LRDC 1, 2, 3**
R1: Suggest discussion related to KE S: Illustrate KE R1: Acknowledge/Accept S: Explain KE using diagram R2: Recommendation related to KE (regarding another example, to further understanding) S: Elaborate on KE R1 & R2: Agreement	S: Suggest KE R1: Request explanation S: Explain KE R1 & R2: Attempt rephrase using another example S: Clarify & Re-explain Some discussion, doubt, further explanation R2: Request further clarification (Further explanation by S & R1) R2: Attempt Rephrase KE	R1: General suggestion related to KE S: Explain KE S: Probe R R2: Elaborate on Explanation R1: Agree S: Begin to illustrate KE R1: Continue illustration R1: Request confirmation on drawing change S: Group maintenance R2: Encouragement/Agreement
R1 & R2: Request info related to KE S: Explain KE R1: Agree	S: Suggest KE R1: Request explanation of KE S: Explain KE R1: Request further clarification of KE (using another example) S: Clarify R2: Agree	S: Suggest KE S: Illustrate KE R1: Motivate
		R1: Request explanation of KE S: Explain KE R1: Agree R2: Encourage/Motivate

General Explanations for Each Effective Group

Group A_e	Group B_e	Group C_e
1.Receiver requests information about KE 2. Sharer provides explanation 3. Receiver agrees	1.Receiver requests information about KE 2. Sharer provides explanation 3.Receiver requests further clarification 4. Sharer provides further clarification	1. Sharer explains or illustrates KE 2. Receiver motivates / encourages

[10] For this analysis, students B and C were distinguished from the knowledge sharer (actor A), but were not distinguished from each other

Table 8: Summarized analysis of episodes corresponding to clustered ineffective HMMs

Group A_i	Group B_i	Group C_i	Group D_i
LRDC 16 MITRE 5, 6, 8, 9	**LRDC 13, 18**	**LRDC 8, 10, 12**	**LRDC 15 MITRE 11**
R1: Propose discussion related to KE S: Explain KE briefly R1: Acknowledge	R1: Request explanation of KE S: Explain KE briefly R1 & R2: Determine that KE is already done S: Recommend group reconsider KE R1 & R2: Acknowledge (no further discussion of KE)	S: Suggest KE R1: Doubt suggestion R1 & R2: Request elaboration of KE S: Explain KE unsatisfactorily (no further discussion of KE)	R1 & R2: Request discussion about KE S: Suggest that the KE is already done S: Offer brief explanation (no further discussion of KE)
S: Propose KE R1: Agree R2: Illustrate KE R2: Request confirmation of actions	S: Explain KE R1: Acknowledge S: Illustrate KE without explanation	S: Suggest KE R1: Doubt suggestion R1: Attempt rephrase KE R2: Suggest alternative R1: Agree with R2 S: Explain KE unsatisfactorily (no further discussion of KE)	R1: Request explanation of KE S: Explain KE (poorly) R1: Request elaboration (elaboration never provided)
S: Suggest KE R1: Acknowledge R1: Illustrate KE S: Reflect on actions		S: Recommend KE R1: Doubt that it is needed S: Explain KE unsatisfactorily R1: Doubt again S: Request help from R2	
S: Suggest R1 take action related to KE (no explanation) R1: Follows instructions from S			
S: Recommend KE R1: Follow instructions from S R2: Request verification of solution S: Provide verification R2: Acknowledge			

General Explanations for Each Ineffective Group

Group A_i	Group B_i	Group C_i	Group D_i
1. Sharer proposes KE 2. Sharer explains or gives instructions for action 3. Receiver acknowledges or requests confirmation	1. Sharer attempts to explain KE 2. Receiver acknowledges	1. Sharer proposes KE 2. Receiver doubts	1. Receiver requests explanation of KE 2. Sharer explains poorly (no further discussion)

The results of the groupings found through the clustering of the Multidimensional Scaled HMM Likelihoods were verified through a post-hoc modified take-2-out HMM cross validation study (similar to the take-2-out cross validation study described earlier). In the first step in the post-hoc cross validation study for the effective clusters, three generalized HMMs (M_{Ae} , M_{Be} , and M_{Ce}) were constructed by combining the models found through the MDS HMM clustering procedure (A_e, B_e, and C_e). For example, model $M_{Ae,}$ which combined models M_5, M_{15}, M_6, and M_{16} (see Table 5), was constructed by training on sequences S_5, S_{15}, S_6, and S_{16}. The knowledge sharing episodes corresponding to these combined models and their B-C matched sequences were then used to test the models. Each sequence was tested against each of the 3 HMMs (deleting the test sequence and its B-C matched sequence from the training set). This resulted in a 92.9% accuracy for the effective sequences.

In the first step in the post-hoc cross validation study for the ineffective clusters, four generalized HMMs (M_{Ai} , M_{Bi} , M_{Ci} , and M_{Di}) were constructed by combining the models found through the MDS HMM clustering procedure (A_i, B_i, C_i and D_i). The generalized ineffective models were then tested using the procedure described in the previous paragraph, resulting in a 95.8% accuracy for the ineffective sequences[11].

Table 9 and Table 10 show the actual log likelihood values for the groups. The shaded boxes in the tables illustrate which generalized models best matches each test sequence. Each test sequence was expected to most closely match the generalized model that represented its cluster. For example, model M_{Ae} represents cluster A_e, and was expected to match sequences L5 and L6, whereas model M_{Be} , representing cluster B_e, was expected to match sequences L4 and M2. The best match was given by the greatest path probability of the test sequence through the model, or in other words, the log likelihood value with the smallest absolute value. In the analysis of the effective groups, only one error was found: the B-C swapped version of M2 was incorrectly classified as a member of M_{Ae} instead of M_{Be}. Likewise, only one error was found in the analysis of the ineffective group: the B-C swapped version of L12 was incorrectly classified as a member of M_{Di} instead of M_{Ci}.

This analysis shows that, given a sequence classified as effective or ineffective (with 74.14% accuracy, from section 7.2), the described method is able to state why the students were

---

[11] For this post-hoc analysis, the algorithms were permitted to overtrain the models because they were considered generalizations of previously determined clusters.

experiencing a knowledge sharing breakdown, with 96% accuracy, or why the knowledge sharing interaction was effective, with 93% accuracy. One caveat, however, is that the breakdown or the success must be one that the system has seen previously, and is trained to recognize. For the current set of effective examples, this covers 70% of the sequences, and for the ineffective examples, this covers 63% of the sequences. A larger dataset, however, would certainly produce more examples, and hence more coverage.

Table 9: Log Likelihoods of effective grouped sequences given HMMs trained from each group

Test Sequences (s denotes B-C swapped version)	**Generalized Effective HMMs Constructed from MDS HMM Clusters**			**Explanations of Groups**
	M_{Ae}	M_{Be}	M_{Ce}	
L5	-105.16	-119.65	-135.29	1.Receiver requests information about KE 2. Sharer explains KE 3. Receiver agrees
L5s	-109.12	-113.55	-139.90	
L6	-43.55	-50.23	-56.23	
L6s	-46.87	-50.23	-56.23	
L4	-329.46	-295.07	-327.92	1.Receiver requests information about KE 2. Sharer explains KE 3.Receiver requests further clarification 4. Sharer clarifies
L4s	-338.69	-282.16	-328.34	
M2	-53.97	-48.73	-55.29	
M2s	-53.97	-57.43	-59.91	
L1	-197.26	-210.98	-172.29	1. Sharer explains or illustrates KE 2. Receiver motivates / encourages
L1s	-197.26	-210.98	-172.29	
L2	-52.48	-55.84	-26.52	
L2s	-52.48	-55.84	-46.75	
L3	-62.20	-65.79	-56.13	
L3s	-62.20	-65.79	-56.38	
			Accuracy: 0.9285	

Table 10: Log Likelihoods of ineffective grouped sequences given HMMs trained from each group

Test Sequences (s denotes B-C swapped version)	Generalized Ineffective HMMs Constructed from MDS HMM Clusters				Explanations of Groups
	M_{Ai}	M_{Bi}	M_{Ci}	M_{Di}	
L16	-24.09	-31.19	-30.10	-35.75	1. Sharer proposes KE
L16s	-22.90	-31.19	-30.10	-35.75	2. Sharer explains or gives
M5	-25.52	-31.19	-30.10	-29.06	instructions for action
M5s	-23.41	-31.19	-30.10	-31.50	3. Receiver acknowledges
M6	-28.94	-37.43	-36.13	-42.87	or requests confirmation
M6s	-29.15	-37.43	-36.13	-42.90	
M8	-18.11	-31.19	-30.10	-35.75	
M8s	-18.11	-31.19	-30.10	-35.75	
M9	-46.82	-62.38	-60.21	-64.82	
M9s	-46.32	-62.38	-60.21	-67.22	
L13	-53.47	-44.25	-52.82	-50.65	1. Sharer attempts to
L13s	-54.87	-41.87	-52.82	-44.96	explain KE
L18	-35.68	-28.19	-35.90	-31.95	2. Receiver acknowledges
L18s	-36.11	-26.78	-35.90	-31.96	
L8	-42.11	-49.70	-34.52	-52.09	1. Sharer proposes KE
L8s	-42.10	-49.70	-37.53	-52.09	2. Receiver doubts
L10	-36.09	-36.20	-28.89	-44.65	
L10s	-36.09	-36.20	-29.15	-44.65	
L12	-60.15	-61.77	-54.05	-54.37	
L12s	-60.16	-66.38	-57.08	-54.37	
L15	-76.75	-72.61	-73.22	-45.07	1. Receiver requests
L15s	-78.09	-72.62	-73.29	-60.93	explanation of KE
M11	-49.91	-51.25	-55.67	-34.42	2. Sharer explains poorly
M11s	-49.91	-51.27	-55.67	-28.57	(no further discussion)
				Accuracy: 0.9583	

The results of the post-hoc cross validation study confirmed that the groups found through the MDS and clustering techniques were stable. But, they did not confirm that the descriptions of these groups, from the qualitative analysis (i.e. 1. Sharer proposes KE, 2. Receiver doubts), reflect all of the commonalities of the examples that form the basis of the groups. The accuracy of these idealized descriptions was studied by testing the generalized HMMs with a set of coded mock test sets. These mock test sets represented, as closely as possible, the idealized descriptions of each cluster, from the last rows of Table 7 and Table 8. They are shown in Table 11.

Table 11: Hand-constructed sequences used to test the accuracy of the general cluster descriptions

Mock Test Sets for Effective Groups		
MOCK SET A_e	MOCK SET B_e	MOCK SET C_e
B-Request-Information	B-Request-Information	A-Inform-Suggest
A-Inform-Explain	A-Inform-Explain	A-KE-Action
A-Inform-Elaborate	B-Request-Clarification	B-Discuss-Agree
B-Discuss-Agree	A-Inform-Justify	C-Motivate-Encourage
C-Acknowledge-Accept	A-Inform-Elaborate	

Mock Test Sets for Ineffective Groups			
MOCK SET A_i	MOCK SET B_i	MOCK SET C_i	MOCK SET D_i
A-Inform-Suggest	A-Inform-Explain	A-Inform-Suggest	B-Request-Information
B-KE-Action	B-Acknowledge-Accept	B-Discuss-Doubt	A-Inform-Explain
A-Inform-Suggest	C-Acknowledge-Accept	A-Inform-Explain	C-Request-Elaboration
B-KE-Action		C-Discuss-Doubt	
C-Acknowledge-Accept		A-Inform-Elaborate	

The mock test sets produced only 50% accuracy (2 out of 4 test sequences were classified correctly) for the ineffective models, and 66.7% accuracy (2 out of 3 test sequences were classified correctly) for the effective models. The results are shown in Table 12.

There are two things we can learn from these results. First, machine learning methods are notorious for producing good, but unexplainable results. This is not unusual, because in situations in which a simple explanation suffices, machine learning techniques would not have been needed in the first place. The techniques applied and described in this chapter produced good results when tasked to classify sequences of coded knowledge sharing interaction. Reverse-engineering these techniques qualitatively may help explain why they worked well; however, by its nature, a qualitative procedure cannot explain beyond the stochastic elements of a computational method.

Second, the mock test sets were designed to filter out all of the noise in the sequences, and include only the essential items that were perceived to be similar within the groups. If the noise were in fact an essential element of the models, we would expect them to perform poorly in classifying test sets that lack this essential component. This analysis supports the concept that some degree of noise is as important in these models as it is natural in human communication. Simple state based models or grammars that do not take some degree of randomness into account may not be as effective at procedures that do account for noise.

Table 12: Log likelihoods for mock test sequences

Mock Test Sequences	HMMs Constructed from Ineffective Sequences in MDS Groups A through D			
	M_{Ae}	M_{Be}	M_{Ce}	M_{De}
Mock Set A_i	-14.2597	-30.6748	-32.9575	-31.5607
Mock Set B_i	-14.5856	-16.9042	-18.8786	-18.2343
Mock Set C_i	-31.0323	-33.4878	-22.9163	-35.9535
Mock Set D_i	-22.7844	-20.4469	-23.7765	-22.7877

Mock Test Sequences	HMMs Constructed from Effective Sequences in MDS Groups A through C		
	M_{Ai}	M_{Bi}	M_{Ai}
Mock Set A_e	-36.7330	-31.4628	-33.4835
Mock Set B_e	-40.2846	-29.0386	-34.6466
Mock Set C_e	-24.9346	-25.9779	-20.0007

8 Discussion

This chapter will briefly summarize the motivation for this research, and then revisit the three claims set forth in chapter 5. The results presented in chapter 7 will be further analyzed and discussed in light of these claims and the literature review in the first three chapters of this dissertation.

The Information Age. It is a time when organizational success requires the constant generation, transfer, and understanding of knowledge, making collaboration an essential and highly valued process. Education and training programs that adopt collaborative learning practices in their curriculums may prepare students for today's fast-moving information-based society better than traditional classroom instruction. Collaboration may also enrich students' individual learning experiences by motivating them to seek new insights and perspectives, ask questions openly, and practice explaining difficult concepts (chapter 2). The extent to which these benefits are realized depend largely on the effectiveness of the group interaction (chapter 3); when students do not collaborate effectively, the social and cognitive advantages of group learning are lost.

Structuring, guiding, and mediating collaborative learning activities can increase both individual and group performance (chapter 4). Providing this support in distance learning environments is particularly challenging because student interaction may be limited by the technology, and instructors often do not have time to mediate online discussions. The long-term vision for this research is to computationally support online students interaction, so that distance learning students may maximize their potential learning gain.

Supporting group learning requires an understanding of the cognitive and social processes that impact collaborative learning, and knowledge of the facilitation methods that target these processes. In chapters 2 and 3, we saw how a group's ability to co-construct knowledge is an important predictor of the value of the group learning experience. The effectiveness of knowledge co-construction depends on the participants' evolving knowledge bases and the group's ability to share and assimilate the bits of knowledge necessary to construct new knowledge. As the shared bits of knowledge are assimilated into the group thinking process, group members evolve and develop a shared understanding. Before attempting to support this process, we must be able to recognize it, and understand why and how students have breakdowns in knowledge sharing. This dissertation research looked at how we might assist a distance learning instructor by computationally performing these actions.

Chapter 5 presented a proof of concept framework for a distance learning system that could support knowledge sharing interaction. The framework is shown in Figure 20. The grayed components in the figure were presented as deliverables in this dissertation. More important than the physical deliverables was the evaluation of a stochastic state-based method for assessing sequences of student conversation and activity. The proof of concept framework is intended to provide the reader with an idea of the overall long-term vision, and to show how this dissertation contributes to the development of a comprehensive distance collaborative learning support environment.

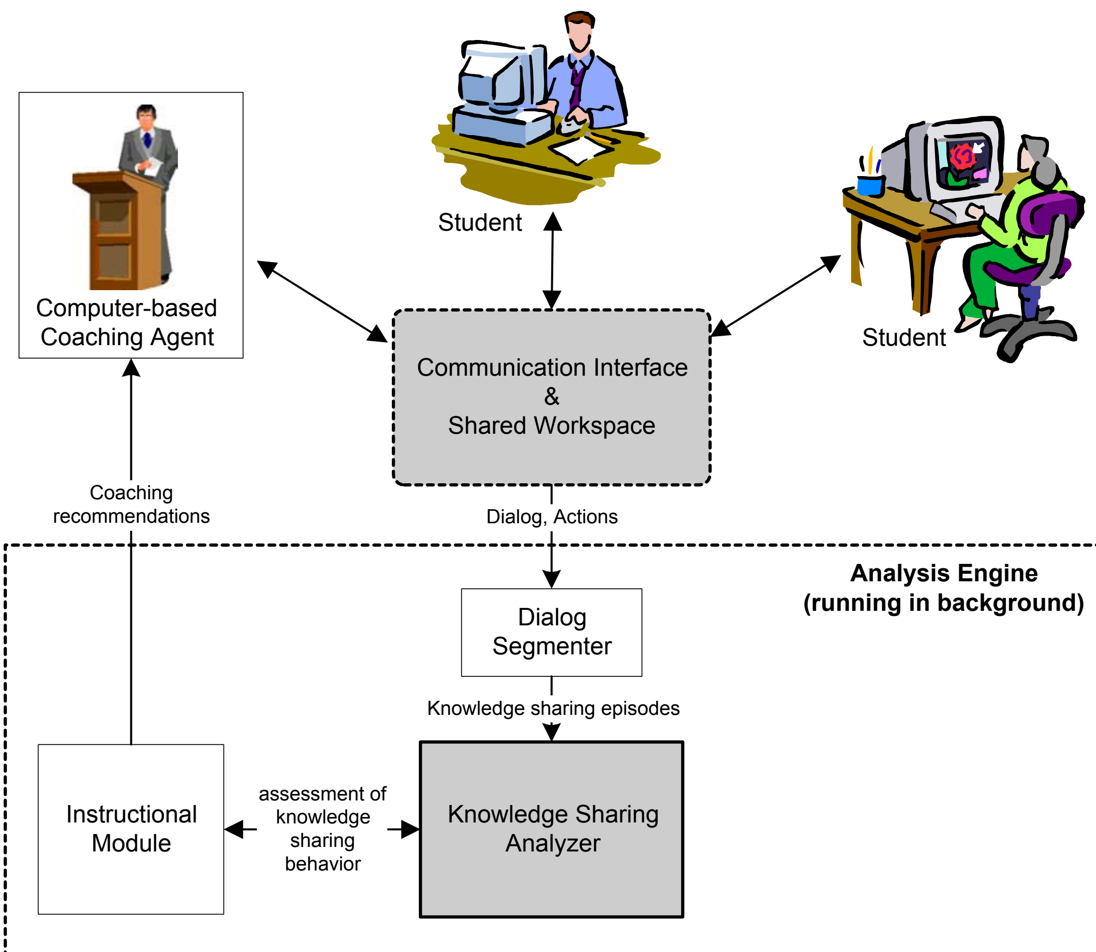


Figure 20: A proof of concept framework for supporting online knowledge sharing during collaborative learning activities

The role of the Knowledge Sharing Analyzer (Figure 20) is to determine how well new knowledge is assimilated by a distance learning group. This is a very difficult task, involving the analysis and assessment of natural language. In chapter 4, we saw a few different methods that other researchers have explored, including finite state machines (McManus & Aiken, 1995), fuzzy inferencing (Barros & Verdejo, 1999), decision trees (Constantino-Gonzalez & Suthers, 2000; Goodman, Hitzeman, Linton, & Ross, 2002), rule learning (Katz, Aronis, & Creitz, 1999), and plan recognition (Muehlenbrock & Hoppe, 1999), for analyzing collaborative learning interaction (see Jermann, Soller, & Muehlenbrock, 2001, for a review of different approaches). CSCL researchers have begun to develop a toolbox of methods and strategies for understanding and computationally supporting various aspects of online collaborative learning behavior. Missing from this toolbox were the tools needed to model and assess complex cognitive and social processes by analyzing naturally occurring sequences of peer dialog and actions (on computer artifacts). One such complex cognitive and social process is that of knowledge sharing and assimilation.

Knowledge sharing conversations, especially among groups larger than 2, are dynamic, progressive, and unpredictable. This dissertation tested the possibility that analyzing these conversations with a stochastic, sequential, machine learning tool could promote a promising new CSCL research direction. Tools that have been used in the past, such as finite state machines and decision trees, can only account for the sorts of conversational patterns that one might predict ahead of time. Computational grammars are useful for analyzing conversations of dyads, but in groups of three or more people, there may be multiple agendas of conversation occurring simultaneously, causing the conversation to take on a stochastic character. Hidden Markov Models are stochastic, state-based, and appear to be able to account for the seemingly random nature of conversation in groups of three. This dissertation explored the notion that HMMs might be used to identify and explain when and why students have breakdowns during knowledge sharing conversations.

8.1 Consideration of Claims 1 and 2

The HMM approach performed at almost 25% above the baseline when tasked to identify whether or not students are effectively sharing the new knowledge they bring to bear on the problem, or experiencing knowledge sharing breakdowns. This approach appears promising, but why? If we can answer this question, then we may also discover what elements are important to consider when distinguishing effective knowledge sharing episodes from ineffective ones.

The HMMS, in this dissertation, were trained to represent the possible ways that a student might share new knowledge with his teammates, and the possible ways that his teammates might react. A knowledge sharing HMM therefore represents a sort of compiled conversational model. This means that, for example, the effective model includes a compilation of the conversational patterns students use when knowledge is effectively built by the group members. These patterns are constructed from sequences of conversation acts and workspace actions. It appears that, armed with an intelligent stochastic machine learning method, knowledge of how students' conversational patterns and actions change over time enables us (with 74% accuracy) to distinguish between effective and ineffective knowledge sharing interaction.

These conversational patterns are noisy; they may contain seemingly random elements unseen to any of the recognizable patterns, and interruptions or breaks. By slightly undertraining the HMMs in this research, I was better able to account for this noise. For example, in testing Claim 1, the best results were found when the HMMs were trained over only two iterations. Initially, the models were seeded with random numbers. After two iterations, the models reflected the elements in the training set, but still contained a significant amount of the random noise from the seeding process. If training were to proceed further, this noise would be eliminated, and replaced by whatever idiosyncrasies existed in the training set. There seems to be an important distinction between the particulars of the training set (which do not necessarily appear as random elements), and the random noise that is necessary for modeling these sequences. A larger training set might lessen this effect, although it would probably not eliminate it because noise appears to be an important element in knowledge sharing episodes. Statistical methods that do not account for noise in these sequences perform less well than the HMM approach (chapter 5).

Another reason the HMMs performed well in this task is that they are specifically designed to model sequences of events. The actions that students take, and the order in which they take them, both help in analyzing the episodes. For example, a situation in which a student explains a concept, and then his peer expresses doubt, is very different from a situation in which a student expresses doubt, and then his peer provides an explanation. When the order of events in the 29 knowledge sharing sequences was randomly scrambled, the HMMs achieved only 65%

accuracy, compared to 74% accuracy for the unscrambled sequences. Scrambling the order of events in these sequences clearly decreases the predictive power of the models, which rely on the order of events for classification. This analysis helps to explain why choosing a method that accounts for the sequentiality of human interaction is essential for analyzing collaborative learning interaction.

HMMs belong to a set of models that make predictions using *Linear Statistical Queries (LSQ) Hypotheses*, allowing them to perform well on unforeseen data (Roth, 1999). These hypotheses represent linear predictors over a set of features that are directly related to the independence assumptions of the probabilistic model. The features reflect statistical properties of the data, and are not necessarily representative of the specific examples from which they are derived. Because of this, LSQ algorithms are robust, and will produce good results even if the training data and testing data are sampled from different distributions. Roth (1999) showed that, "the Markov predictor is an LSQ algorithm over a set of X of singletons and pairs of [outputs] and [states] (Roth, 1999)".

Other LSQ algorithms, or stochastic, sequentially oriented machine learning approaches may perform as well as the HMM approach. For example, Alan Berfield trained a set of Genetic Algorithms (GAs) with data from the first 5 LRDC groups. The GAs were trained to learn either a state machine, or a set of coefficients for a set of linear combination functions. The GAs performed comparably to the HMMs on the same data (A. Berfield, personal communication, April 22, 2002). The GA approach, like the HMM approach, attends to the stochastic and sequential nature of the data.

The HMMs in this study performed well compared to the baseline and statistical procedures. But they still retain a 25% error rate, and this is reason enough to consider the consequences of error. In the implementation of the proof of concept system (Figure 20), the most interesting case for the knowledge sharing analyzer is that in which a knowledge sharing breakdown occurred. It is the breakdown that should trigger the instructional module to take action to support the students. In this case, the consequences of the knowledge sharing analyzer making an error in diagnosing the student interaction are not severe. They might, for example, involve an instructor or computer-based coaching agent briefly interrupting the student interaction, and perhaps offering advice or suggestions when it is not necessarily needed. In the alternative case, the instructor or the computer-based coaching agent might miss the opportunity to offer advice to the group when it is needed. In this case, however, the data suggests that there is a good chance the system will pick up on the breakdown the next time it occurs.

8.2 Consideration of Claim 3

In the second phase of this dissertation work, I attempted to provide some explanation for why the students did, or did not, effectively share the new knowledge they brought to bear on the problem. For this, I applied a multidimensional scaling procedure to the HMM likelihood vectors, and clustered the resulting matrix using the self-organizing ISODATA algorithm. This procedure produced three groups of effective knowledge sharing models, and four groups of ineffective knowledge sharing models.

Chapter 7 presented qualitative descriptions for the seven groups of HMMs that were found from the MDS and clustering procedures. One might wonder how well these characterizations match those that, according to educational research, indicate when students are learning together effectively. Although most educational research does not perform the sort of low-level sequential analysis that is necessary for machine understanding and classification, some parallels can be drawn between the findings here, and the research summarized in chapter 2.

The qualitative analysis of the four ineffective HMM clusters describes sequences of student activity that may lead to knowledge sharing breakdowns. These sequences bear strong parallels to Webb's (1992) model of student helping behavior. Webb (1992) describes the actions that might lead students to positive learning outcomes after they request help from a peer (see Figure 21). The student either answers his own question and corrects his own error (lower path in Figure 21), or receives help that is timely, relevant, and sufficiently elaborated such that the he understands and is able to apply the help (upper path in Figure 21). There exists a possibility of stagnancy at each of the stages in the upper path. The student may not express his need for help, or he may not receive timely, relevant or elaborated help. And even if he does, there is the possibility he may not understand the explanation, or get the chance to apply it. Finally, satisfying all of these stages may not bring the student to the end stage of learning the material.

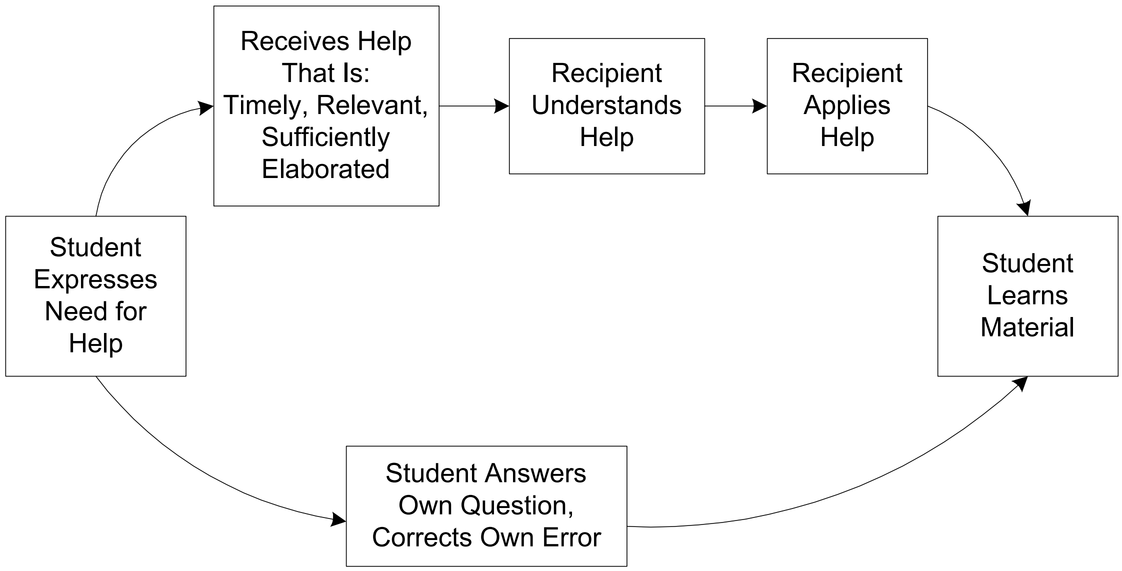


Figure 21: Sequences of experiences leading to positive learning outcomes for students who express a need for help. From "Testing a Theoretical Model of Student Interaction and Learning in Small Groups," by N. Webb, 1992, In R. Hertz-Lazarowitz and N. Miller (Eds.), *Interaction in Cooperative Groups: The Theoretical Anatomy of Group Learning*, p. 105. Copyright 1992 by the Cambridge University Press. Reprinted with permission.

If any of the conditions along the upper path fail while one student is helping another learn some new material, a knowledge sharing breakdown might occur. Some of these breakdowns are described by the qualitative analysis of the clusters derived from the MDS analysis. They should, therefore, roughly describe what happens "inside the box", when there is a failure at each of the Webb's stages.

Table 13 shows the qualitative descriptions for the four groups of ineffective HMMs that were found from the MDS and clustering procedures across from the stages along the upper path of Webb's model. Each sequential description is shown across from the stage in which the breakdown most likely occurred. For example, in the first row of the table, if the sharer proposes that the group consider her Knowledge Element, and one of the receivers doubts that suggestion, then the receiver has not realized, much less expressed, his need for help. In the second row, if the receiver does express his need for help, but then the sharer provides a poor explanation, ending the topic of conversation at that point, then the receiver has not received sufficiently elaborated help – the condition in Webb's second stage.

Table 13: Parallels between Webb's model, and the descriptions of ineffective MDS clusters found in this research

Stage from Upper Path in Figure 21	Description of Failed Stage	MDS Cluster	Descriptions of Ineffective MDS Clusters
Student expresses need for help	Student does not express the need for help	C_i	1. Sharer proposes KE 2. Receiver doubts
Student receives help that is timely, relevant, and sufficiently elaborated	Student does not receive timely, relevant, or elaborated help	D_i	1. Receiver requests explanation of KE 2. Sharer explains poorly (no further discussion)
Student understands the help	Student does not understand the help	A_i	1. Sharer proposes KE 2. Sharer explains or gives instructions for action 3. Receiver acknowledges or requests confirmation
Student applies the help	Student does not get the chance to apply the help	B_i	1. Sharer attempts to explain KE 2. Receiver acknowledges

Table 14 shows the qualitative descriptions of the three effective HMM clusters from chapter 7. Two of these (groups A_e and C_e) involve the knowledge sharer providing an explanation, and the knowledge receiver(s) indicating that he understands the explanation by agreeing with or encouraging the sharer. These sequences correspond to at least the third stage in Webb's (1992) model. The sequences in Group B_e involve cognitive conflict; the receiver recognizes an inconsistency between his understanding and that of the sharer, and probes the sharer for additional information. Cognitive conflict, under certain conditions, has been shown to produce effective collaboration. Recall from chapter 2 that Doise, Mugny, and Perret-Clermont (1975) found the most effective conversations to be those in which students were requested to defend their viewpoints in a coherent manner. It is no surprise that the successful students are the ones who, when experiencing cognitive conflict, resolve their differences of opinions through knowledge sharing and discovery (Okada & Simon, 1997).

Table 14: Descriptions of effective HMM clusters from MDS analysis

Group A_e	Group B_e	Group C_e
1.Receiver requests information about KE 2. Sharer provides explanation 3. Receiver agrees	1.Receiver requests information about KE 2. Sharer provides explanation 3.Receiver requests further clarification 4. Sharer provides further clarification	1. Sharer explains or illustrates KE 2. Receiver motivates / encourages

Not only do the groups of models describing knowledge sharing interaction, that were found computationally, make sense in terms of the sorts of patterns found in educational research – they also broaden our understanding of the kinds of patterns we might expect to see during collaborative distance learning. Most importantly, they help to further specify the patterns of student activity inside the boxes of Webb's model, when students may be having trouble.

This dissertation has contributed toward the development of a system for supporting knowledge sharing through the implementation of the knowledge sharing analyzer and the communication interface. Developing and evaluating these components also contributed to our understanding of knowledge sharing activities, and toward the realization of the other components in the proof of concept framework shown in Figure 20. Table 15 lists the components in the framework, and summarizes this dissertation's deliverables. The novel contributions made by this research are those listed across from the Knowledge Sharing Analyzer. The contributions made by the first three components were discussed in chapters 5, 6 and 7. The instructional module and online coach will be discussed in the next chapter.

Table 15: A summary of the proof of concept framework, and deliverables

System Component	Contribution
Networked Communication Interface & Shared Workspace	- Conversation Act chat style interface based on CLC Conversation Skills Network - Shared OMT workspace - Integrated logging facility - Agenda tool
Dialog Segmenter	- Suggestions and guidelines on how to recognize the onset and termination of knowledge sharing episodes.
Knowledge Sharing Analyzer	- Determination of knowledge sharer - Assessment of coded knowledge sharing interaction using HMMs - Strengths & weaknesses of HMM approach - Explanation of possible knowledge sharing breakdowns using Multidimensional Scaling, clustering techniques, and qualitative analysis
Instructional Module	- Recommendations, based on literature in educational and small group psychology, for supporting knowledge sharing under the various situations recognized by the knowledge sharing component
Online Computer-based Coaching Agent	- This component should be straightforward to develop if driven directly by the analysis engine, as suggested. This dissertation contributes to the development of this component indirectly, through its contributions toward the development of the analysis engine

9 Directions for Future Research

This research has shown how a stochastic analysis of coded sequences of student interaction, composed of conversation acts and student actions on a shared workspace, can provide useful information about whether or not students are effectively sharing and learning new knowledge with each other. My hope is that this research will encourage others to pursue similar paths. Because this is a new area of research, there are many paths available, none of which have yet been traveled.

Four clear paths come to mind. The first path involves improving the automated coding, or HMM, MDS, and clustering procedures with the intent to increase the accuracy of the system. The second involves trying out similar machine learning procedures on similar forms of data, and comparing the results to those reported here. The third involves studying how this procedure fares in analyzing aspects of collaboration other than knowledge sharing. And the fourth involves taking this work one step further, by using it to provide instructional support. This chapter will provide some direction to assist researchers in pursuing these four paths.

9.1 Improving the Procedure

Research along the first path looks at improving the method described here. Although the HMMs in the first phase of this research performed well, the 25% error rate suggests that there is still room for improvement. The codes that make up the HMM training sequences are given by the sentence openers that the students choose. The kappa analysis in chapter 5 showed that students generally begin their contributions with the most appropriate sentence openers on the interface. But because each sentence opener is associated with only one intention (i.e. *Suggest* or *Justify*), the coding scheme is only able to account for the primary intention. It cannot capture complex intentions, such as a *Discuss/Agree* act that both expresses agreement and doubt. To what degree might a more complicated coding scheme improve the system's accuracy? Assigning complex intentions to utterances is quite a difficult task for human raters, and an even more difficult task for computers. Having a system assign secondary intentions to student contributions may introduce additional error into the training data, reducing the accuracy of the system. Further research along these lines may help to determine the extent to which considering only the primary intention limits our ability to assess learning during knowledge sharing interaction.

Alternatively, the HMM approach itself might be improved upon. For example, it is possible to manipulate the number and placement of nodes and links in the HMM, changing its topology. This means that one could form hypotheses about various phases that might be present in knowledge sharing episodes (or episodes of other types of interaction), and test these hypotheses by studying the output from different HMM structures (Schrodt, 1999). If, in fact, definable phases exist for learning interaction (e.g. a proposal phase, or an explanation phase), then a topology that better reflects this phase structure may increase the system's accuracy. Hill climbing procedures may also be used to train a system to learn the optimal HMM topology (Freitag & McCallum, 2000).

In the second phase of this research, 58 individual HMMs were trained from the 58 knowledge sharing episodes before applying the MDS and clustering routines. Training an HMM on only one example introduces a good deal of noise into the process. It may be possible to generalize the examples before this training occurs, and reduce the amount of noise generated before running the MDS procedure. One way to do this is to remove parts of sequences that do not recur often. For example, one training sequence may contain the code, A-Discuss-Suppose, which is not present in any other example, and is highly unlikely to recur. This code could probably be safely omitted from the model, because it is not particularly helpful for classification. Failing to omit such a code may even reduce the model's ability to classify new instances accurately if training on the unlikely code reduces the effectiveness of training on other codes or factors that harbor more predictive power.

Perhaps an obvious way to improve the accuracy of the system is to gather more data. Data could be gathered by running more experiments, similar to the ones described here, or by having students interact with the system and provide feedback. Imagine a scenario in which the system identifies a knowledge sharing breakdown and offers advice to the students, then asks the students to respond with an evaluation of the advice. The system might then integrate the student evaluations into its models. Or, if the system fails to offer advice during a knowledge sharing episode that it believes is effective, and the students are able to request help (i.e. by clicking on a "HELP" button) at the time, then the system might self-update by adding the current knowledge sharing episode to its database.

9.2 Testing and Comparing Similar Methods

The second path worthwhile considering is that in which methods similar to the one employed here are applied, and compared to my approach. Hidden Markov Models may not be the only stochastic, state-based method useful for analyzing and assessing sequences of student interaction. It seems clear that some degree of sequencing, randomness, and noise is important; however I see no reason why other, similar LSQ algorithms (Roth, 1999), would not perform comparably. As discussed in chapter 8, a preliminary analysis done on the first five LRDC run groups from this dissertation study has shown that genetic algorithms may also perform just as well as HMMs (A. Berfield, personal communication, April 22, 2002).

Since it is clear that HMMs are useful tools for evaluating sequences of student interaction, another approach might be to use them in conjunction with other methods, and compare the results to the combination of methods used in this research. For example, the path probabilities obtained from the HMM approach may be used as one factor, among others, that contributes to a weighted assessment function. Walker, Litman, Kamm, and Abella (1997) have designed a method for evaluating systems that involves computing weighted combinations of factors such as user satisfaction and task success. In a similar vein, a cost function that evaluates knowledge sharing effectiveness might combine weighted HMM path probabilities with other variables, obtained statistically, or through other methods. Weighted combinations of factors can also serve as feature vectors in decision trees, or input layers in neural networks.

An example of another factor that might be combined with HMM path probabilities was hinted at in the previous section. While altering the HMM topology may or may not increase the accuracy of the system, the process of manipulating and studying the topology might provide clues about which elements of the interaction best predict knowledge sharing effectiveness. These elements could then be used as predicting features in an evaluation function, neural network, or decision tree algorithm.

9.3 Applying this Research to Other Aspects of Collaboration

As we saw in chapter 3, many factors influence the outcome of group learning situations, and knowledge sharing is just one of these factors. Supporting collaborative learning means attending to all of these factors, many of which are interrelated. The success of the HMM approach in analyzing knowledge sharing activities suggests that this sort of method might also be used to analyze other aspects of collaboration. But not every aspect is candidate. Knowledge sharing is different from some other aspects of collaboration in that it can be measured and evaluated post-hoc. This is important because supervised machine learning methods, such as HMMs, require the training data to be classified. In this case, the experiments were set up such that the sharing and learning of specific knowledge elements could be measured via pre and post tests.

Lately, grounding has gotten a lot of attention in the CSCL community. Researchers are finding more and more evidence that effective grounding leads to learning during collaborative activities (e.g. Baker, Hansen, Joiner, & Traum, 1999). Grounding is similar to knowledge sharing in that it helps students establish and maintain a shared understanding. It may, however, not be as good a candidate for an intelligent machine driven assessment approach, because its application would entail attempting to measure the degree to which two collaborators are sharing (or shared) the same sort of experience (Koschmann, 2002). Measuring this is impractical because even collaborators often do not know whether or not they share a common conception of their discussion. They often think they share the same interpretation of a conception, only to later find out that their manifestations are different. Likewise, we cannot assume that students who are working together and sharing a common workspace also share a common mental representation (Dillenbourg & Traum, 1999). This does not mean we need to retire the idea of computationally analyzing complex factors like grounding. Instead, we should try to break these factors down into their constituent sub-factors. For example, if we were to take a closer look at the grounding process, we might find inferencing, self-construction, and … we might even find a little knowledge sharing!

A clearer, perhaps simpler route to take might entail modeling factors such as cognitive conflict or confidence. Cognitive conflict might be measured by comparing the degree of disagreement before and after a problem solving session. Confidence, although less concrete, might be measured through questionnaires, and analyzed by modeling sequences of student interaction and problem solving. The task might also be designed to include various situations intended to alter students' confidence levels, so that the interaction can be observed in those contexts.

Applying this research to other aspects of collaboration need not mean taking an HMM or similar approach. This dissertation demonstrated not only the utility of the HMM approach, but also the ability of a system to make inferences about student learning from sequences of conversation acts and student actions. Researchers interested in studying other aspects of collaboration may also benefit from analyzing such sequences. For example, work is now in progress at the MITRE Corporation, in Bedford, Massachusetts, to analyze coded sequences of student interaction in order to assess students' confidence levels (F. Linton and H. Ross, personal communication, July 3, 2002). The sequences were obtained from the interface described in chapter 5, and consequently use the same coding scheme as this research.

9.4 Providing Instructional Support

Once a knowledge sharing breakdown is identified, the next step is to determine how to facilitate the group interaction. Research along this fourth path focuses on how to provide this support. This section is intended to provide some direction to the curriculum developers and instructional designers who might assist system developers in constructing and implementing appropriate responses to student actions.

The first step in providing support is determining when the students could use help. If the system were to offer guidance to the students each time the knowledge sharing analyzer identified a knowledge sharing breakdown, the students would receive help about 74% of the time they need it. This is based on the results of this research showing that the knowledge sharing analyzer can correctly classify the coded sequences about 74% of the time. As discussed in chapter 8, this may be sufficient because if the knowledge sharing breakdown is severe, it will probably recur, in which case the system would have another opportunity to address it. It might also be advantageous to include a help button on the interface, so that students can request help even if the system fails to detect a problem (discussed earlier in this chapter).

The next step in providing support is to determine whether or not the knowledge sharing episode in question matches one of the identified clusters that were described in chapter 7. If a cluster is identified, then the nature of the cluster should drive the selection of an appropriate support strategy. If no cluster is identified, but the episode has been classified as a knowledge sharing breakdown, then two possibilities exist: some general advice could be offered based on the particular knowledge element that is in question, or more tailored advice could be offered based on the HMM cluster closest to the target episode.

In the development and selection of appropriate support strategies, understanding what students are doing when they succeed is just as important as understanding why students are failing. The data collected for this study revealed that effective knowledge sharing patterns include situations in which the receiver probes the sharer for information, the sharer provides justification and clarification, and the receiver provides motivation and encouragement[12]. In developing support strategies, these are some of the behaviors that we might like to encourage. This means that when the system recognizes that the sharer has explained his Knowledge Element, and the receiver has either doubted (case C_i) the suggestion, or said nothing at all (case D_i), there is an opportunity for the system, or instructor, to encourage the sharer to justify, clarify, or re-explain the Knowledge Element. If the receiver acknowledges the explanation, but the system still has evidence that the receiver does not understand the new material, then there is an opportunity for the system, or instructor, to recommend that the receiver attempt to apply the new knowledge to the problem at hand. These suggestions follow from Webb's (1992) model of student interaction, discussed in chapter 8.

In the future, collaborative learning technology will continue to enrich distance learning programs by connecting peers, bringing the virtual classroom to life. Distance learning programs will become seamlessly integrated into our educational system, and meeting the educational and social needs of evolving online learning groups will become even more of a challenge. Efforts to leverage cross disciplinary research to further develop computer-based support for online groups will no doubt benefit both instructors and students. We have just begun to construct a toolbox of methods for helping to manage, facilitate, and support various online collaborative learning activities. Future research should continue to fill, examine, and augment this toolbox with new techniques. With a more complete toolbox at hand, researchers may be better suited to adopt flexible and holistic views of supporting learning communities. This dissertation contributed to the development of this toolbox by introducing and testing a new tool that aids in the understanding and analysis of a critical aspect of collaborative learning – knowledge sharing.

[12] This list is certainly not exhaustive, and a larger data collection would almost certainly produce more patterns.

APPENDICES

APPENDIX A

Sentence Opener Tutorial

Note: Words enclosed in <> are instructions to the experimenter. All other text is spoken verbatim to the subjects. Each category includes three exercises. The experimenter should read one exercise per category to each of the three subjects, in round robin style.

Imagine that you have decided to be roommates, and you are moving into your new 3 bedroom virtual apartment. Each of you has brought some furniture, appliances, and decorations for the apartment, and you are now deciding how to arrange everything

Request

How would you (using the sentence opener interface)…
1. Ask your roommate if s/he thinks it's ok if you put the TV in the hallway
2. Ask your roommate to show you where she is going to put her impressionist painting
3. Ask your roommate why s/he thinks that splitting the cost of digital cable is unfair

<Read (out loud) the sentence openers in the Request category >

Inform

How would you (using the sentence opener interface)…
1. Tell your roommates that you think everyone should chip in to buy a nice sofa
2. Justify your idea by saying that splitting the cost of a sofa 3 ways makes it much cheaper for everyone.
3. Add to your justification that watching TV on the wooden kitchen chair is uncomfortable

<Read (out loud) the sentence openers in the Inform category >

Motivate

How would you (using the sentence opener interface)…
1. Encourage your roommate by saying he made a good point about sitting on the wooden chair
2. Motivate your roommate
3. Reinforce your roommate by telling him that he's right about the sofa

<Read (out loud) the sentence openers in the Motivate category >

Task

How would you (using the sentence opener interface)…
1. Summarize the location of all the major appliances in the kitchen
2. Tell your roommate that you will show her where the lava lamp will go
3. Tell your roommate that you want to move on to decorating the dining room

<Read (out loud) the sentence openers in the Task category >

Acknowledge

How would you (using the sentence opener interface)…

1. Acknowledge your roommate's comment
2. Thank your roommates for their help in carrying your bed
3. Answer your roommate's question, "Are you a vegetarian?"

<Read (out loud) the sentence openers in the Acknowledge category >

Discuss

How would you (using the sentence opener interface)…

1. Tell your roommates that the 3 of you need to consider making house rules
2. Agree with your roommate about the house rules
3. Tell your roommate that you aren't so sure about the house rules, since they will inevitably get broken

<Read (out loud) the sentence openers in the Discuss category >

Maintenance

How would you (using the sentence opener interface)…

1. Ask your roommate if he would please help you choose a new sofa
2. Tell your roommate that you hear what he is saying about the house rules.
3. Apologize for the mess you made in the kitchen

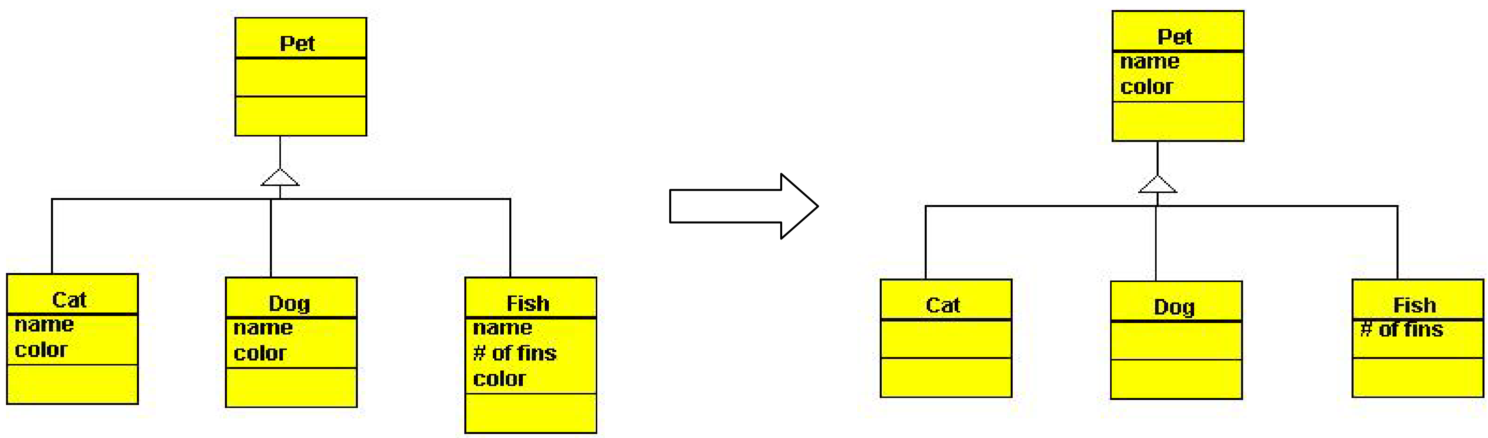
<Read (out loud) the sentence openers in the Maintenance category >

APPENDIX B

Example of Individual Knowledge Element: Generalization: Attribute Inheritance

Recall that *Generalization* is the ("type-of") relationship between a class and one or more refined versions of it. The class being refined in called the *superclass*, and each refined version is called a *subclass*. For example, *Pet* is the superclass of *Cat*, *Dog*, and *Fish*. **Attributes common to a group of subclasses should be attached to the superclass**. This allows them to be shared by each subclass. Each subclass is said to inherit the features of its superclass. For example (see diagram below), *Cat* inherits the attributes name and color from *Pet*.

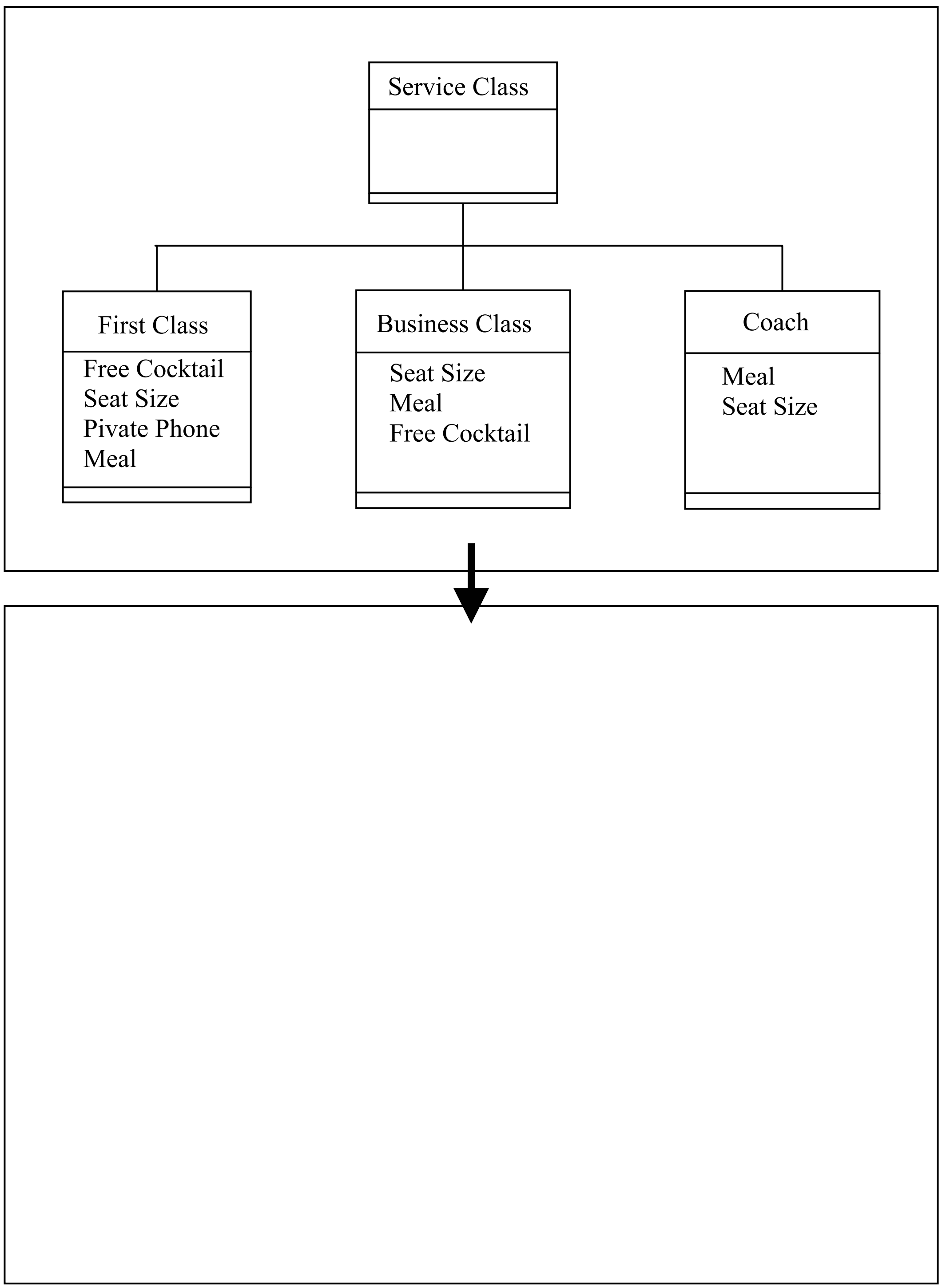
INCORRECT CORRECT



APPENDIX C

Example of Pre/Post Test Question

Elaborate or Correct the diagram shown below to capture the relationship between service classes, first class, business class, and coach.



APPENDIX D

Group Exercise and Solution

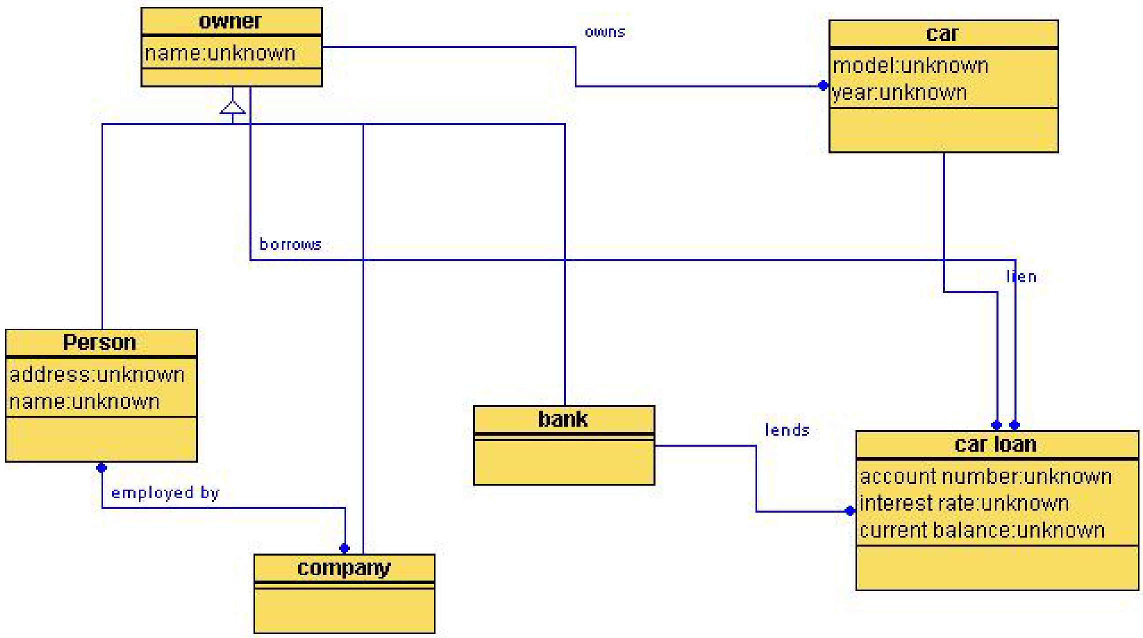
Shown below are five object classes: Person, Car, Car loan, Company, and Bank. These are some of the objects typically involved in the purchase of a car. Some of the object classes shown below have attributes that are really pointers to other object classes, and which should be replaced by associations.

A person may have several companies as employers. Each person has an ID. A car is assigned an ID. Cars may be owned by persons, companies, or banks. Car owner ID is the ID of the person, company, or bank who owns the car. A car loan may be involved in the purchase of a car.

Burying object references as pointers is the incorrect way to construct an object model. Prepare an object diagram in which the pointers are replaced with relationships. Try to get multiplicities right. You may need to add one or more objects of your own. Eliminate all IDs. At least one attribute should be converted to a discriminator.

Person	**Car**	**Car loan**	**Company**	**Bank**
name	owner ID	vehicle ID	company ID	bank ID
employer 1 ID	vehicle ID	customer type		
employer 2 ID	owner type	customer ID		
person ID	model	account number		
address	year	bank ID		
		interest rate		
		current balance		

Solution


owner
name:unknown
owns
car
model:unknown
year:unknown
borrows
lien
Person
address:unknown
name:unknown
bank
lends
car loan
account number:unknown
interest rate:unknown
current balance:unknown
employed by
company

APPENDIX E

Effective Knowledge Sharing Sequences
LRDC Episodes 1-7

LRDC 1 – KE 3
ITK - EFFECTIVE, Ilya is Knowledge Sharer

R1: General suggestion related to KE
S: Explain KE
S: Probe R
R2: Elaborate on Explanation
R1: Agree
S: Begin to illustrate KE
R1: Continue illustration
R1: Request confirmation on drawing change
S: Group maintenance
R2: Encouragement/Agreement

Tracey	Request	Opinion	Do you think we need the "owner type" in the Car object?	R1: General Suggestion related to KE
Ken	Acknowledge	Accept	Yes , to distinguish from bank or person or company	
Ilya	Inform	Assert	I'm reasonably sure that that's the discriminator	S: Explain KE
Ilya	Inform	Elaborate	To elaborate , we need an owner object that company, person and bank inherit from.	
Ilya	Maintenance	Apologize	Sorry , I mean owner "class" not object	
Tracey	Motivate	Encourage	Very Good ! Sounds right to me	
Ilya	Request	Opinion	Ken, Do you think that's right?	S: Probe R
Ken	Motivate	Reinforce	That's Right , so do we just create an object at the top and connect the company, bank, and person to it?	R2: Elaborate on Explanation
Ilya	Created new class $Owner$			
Tracey	Inform	Suggest	I think so	R1: Agree
Ken	Motivate	Encourage	Very Good	
Ilya	Added new association linking $Owner$ to $Bank$			S: Illustrate KE

Ilya	Mediate	Teacher mediation	Let's ask the teacher something, one second guys	
Ken	Acknowledge	Accept	OK	
Ilya	Added new association linking $Company$ to $Company$			
Ilya	Added new association linking $Owner$ to $Company$			
Ilya	Added new association linking $Owner$ to $Person$			
Ilya	Stopped Drawing			
Ilya	Inform	Lead	Perhaps we should also remove the owns lines and draw just one from owner to car	
Ken	Acknowledge	Accept	Yes , and is it possible to space things out a little to make it somewhat more legible?	
Tracey	Motivate	Encourage	Very Good	
Tracey	Started Drawing			
Ilya	Discuss	Agree	Yes, I agree	
Tracey	Deleted link connecting$Company$ to Car			R1: Continue illustration
Tracey	Deleted link connecting$Person$ to Car			
Tracey	Deleted link connectingCar to $Bank$			
Tracey	Added new association linking Car to $Owner$			
Tracey	Changed multiplicity of $Owner$ to $Many (0 or more)$ on association			

	linking Car to $Owner$			
Tracey	Changed multiplicity of $Owner$ to $1 or more$ on association linking Car to $Owner$			
Tracey	Changed multiplicity of Car to $Many (0 or more)$ on association linking Car to $Owner$			
Tracey	Stopped Drawing			
Tracey	Maintenance	Request Confirmati on	Is this OK?	R1: Request confirmation on drawing change
Ilya	Not classified	Not classified	Don't worry about the triangle next to company	
Ken	Motivate	Encourage	Very Good	
Tracey	Discuss	Doubt	I'm not so sure about the multiplicity	
Ilya	Acknowledge	Accept	Yes , looks great.	S: Group maintenance
Ilya	Task	Coordinate Group Activity	OK. Let's move on to verify the diagram	
Tracey	Inform	Suggest	I think we can delete owner type. Right?	
Ken	Motivate	Encourage	Yeah, we can now since we created "owner-	R2: Encouragement/Agree ment
Tracey	Acknowledge	Accept	OK	
Tracey	Started Drawing			
Ilya	Acknowledge	Accept	OK	
Tracey	Edited attributes for class Car - $model,year,$			
Tracey	Edited operations for class Car - $$			

LRDC 2 – KE 1
ITK - EFFECTIVE, Tracey is Knowledge Sharer

S: Suggest KE
S: Illustrate KE
R1: Motivate

Tracey	Inform	Suggest	I think we can put "name" in the Owner class now.	S: Suggest KE
Tracey	Edited attributes for class $Owner$ - $name$			S: Illustrate KE
Tracey	Edited operations for class $Owner$ - $$			
Tracey	Edited attributes for class $Company$ - $,$			
Tracey	Edited operations for class $Company$ - $$			
Tracey	Edited attributes for class $Person$ - $age,address,$			
Tracey	Edited operations for class $Person$ - $$			
Tracey	Edited attributes for class $Bank$ - $$			
Tracey	Edited operations for class $Bank$ - $$			
Ilya	Motivate	Encourage	Very Good	R1: Motivate

LRDC 3 – KE 1
ITK - EFFECTIVE, Ilya is Knowledge Sharer

R1: Request explanation of KE
S: Explain KE
R1: Agree
R2: Encourage/Motivate

Tracey	Request	Justification	Why do you think that "name" should be added ?	R1: Request explanation of KE
Ilya	Maintenance	Request Attention	Excuse Me	
Ilya	Maintenance	Request Action	Would you please remember that we don't need the name in company and bank, because they inherit name from Owner?	S: Explain KE
Ilya	Edited attributes for class $Company$ - $$			
Ilya	Edited operations for class $Company$ - $$			
Tracey	Discuss	Agree	Yes, I agree , Name should not be added	R1: Agree
Ken	Maintenance	Apologize	Sorry	
Ilya	Edited attributes for class $Bank$ - $,$			
Ilya	Edited operations for class $Bank$ - $$			
Ken	Motivate	Encourage	Good Point , I forgot about that.	R2: Encourage/Motivate

LRDC 4 – KE 3
GMJ - EFFECTIVE, Gregg is Knowledge Sharer

S: Suggest KE
R1: Request explanation
S: Explain KE
R1 & R2: Attempt rephrase using another example
S: Clarify & Re-explain
Some discussion, doubt, further explanation
R2: Request further clarification
(Further explanation by S & R1)
R2: Attempt Rephrase KE

Gregg	Inform	Assert	I'm reasonably sure it's all ok, but we are supposed to create a class from an attribute	S: Suggest KE
Jon	Maintenance	Apologize	Sorry, I'm not sure what you mean.	R1: Request explanation
Gregg	Inform	Explain/Clarify	Let me explain it this way - discriminator.	
Gregg	Request	Information	Do you know the last question on the pre-test.	
Jon	Discuss	Agree	Yes, I agree	
Jon	Request	Clarification	Can you explain how we can do that, though?	
Gregg	Inform	Suggest	I think we could take account from bank and create classes for types of accounts.	S: Explain KE
Gregg	Inform	Elaborate	Also we could separate something out from car.	
Mark	Inform	Suggest	I think separating from car would be better	
Jon	Discuss	Offer Alternative	Alternatively, we could say that making bank a subclass of company already created a discriminator, and got rid of Bank name and ID.	R1 & R2: Attempt rephrase using another example
Mark	Motivate	Encourage	Good Point	
Gregg	Discuss	Disagree	I disagree because we did not create more than one subclass	S: Clarify & Re-explain
Jon	Maintenance	Listening	I see what you're saying.	
Gregg	Discuss	Infer	Therefore I think we need to create a whole new class.	
Gregg	Inform	Elaborate	Also putting at least a couple examples below it	
Gregg	Request	Information	Do you know what attribute we should use?	
Jon	Discuss	Doubt	I'm not so sure. We've gotten rid of a lot of them already.	
Gregg	Inform	Suggest	I still think the account type is a good	

			one	
Jon	Maintenance	Request Action	Would you please list some examples?	
Gregg	Request	Information	Mark, Do you know what we would use from car?	
Mark	Inform	Suggest	I think model could be broken out	
Jon	Discuss	Doubt	I'm not so sure that's a good idea. I think it will muddy up the diagram.	
Gregg	Discuss	Doubt	I'm not so sure what we should use. It all seems like it would just cause more confusion	
Gregg	Discuss	Propose Exception	But we need to consider the instructions	
Mark	Discuss	Agree	Yes, I agree	
Jon	Discuss	Agree	Yes, I agree - I think that the creation of one discriminator is good, though. We got rid of two attributes and one line, and left things clean.	
Gregg	Maintenance	Request Action	Would you please tell me when we created one?	
Jon	Acknowledge	Accept	Yes - we made bank a subclass of company.	
Gregg	Discuss	Disagree	I disagree because we did not take an attribute and create a class.	
Jon	Maintenance	Listening	I see what you're saying - perhaps we should do the account thing, then.	
Gregg	Request	Opinion	Mark, what do you think	
Mark	Request	Information	Do you know if creating a class is what they mean by converting to a discriminator?	R2: Request Clarification
Gregg	Inform	Assert	I'm reasonably sure	
Jon	Discuss	Doubt	I'm not so sure that it is. I think that it means making a new association, instead.	
Jon	Inform	Elaborate	To elaborate, in the earlier part, we got rid of the "name" and "color" thing on the pets, and made "cat", "dog" and "fish" subclasses of pets.	
Gregg	Discuss	Disagree	I disagree because the question 3 explicitly asked fot the creation of a new class from an attrubute	
Jon	Inform	Suggest	I think we have a good diagram, and no time left. Let's leave it as is rather than trying to create a large new addition.	
Gregg	Task	Coordinate Group Activity	whatever	
Jon	Inform	Elaborate	Also, we'll just screw something up in the short time left...	
Gregg	Acknowledge	Accept	OK	

Gregg	Acknowledge	Accept	OK	
Gregg	Acknowledge	Accept	OK	
Jon	Maintenance	Apologize	Sorry	
Mark	Inform	Suggest	I think we may have accomplished the attribute thing with something we already did, like taking Bank ID and and saying bank issues a car loan	R2: Attempt Rephrase KE
Gregg	Discuss	Agree	I agree because we never created a new class.	
Jon	Discuss	Agree	{Re Line 160} Yes, I agree	
Jon	Task	End participation	Goodbye	
Mark	Task	Summarize Information	see ya	

LRDC 5 – KE 3
IKH - EFFECTIVE, Ken is Knowledge Sharer

R1: Suggest discussion related to KE
S: Illustrate KE
R1: Acknowledge/Accept
S: Explain KE using diagram
R2: Recommendation related to KE (regarding another example, to further understanding)
S: Elaborate on KE
R1 & R2: Agreement

Helen	Inform	Elaborate	Also , my instructions say "at least one attribute should be converted to a discriminator."	R1: Suggest discussion related to KE
Helen	Request	Opinion	Do you think we have created a discriminator? I'm not sure what is meant by that.	
Ken	Inform	Suggest	I think that I have the information for that in my pretest. Let me look over it and see what I can come up with.	
Ken	Task	Propose Illustration	Let me show you what I believe a good discriminator should look like. Check out the pictures above. Give me a minute to finish.	
Ken	Created new class $Owner$			S: Illustrate KE
Ken	Deleted link connecting$Car loan$ to Car			
Ken	Deleted link connectingCar to $Company$			
Ken	Added new association linking $Car loan$ to $Owner$			
Helen	Acknowledge	Accept	OK	R1: Acknowledge/Accept
Ken	Added new association linking $Company$ to $Owner$			
Ken	Added new association linking $Owner$ to Car			
Ken	Changed multiplicity of			

	Car to $Many (0 or more)$ on association linking $Owner$ to Car			
Ken	Acknowledge	Accept	Yes check out what I did above. A discriminator takes out a quality, such as "owner type" or "owner ID" and basically makes that the lone requisite for a relation with that box.	S: Explain KE using diagram
Helen	Maintenance	Listening	I see what you're saying, looks good. Are we done?	
Ken	Inform	Suggest	I think the subclassification probihits other relations. Something like that.	
Igor	Inform	Suggest	I think a discriminator "owner" should be between person, bank and company.	R2: Recommendation related to KE (regarding another example, to further understanding)
Ken	Inform	Assert	I'm reasonably sure that's again included because bank is a subcategory of company. No change is necessary.	S: Elaborate on KE
Helen	Discuss	Agree	Yes, I agree with Ken	R1 & R2: Agreement
Ken	Maintenance	Request Confirmation	Is this OK?	
Helen	Acknowledge	Accept	Yes	
Ken	Maintenance	Request Confirmation	Igor, what do you think?	
Igor	Inform	Suggest	I think , you are right	

LRDC 6 – KE 3
ARS - EFFECTIVE, Rikin is Knowledge Sharer

R1 & R2: Request info related to KE
S: Explain KE
R1: Agree

Salil	Request	Information	Do you know what a discriminator is?	R1 & R2: Request info related to KE
Anima	Created new class $Car Loan$			
Rikin	Acknowledge	Accept	Yes	
Salil	Acknowledge	Accept	Yes	
Anima	Stopped Drawing			
Anima	Acknowledge	Reject	Anima, No i don't know what discriminator is	
Rikin	Inform	Suggest	I think the vehicle class should be an "owner type" link to the 3 buyers	S: Explain KE
Rikin	Maintenance	Request Confirmation	Right?	
Anima	Discuss	Agree	Yes, I agree	R1: Agree

LRDC 7 – KE 2
ARS - EFFECTIVE, Salil is Knowledge Sharer

R1: Suggest KE
S: Agree/Accept
R2: Motivate
S: Motivate

Anima	Inform	Suggest	I think we can eliminate all the IDs because reelationships show that	R1: Suggest KE
Salil	Acknowledge	Accept	Yes you are right anima	S: Agree/Accept
Rikin	Motivate	Reinforce	That's Right Anima I think they are eliminated just by the connections	
Anima	Motivate	Encourage	Good Point	R2: Motivate
Salil	Discuss	Agree	I agree that we need to eliminate all ID's	S: Motivate

Ineffective Knowledge Sharing Sequences
LRDC Episodes 8-18

LRDC 8 – KE 3
JAK - INEFFECTIVE, Jin is Knowledge Sharer

S: Suggest KE
R1: Doubt suggestion
R1 & R2: Request elaboration of KE
S: Explain KE unsatisfactorily
(no further discussion of KE)

Jin	Request	Opinion	Do you think we need a discriminator for the car ownership	S: Suggest KE
Kim	Created new class $Person$			
Kim	Created new class Car			
Kim	Discuss	Doubt	I'm not so sure	R1: Doubt suggestion
Kim	Created new class $Car Loan$			
Kim	Started Drawing			
Kim	Created new class $Person$			
Kim	Created new class Car			
Kim	Created new class $Company$			
Jin	Created new class $bank$			
Jin	Created new class $bank$			
Kim	Created new class $Bank$			
Andy	Request	Elaboration	Can you tell me more about what a discriminator is	R2: Request elaboration of KE
Jin	Created new class $person$			
Kim	Discuss	Agree	Yes, I agree because I myself am not so sure as to what its function is	R1: Request elaboration of KE
Andy	Edited attributes for class $Person$ - $Name$			
Andy	Edited operations for class $Person$ - $$			
Kim	Edited attributes for class $Person$ - $name,age,employer1			

	ID,employer2 ID,person ID,address$			
Kim	Edited operations for class $Person$ - $$			
Andy	Edited attributes for class $Company$ - $Name$			
Andy	Edited operations for class $Company$ - $$			
Andy	Edited attributes for class $Bank$ - $Name$			
Andy	Edited operations for class $Bank$ - $$			
Kim	Edited attributes for class Car - $owner Id,vehicle type,owner type,model,year$			
Kim	Edited operations for class Car - $$			
Jin	Inform	Explain/ Clarify	Let me explain it this way - A car can be owned by a person , a company or a bank. I thiink ownership type is the discrinator.	S: Explain KE unsatisfactorily (no further discussion of KE)
Jin	Maintenance	Apologize	Sorry I mean discriminator.	

LRDC 9 – KE 3
JAK - INEFFECTIVE, Jin is Knowledge Sharer

S: Suggest KE
R1: Recommend that group ignore KE
S: Refute recommendation
R2: Request explanation of KE
S: Explain KE unsatisfactorily
(no further discussion of KE)

Jin	Maintenance	Request Action	Would you please think about the discriminator that the instruction mentions?	S: Suggest KE
Andy	Request	Opinion	Do you think any thing would be lost if we deleted it?	R1: Recommend that group ignore KE
Jin	Inform	Suggest	I think we are doing good job. We don't need delete anything.	S: Refute recommendation
Kim	Maintenance	Listening	I see what you're saying but I am stil not clear on the ownshiptype.	R2: Request explanation of KE
Kim	Request	Opinion	Do you think we can finish this in 2 minutes?	
Jin	Inform	Explain/Clarify	Let me explain it this way that ownership type is the discriminator.	S: Explain KE unsatisfactorily (no further discussion of KE)
Kim	Changed multiplicity of $Person$ to $Many (0 or more)$ on association linking $Car Loan$ to $Person$			
Kim	Changed multiplicity of $Person$ to $Many (0 or more)$ on association linking $Car Loan$ to $Person$			
Andy	Acknowledge	Accept	OK	

LRDC 10 – KE 2
GMJ - INEFFECTIVE, Jon is Knowledge Sharer

S: Suggest KE
R1: Doubt suggestion
R1: Attempt rephrase KE
R2: Suggest alternative
R1: Agree with R2
S: Explain KE unsatisfactorily
(no further discussion of KE)

Jon	Request	Opinion	Do you think we can assume that the person or company is linked to the loan _through_ the car class?	S: Suggest KE
Gregg	Inform	Suggest	{Re Line 19} I think you are saying two different things.	R1: Doubt suggestion
Jon	Maintenance	Request Action	Would you please elaborate? I'm not sure I understand you.	
Gregg	Inform	Suggest	I think you are saying to not put in pointers for car loan, but that we must have a pointer to determine who owns the car.	R1: Attempt rephrase KE
Mark	Inform	Suggest	I think the car loan should be linked through bank	R2: Suggest alternative
Gregg	Discuss	Agree	Yes, I agree	R1: Agree with R2
Gregg	Inform	Elaborate	Also, we are supposed to be getting rid of all IDs	
Jon	Inform	Elaborate	To elaborate, I'm using the term pointer in the database sense, ie, an attribute that shows linkage, rather than a line.	S: Explain KE unsatisfactorily (no further discussion of KE)
Jon	Stopped Drawing			
Gregg	Acknowledge	Accept	OK	

LRDC 11 – KE 2
GMJ - INEFFECTIVE, Jon is Knowledge Sharer

R1: Propose discussion related to KE
S: Suggest that discussion is not needed (KE done already)
R1: Express confusion
S: Reaffirm that the problem is taken care of
R1: Apologize

Gregg	Discuss	Propose Exception	But we need to consider the employer attribute in the person class	R1: Propose discussion related to KE
Gregg	Request	Opinion	{Re Line 55} Do you think?	
Jon	Inform	Suggest	{Re Line 64} I think that's covered by the link to the company.	S: Suggest that discussion is not needed (KE done already)
Gregg	Inform	Assert	I'm reasonably sure that a company object will not explain who a person works for.	R1: Express confusion
Jon	Inform	Suggest	{Re Line 67} I think that the "works for" label will handle it.	S: Reaffirm that the problem is taken care of
Jon	Maintenance	Apologize	Sorry, I can see that it's covered up. It says "works for" next to company, on the line to person.	
Gregg	Maintenance	Apologize	Sorry, I didn't realize that was there.	R1: Apologize
Jon	Acknowledge	Appreciation	Thank you.	

LRDC 12 – KE 3
ARS - INEFFECTIVE, Rikin is Knowledge Sharer

S: Recommend KE
R1: Doubt that it is needed
S: Explain KE unsatisfactorily
R1: Doubt again
S: Request help from R2

Rikin	Request	Opinion	Do you think I should make account number a separate class?	S: Recommend KE
Rikin	Stopped Drawing			
Anima	Discuss	Doubt	I'm not so sure why you'd wnat to do that	R1: Doubt that it is needed
Rikin	Inform	Justify	To justify its a way of discriminating since both person and company have account numbers	S: Explain KE unsatisfactorily
Anima	Acknowledge	Accept	Yes but each has seperate relationship also	R1: Doubt again
Rikin	Maintenance	Request Action	Salil what do you think since you know about discriminating as well?	S: Request help from R2

LRDC 13 – KE 2
ARS - INEFFECTIVE, Salil is Knowledge Sharer

R1: Request explanation of KE
S: Explain KE briefly
R1 & R2: Determine that KE is already done
S: Recommend group reconsider KE
R1 & R2: Acknowledge
(no further discussion of KE)

Rikin	Request	Information	Do you know what the paper means when it says "Burying object references as pointers is the incorrect way to construct an object model"?	R1: Request explanation of KE
Salil	Acknowledge	Accept	Yes burying means to hide the information inside the class	S: Explain KE briefly
Salil	Discuss	Propose Exception	Rikin, But we have done that job	R1 & R2: Determine that KE is already done
Anima	Discuss	Agree	I agree because the job is finished	
Salil	Inform	Justify	Rikin, To justify we have made the relationships right and that takes care of the entire burying business	S: Recommend group reconsider KE
Rikin	Motivate	Reinforce	That's Right I think Salil	R1 & R2: Acknowledge (no further discussion of KE)
Anima	Inform	Suggest	I think rikin is right salil	

LRDC 14 – KE 3
ARS - INEFFECTIVE, Rikin is Knowledge Sharer

S: Recommend KE
R1: Request explanation of KE
S: Illustrate KE (incorrectly) without explanation
R1 & R2: Acknowledge

Rikin	Inform	Suggest	I think we have to create a discriminator according to the agenda	S: Recommend KE
Rikin	Request	Opinion	Do you think we have?	
Anima	Request	Information	Do you know if we created create discriminator??	
Salil	Discuss	Doubt	Rikin, I'm not so sure what a discriminator is do you know rikin?	R1: Request explanation of KE
Rikin	Created new class $Account Number$			
Anima	Inform	Suggest	I think its too late to worry about it now...	
Salil	Request	Information	Anima, Do you know what that discriminator is	
Rikin	Task	Propose Illustration	Let me show you	S: Illustrate KE (incorrectly) without explanation
Salil	Acknowledge	Accept	Anima, Yes its late now lets get the agenda checked	
Salil	Acknowledge	Accept	OK	
Salil	Acknowledge	Accept	OK	
Anima	Acknowledge	Appreciation	Thank you	
Rikin	Edited attributes for class $Car Loan$ - $Interest Rate,Current Balance,$			
Rikin	Edited operations for class $Car Loan$ - $$			
Anima	Acknowledge	Appreciation	Rikin, Thank you	
Rikin	Added new association linking $Account Number$ to $Car Loan$			
Anima	Maintenance	Listening	I see what you're saying	
Rikin	Added new association linking $Account			

	Number$ to $Person$			
Rikin	Added new association linking $Person$ to $Person$			
Rikin	Added new association linking $Account Number$ to $Company$			
Salil	Acknowledge	Accept	OK	
Anima	Acknowledge	Appreciation	Thank you rikin and salil....	R1 & R2: Acknowledge
Salil	Acknowledge	Appreciation	Rikin, Thank you	
Rikin	Maintenance	Request Confirmation	Is this OK?	
Salil	Acknowledge	Appreciation	Anima, Thank you	
Anima	Inform	Elaborate	Also , diagram looks fine...	
Rikin	Changed multiplicity of $Account Number$ to $Optional (0 or 1)$ on association linking $Account Number$ to $Person$			
Salil	Acknowledge	Accept	Yes now let that be as it is , its time up	
Rikin	Changed multiplicity of $Account Number$ to $Optional (0 or 1)$ on association linking $Account Number$ to $Company$			
Anima	Task	End participation	Goodbye	
Salil	Motivate	Encourage	Rikin, Very Good you did a great job	
Rikin	Task	End participation	Goodbye	
Salil	Motivate	Encourage	Rikin, Very Good you were fastt	

LRDC 15 – KE 3
ASJ - INEFFECTIVE, Sheldon is Knowledge Sharer

R1 & R2: Request discussion about KE
S: Suggest that the KE is already done
S: Offer brief explanation
(no further discussion of KE)

Jim	Inform	Suggest	I think we need a descriminator	R1 & R2: Request discussion about KE
Alex	Discuss	Agree	{Re Line 358} Yes, I agree	
Alex	Request	Clarification	Can you explain why/how	
Alex	Request	Illustration	Please show me	
Sheldon	Inform	Suggest	I think it was car owner id	S: Suggest that the KE is already done
Sheldon	Inform	Suggest	I think it links to car	
Jim	Acknowledge	Accept	Yes , i agree	
Sheldon	Inform	Elaborate	To elaborate car owner id..is like that example we did b4	S: Offer brief explanation (no further discussion of KE)
Sheldon	Inform	Suggest	I think person, bank, and company follows car owner ID	
Sheldon	Maintenance	Apologize	Sorry I see it now	

LRDC 16 – KE 2
ASJ - INEFFECTIVE, Alex is Knowledge Sharer

R1: Propose discussion related to KE
S: Explain KE briefly
R1: Acknowledge

Sheldon	Inform	Suggest	I think car owner ID	R1: Propose discussion related to KE
Alex	Discuss	Disagree	I disagree because	
Jim	Changed multiplicity of Car to $Many (0 or more)$ on association linking $Company$ to Car			
Alex	Discuss	Disagree	I disagree because all IDs are pointers and should now be eliminated because we use links	S: Explain KE briefly
Sheldon	Motivate	Encourage	Very Good	R1: Acknowledge
Alex	Edited attributes for class $Car Loan$ - $customer type,account number,interest rate,current balance$			
Alex	Edited operations for class $Car Loan$ - $$			

LRDC 17 – KE 3
KKN - INEFFECTIVE, Kirstin is Knowledge Sharer

S: Recommend KE
R1: Acknowledge
R2: Request explanation of KE
S: Explain KE unsatisfactorily
(no further discussion of KE)

Kirstin	Request	Opinion	Do you think that customer type and owner type might become	S: Recommend KE
Kyle	Changed multiplicity of $Company$ to $Many (0 or more)$ on association linking $Company$ to $Person$			
Kirstin	Inform	Suggest	I think descriminators	
Kyle	Changed multiplicity of $Person$ to $1 or more$ on association linking $Company$ to $Person$			
Kirstin	Inform	Suggest	I think what do you think?	
Kyle	Acknowledge	Accept	Yes	R1: Acknowledge
Natalia	Maintenance	Request Action	Would you please explain to mewhat the "discriminator" is? I missed it...	R2: Request explanation of KE
Kirstin	Inform	Justify	To justify descriminators represent an is a or has a relationship	S: Explain KE unsatisfactorily (no further discussion of KE)
Kyle	Stopped Drawing			
Kirstin	Inform	Justify	To justify so they become a class	

LRDC 18 – KE 3
KKN - INEFFECTIVE, Kirstin is Knowledge Sharer

S: Explain KE
R1: Acknowledge
S: Illustrate KE without explanation

Kirstin	Acknowledge	Accept	Yes Now I link company, person, bank to customer type
Kirstin	Inform	Justify	To justify that makes it a descriminator
Kyle	Acknowledge	Accept	OK
Kirstin	Changed name of association linking $Company$ to $Car Loan$ to $gets a loan from$		
Kirstin	Inform	Justify	To justify
Kirstin	Deleted link connecting$Company$ to Car		
Kirstin	Added new association linking $Customer Type$ to Car		
Kirstin	Changed name of association linking $Customer Type$ to Car to $owns a$		

Effective Knowledge Sharing Sequences
MITRE Episodes[13] 1-3

MITRE 1 – KE 3
ADC - EFFECTIVE, Carol is Knowledge Sharer

S: Propose KE
R1: Request explanation
S: Illustrate KE
R1: Question/Debate self
R2: Reinforce Sharer's actions
R1 & R2: Explain actions to self
R1 & R2: Agree

Camille	Discuss	Propose Exception	But we need to consider we need a discriminator.	S: Propose KE
Alan	Inform	Suggest	we need to convert at least one attribute to a discriminator.... why owner type?	R1: Request explanation
Camille	Acknowledge	Accept	Yes do you see what I'm doing?	S: Illustrate KE
Alan	Acknowledge	Reject	No	
Camille	Added new association linking Car to $owner type$			
Alan	Inform	Elaborate	To elaborate, i don't see why owner type should be a disriminator	
Camille	Added new ingeritance linking $owner type$ to $bank$			
Camille	Added new inheritance linking $owner type$ to $company$			
Alan	Inform	Suggest	I think an owner type does not own a car	R1: Question/ Debate self
Alan	Inform	Suggest	I think a person owns a car	
Camille	Discuss	Agree	Yes, I agree it should be owner	
Alan	Discuss	Disagree	I disagree because a person owns a car	
Alan	Maintenance	Apologize	Sorry	
Alan	Maintenance	Apologize	Sorry, i forgot that other things can own cars too	
Camille	Maintenance	Request	Is this OK? if we change owner	

[13] The names of subjects have been changed.

		Confirmation	type to owner	
Camille	Added new association linking $owner type$ to $owner type$			
Dave	Motivate	Reinforce	That's Right	R2: Reinforce Sharer's actions
Alan	Maintenance	Apologize	the directiosn indicate, "cars can be owned by persons, companies, or banks"	
Alan	Inform	Suggest	I think that we don't need a class owner, and if we do, then person, company, and bankm should inherit from it	R1 & R2: Explain actions to self
Dave	Acknowledge	Accept	Yes . From class owner we need 3 lines, to bank, company, person	
Camille	Discuss	Agree	Yes, I agree do you see that is what I am doing?	
Camille	Changed name of class $owner type$ to $owner$			
Camille	Added new inheritance linking $owner$ to $Person$			
Dave	Acknowledge	Accept	Yes	R1 & R2: Agree
Alan	Inform	Suggest	I think i see what you are doing, but i'm not sure why you're doing it... although it's not important now	
Camille	Inform	Suggest	I think I accidentally hit the inherit button on car too	
Dave	Acknowledge	Accept	Yes	
Alan	Deleted link connectingCar to $owner$			
Alan	Added new inheritance linking $owner$ to Car			

MITRE 2 – KE 3
SMJ - EFFECTIVE, Sarah is Knowledge Sharer

S: Suggest KE
R1: Request explanation of KE
S: Explain KE
R1: Request further clarification of KE (using another example)
S: Clarify
R2: Agree

Sally	Inform	Assert	I'm reasonably sure we could have person and company as discriminators for car owner	S: Suggest KE
Mary	Started Drawing			
Mary	Stopped Drawing			
Mary	Request	Clarification	Can you explain what a discriminator is again (Sorry)?	R1: Request explanation of KE
Sally	Inform	Explain/Clarify	Let me explain it this way , a discriminator is an attribute represented as a class	S: Explain KE
Sally	Inform	Elaborate	To elaborate , the last problem on the pretest was about attributes and discriminators	
Mary	Request	Opinion	Do you think we should then have an id type for "Car owner" to differentiate between person/co./bank?	R1: Request further clarification of KE (using another example)
Sally	Discuss	Disagree	I disagree because if you read the sheet, we are to eliminate all IDs	S: Clarify
Jeb	Discuss	Agree	I agree because it could be helpful to have a serial number on a car	R2: Agree
Sally	Inform	Justify	To justify if you read the last paragraph, 2nd sentance from the end	

MITRE 3 – KE 2
ERP - EFFECTIVE, Eli is Knowledge Sharer

S: Recommend KE
R1: Request information related to KE
S: Explain KE
R2: Rephrase KE
R1: Acknowledge/Accept
R2: Illustrate KE

Ernie	Inform	Suggest	I think we should do agenda "Replace pointers with associations"	S: Recommend KE
Patrick	Discuss	Agree	Yes, I agree , but either a company needs to have a bunch of employees, or visa versa	
Ernie	Discuss	Infer	Therefore we can get rid of attributes that were pointers, but are now associations	
Rachel	Inform	Justify	{Re Line 211} To justify that is taken care of by the multiplicities (the circles)	
Patrick	Inform	Elaborate	{Re Line 214} To elaborate its not a pointer, its a vector of person classes.	
Ernie	Request	Clarification	Can you explain why {Re line 214} what we have now isn't right?	
Patrick	Request	Opinion	Do you think we shouldn't have any attributes that refer to classes then?	R1: Request information related to KE
Rachel	Inform	Suggest	I think that's right	
Ernie	Inform	Suggest	I think {Re Line 219} we shouldn't have any attributes that...	S: Explain KE
Rachel	Inform	Explain/Clarify	Let me explain it this way the links, or associations take care of the references	R2: Rephrase KE
Ernie	Edited attributes for class $Company$ - $,$			
Ernie	Edited operations for			

	class $Company$ - $$			
Ernie	Stopped Drawing			
Patrick	Acknowledge	Accept	OK	R1: Acknowledge/Accept
Rachel	Task	Propose Illustration	Let me show you	R2: Illustrate KE
Rachel	Started Drawing			
Ernie	Acknowledge	Accept	Yes , that what i thought.	
Rachel	Edited attributes for class $Company$ - $,,$			
Rachel	Edited operations for class $Company$ - $$			
Rachel	Edited attributes for class $Bank$ - $$			
Rachel	Edited operations for class $Bank$ - $$			
Ernie	Request	Illustration	Please show me the car owner class w/o the name attribute	
Rachel	Edited attributes for class $Car Loan$ - $account number,interest rate,current balance,$			

Ineffective Knowledge Sharing Sequences
MITRE Episodes 4-11

MITRE 4 – KE 1
ARK - INEFFECTIVE, Alan is Knowledge Sharer

R1: Request explanation of KE
S: Explain KE
R2: Further discussion/questioning about KE
S: Elaborate on KE, & finalize decision

Kris	Request	Clarification	Andy, Can you explain why/how organization is related here?	R1: Request explanation of KE
Andy	Inform	Elaborate	To elaborate , an organization is a "generalization " of Person, Company, and Bank. Name	S: Explain KE
Andy	Inform	Elaborate	To elaborate , "Name" should be part of the "organization" superclass.	
Andy	Inform	Explain/ Clarify	Let me explain it this way , with a link between car and organiztion, we can be explicit that a single organization owns 0 to many cars (see diagram)	
Andy	Inform	Elaborate	To elaborate , one organization (once we add a discriminator) is exactly one of a person, company, or bank.	
Ron	Maintenance	Listening	I see what you're saying and that implies we accept all as car owning classes. Is that an exhaustive list of car owners? Does it need to be?	R2: Further discussion and questioning about KE
Andy	Inform	Suggest	I think it is an exhaustive list within the scope of the stated problem	
Ron	Inform	Elaborate	To elaborate an automobile manufacturer owns cars and the dealer often borrows to get the cars he sells, etc.	
Andy	Discuss	Agree	Yes, I agree , but our problems states "Cars my be owned by persons, companies or banks".	S: Elaborate on KE, & finalize decision

MITRE 5 – KE 2
RAK - INEFFECTIVE, Russ is Knowledge Sharer

S: Propose KE
R1: Agree
R2: Illustrate KE
R2: Request confirmation of actions

Ron	Inform	Suggest	I think customer type is actually a relation to the borrowing organization oin our example.	S: Propose KE
Andy	Discuss	Agree	Yes, I agree.	R1: Agree
Kris	Edited attributes for class $Car loan$ - $account number,interest rate,current bAndyce,$			R2: Illustrate KE
Kris	Edited operations for class $Car loan$ - $$			
Kris	Edited attributes for class $Organization$ - $customer type$			
Kris	Edited operations for class $Organization$ - $$			
Kris	Stopped Drawing			
Kris	Inform	Suggest	I think I changed the attributes as you described them is that true so far?	R2: Request confirmation of actions

MITRE 6 – KE 1
RAK - INEFFECTIVE, Andy is Knowledge Sharer

S: Suggest KE
R1: Acknowledge
R1: Illustrate KE
S: Reflect on actions

Andy	Inform	Suggest	I think Kris should move "name" into organization	S: Suggest KE
Kris	Acknowledge	Accept	OK	R1: Acknowledge
Kris	Started Drawing			
Kris	Edited attributes for class $Person$ - $age,address,$			R1: Illustrate KE
Kris	Edited operations for class $Person$ - $$			
Kris	Edited attributes for class $Organization$ - $customer type,name$			
Kris	Edited operations for class $Organization$ - $$			
Andy	Task	Summarize Information	To summarize , do we agree we have the correct object classes?	S: Reflect on actions

MITRE 7 – KE 2
RAK - INEFFECTIVE, Ron is Knowledge Sharer

S: Suggest KE
R1: Request explanation
S: Explain KE (poorly)
S: Probe Rs
R1 & R2: Doubt suggestion
Small debate (S: Rebuttal, R1: Suggest Alternative)
S: Conciliate

Ron	Inform	Suggest	I think "owner type" under car is defined by relation to organization or whatever we might rename it.	S: Suggest KE
Andy	Mainten ance	Request Action	Would you please make a suggestion?	R1: Request explanation
Ron	Inform	Explain/Clari fy	Let me explain it this way. There are both buying and selling entities reprtesented in a car sale. We have focused on buyers and borrowers. Organization as we have shown it is really "Buyer". I think we do not have the "Seller" repesented in this, yet.	S: Explain KE (poorly)
Ron	Request	Opinion	Do you think Organization needs customer type since that is reflected in the relationship to the subclasses?	S: Probe Rs
Andy	Inform	Suggest	I think you are expanding the scope of the problem based on your (correct) view of the world outside the problem. The problem does not address buying and selling cars.	R1 & R2: Doubt suggestion
Kris	Discuss	Doubt	I'm not so sure , the exercise does say that "a car loan may be involved in the purchase o a car"	
Ron	Discuss	Doubt	I'm not so sure that is true. My sheet says "these are some of the objects involved in the purchase of a car."	
Andy	Inform	Suggest	I think the data model for this problem addresses the "current" owner of the car. Changing owners from buyer to seller is a simple matter of changing the data.	
Ron	Task	Summarize Information	To summarize It is a two sided deal buyer and seller and a loan may or not be involved and it could be any of several types of buyers. Employment relationships are less relevant, but I will acquiesce to there relationship.	

MITRE 8 – KE 1
RAK - INEFFECTIVE, Andy is Knowledge Sharer

S: Suggest R1 take action related to KE (no explanation)
R1: Follows instructions from S

Andy	Inform	Suggest	I think Kris should remove "name" from Company and Bank.	S: Suggest R1 take action related to KE (no explanation)
Andy	Inform	Suggest	I think Kris should rename organization as Owner	
Kris	Edited attributes for class \$Company\$ - \$,\$			R1: Follows instructions from S
Kris	Edited operations for class \$Company\$ - \$\$			
Kris	Edited attributes for class \$Bank\$ - \$,\$			
Kris	Edited operations for class \$Bank\$ - \$\$			

MITRE 9 – KE 2
RAK - INEFFECTIVE, Alan is Knowledge Sharer

S: Recommend KE
R1: Follows instructions from S
R2: Request verification of solution
S: Provide verification
R1: Apologize
R2: Acknowledge

Andy	Inform	Suggest	I think we should remove "owner type" and "customer type". I agree with Ron.	S: Recommend KE
Kris	Edited attributes for class $Owner$ - $name,$			R1: Follows instructions from S
Kris	Edited operations for class $Owner$ - $$			
Kris	Edited attributes for class car - $model,year,,$			
Kris	Edited operations for class car - $$			
Ron	Request	Opinion	Do you think model is accurate? I am not sure. How do you both feel?	R2: Request verification of solution
Kris	Inform	Suggest	I think model is a valid attribute of a car. Does that answer your question?	
Andy	Inform	Assert	I'm reasonably sure that Ron was referring to the object model, not the model attriubtue of car.	
Andy	Inform	Suggest	I think the model is reasonably accurate. I think we should check multiplicities.	S:Provide verification
Kris	Maintenance	Apologize	Sorry	R1: Apologize
Ron	Acknowledge	Accept	Yes Imeant object model.	R2: Acknowledge

MITRE 10 – KE 3
RAK - INEFFECTIVE, Kris is Knowledge Sharer

R1: Request explanation of KE
S: Suggest that KE is already done
(no further discussion on KE)

Andy	Request	Illustration	Please show me how to add a discriminator to owner. An owner is a single person, company, or bank.	R1: Request explanation of KE
Andy	Acknowledge	Accept	Yes , we only have a few minutes left, but I think we only need to add a discriminatior for owner.	
Kris	Inform	Suggest	I think the discriminator is set up OK	S: Suggest that KE is already done (no further discussion on KE)
Kris	Inform	Assert	I'm reasonably sure it looks like the one on my Individual knowledge Element	

MITRE 11 – KE 2
ADC - INEFFECTIVE, Dan is Knowledge Sharer

R1: Request explanation of KE
S: Explain KE (poorly)
R1: Request elaboration
(elaboration never provided)

Alan	Inform	Suggest	I think i wonder what "pointers" are (sorry to slow you guys down so much)	R1: Request explanation of KE
Alan	Inform	Suggest	I think we should decide what all the pointers are on our exercise sheet	
Alan	Inform	Suggest	I think we should do this, because need to replace the pointers with relationships	
Dave	Inform	Suggest	Dave, I think a ptr is just an attribute as opposed to a relationship	S: Explain KE (poorly)
Dave	Inform	Elaborate	To elaborate , we will convert some attribs into relationships	
Alan	Inform	Suggest	I think if that were so, we could not have any attributes in our final diagram!	R1: Request elaboration (elaboration never provided)
Camille	Motivate	Reinforce	That's Right Dave	

APPENDIX F

Questionnaire

Related experiences

1. Briefly describe any courses you have had, formal or informal, where collaboration was an important aspect of the course.
2. Briefly describe any courses you have had, formal or informal, in Object Modeling Technique (OMT) or a related subject.
3. Briefly characterize your experience with Chat tools.
4. Briefly characterize your experience with Palette-based tools similar to the OMT editor.

The Comet software

1. Did you notice any problems with the functionality of the software?
2. Please comment on the usability of the software. Was it easy to use? What were its best & worst features? Any suggestions for improvement?
4. Please characterize your degree of engagement during the exercises. I was:

   Not at all engaged Totally engaged

   -3 -2 -1 0 1 2 3
5. Please characterize your degree of control during the exercises. I had: _____

   No control Full control

   -3 -2 -1 0 1 2 3
6. Please characterize your attitude towards Chat tools. I: _____

   Dislike Chat Like Chat

   -3 -2 -1 0 1 2 3
7. Please characterize your learning of OMT during these exercises. I _____

   Learned no OMT Learned a lot about OMT

   -3 -2 -1 0 1 2 3
8. How do you feel about using tools like these in a formal, collaborative, course? I_____
   would not want to use this sort of tool would like to use these tools in a course

   -3 -2 -1 0 1 2 3

The sentence opener interface

1. Given the set of sentence openers provided, it was ________ to find the opener for what I wanted to say.

   Very difficult Very easy

   -3 -2 -1 0 1 2 3

2. In general, please characterize your ability to express yourself using the provided sentence openers. I____

   Could not easily express myself Could easily express myself

   -3 -2 -1 0 1 2 3

3. The set of sentence openers provided contained ________ the openers I needed.

   None All

   0 1 2 3 4 5 6

APPENDIX G

An Extracted Knowledge Sharing Episode

Time	Student	Subskill/Action	Attribute	Textual Contribution
20:35:40	Sheldon	Inform	Elaborate	To elaborate the link between car loan and car...shouldn't be there
20:35:47	Jim	Inform	Suggest	I think I think
20:35:59	Jim	Task	Summarize Information	To summarize , I think
20:36:12	Alex	Changed multiplicity of $Company$ to $Many (0 or more)$ on association linking $Company$ to Car		
20:36:17	Alex	Changed multiplicity of $Person$ to $Many (0 or more)$ on association linking $Person$ to Car		
20:36:21	Alex	Changed multiplicity of Car to $Many (0 or more)$ on association linking $Bank$ to Car		
20:36:23	Jim	Request	Opinion	Do you think we have enough?
20:36:25	Alex	Changed multiplicity of $Bank$ to $Many (0 or more)$ on association linking $Bank$ to Car		
20:36:31	Alex	Inform	Suggest	I think Is this correct?
20:36:43	Jim	Acknowledge	Accept	Yes , I think it is correct.
20:36:53	Sheldon	Inform	Suggest	I think car is linked to car loan through comapny/bank/person
20:37:05	Jim	Task	Coordinate Group Activity	OK. Let's move on
20:37:10	Alex	Discuss	Doubt	I'm not so sure
20:37:17	Sheldon	Discuss	Agree	Yes, I agree
~ Start of Knowledge Sharing Episode ~				
20:37:20	Jim	Inform	Suggest	I think we need a descriminator
20:37:25	Alex	Discuss	Agree	Yes, I agree
20:37:34	Alex	Request	Clarification	Can you explain why/how
20:37:38	Alex	Request	Illustration	Please show me
20:37:41	Sheldon	Inform	Suggest	I think it was car owner id
20:37:54	Sheldon	Inform	Suggest	I think it links to car
20:38:01	Jim	Acknowledge	Accept	Yes , i agree
20:38:47	Sheldon	Inform	Elaborate	To elaborate car owner id..is like that example we did b4
20:39:17	Sheldon	Inform	Suggest	I think person, bank, and company follows car owner ID
20:39:34	Sheldon	Maintenance	Apologize	Sorry I see it now
~ End of Knowledge Sharing Episode ~				
20:39:43	Jim	Task	Coordinate Group Activity	Are you ready to finish?
20:39:47	Sheldon	Changed current agenda item to > Verify diagram		
20:40:09	Alex	Acknowledge	Accept	OK
20:40:12	Sheldon	Task	Coordinate Group Activity	Are you ready
20:40:19	Jim	Acknowledge	Accept	Yes , I am ready

APPENDIX H

MATLAB code for ISODATA Algorithm

```
function [ K_c, class, n] = isodata_with_thresholding(S, K_c, K, thresh)
% S, K_c, K, sigma_split, d_l, n_min, L_max, I
% ISODATA : The iterative self organizing data analysis technique
%  Inputs:
% S: feature vectors
% K_c: initial number of clusters
% K: desired number of clusters
% sigma_split: max std dev threshold
% d_l: minimum distance threshold for lumping clusters
% n_min: minimum number of feature vectors that a cluster must have to not be
eliminated
% L_max: maximum number of cluster pairs that can be lumped in an iteration
% I:  maximum iterations

sigma_split = .5;
d_l = .5;
n_min = 2;
L_max = 2;
I = 100;

%STEP 1
% Initialize iteration number and parameters
i = 0;
Q = length(S);
cc = transpose([[1:Q];zeros(1,Q)]);  % cluster change assignment
class = transpose([[1:Q];zeros(1,Q)]);
n = zeros(2*K+1, 1); % number of vectors per cluster

% STEP 2
% Assign first K_c sample vectors as cluster centers
for k=1:K_c
    z(k,:) = S(k,:);  % Assign K_c centers
    class(k, 2) = k;
end

split = 1;
while i <= I
    while split == 1
            % STEP 3
            % Assign feature vectors to clusters via minimum distance, &
            % check for change
            change_cluster = 0;
            for q=1:Q
                  d_min = 999999.9;
                  for k=1:K_c
                      % Calculate the euclidean distance between vector and
                       % center
                      if euclid(S(q,:), z(k,:)) < d_min
                          d_min = euclid(S(q,:), z(k,:));
                          k_min = k;
                      end
                  end
                  % assign the qth vector to class k_min
                  if d_min < thresh
                      cc(q,2) = k_min;
                  end
                  % adjust the number of vectors per cluster
```

```
        % n(k_min) = length(find(class(:,2) == k_min));
        if cc(q,2) ~= class(q,2) % cluster center has changed
            change_cluster = 1;
        end
end
% adjust the number of vectors per cluster
for k=1:K_c
    n(k) = length(find(cc(:,2) == k));
end
% If no clusters change, exit program and return values, else
% continue
if change_cluster == 0
    return
    else class = cc;
    end

    % STEP 4
    % Eliminate clusters with less than n_min vectors, & update #
    % of clusters
    for k=1:K_c
        if n(k) < n_min % # of vectors in cluster is too small
            for r = k+1:K_c
                n(r-1)= n(r); % adjust indices (shift down one)
                z(r-1,:) = z(r,:);
            end
            for q=1:Q
                if class(q,2) > k
                    class(q,2) = class(q,2) - 1;
                end
            end
            K_c = K_c - 1; % Reduce current number of clusters
        end
    end

    % STEPS 5 & 6
    % Compute new centers for all clusters
    % Compute mean squared error (MSE) of each cluster
    for k=1:K_c
        % find vectors assigned to cluster k
        class_k = find(class(:,2) == k);
        z(k,:) = 1/n(k)*sum(S(class_k,:));  % compute new center
        % create a matrix with as many rows as length(class_k)
        sumB = 0;
        for h=1:length(class_k)
            sumB = sumB + euclid(S(class_k(h),:), z(k,:))^2;
        end
        D(k) = 1/n(k)*sumB; % MSE
    end

    % STEP 7
    % Compute total mean squared error and maximum component
    % variance for each cluster
    D_total = 1/Q*sum(n(1:K_c).*D');
    for k=1:K_c
        s_max = 0.0;
        sumA = 0.0;
        class_k = find(class(:,2) == k);    % find indices of
                                            % vectors in class k
        for h=1:length(class_k)
            sumA = (sum(sumA + (S(class_k(h),:) –
              z(k,:)).^2))/n(k);
        end
        if sumA >= s_max   % find maximum stdev from S_max
            s_max = sumA;
```

```
            end
            sigma_max(k) = sqrt(sumA);
            [y,i] = max(sum(S(class_k,:)));
            n_max(k) = i;  % index of feature with max value
      end

      % STEP 8
      % Check for performing splitting or lumping
      split = 0;
      if (K_c <= K/2 & I <=i)
            % STEP 9
            % Check conditions to split cluster into 2 clusters
            for k=1:K_c
                if sigma_max(k) > sigma_split
                    if ((n(k) > 2*n_min+1 & D(k) > D_total) |
                        (K_c < K/2))
                        split = 1;
                        K = K+1; % inc. total number of clusters
                        z(k,n_max(k)) = z(k,n_max(k)) –
                              0.5*sigma_max(k);
                        z(K,n_max(k)) = z(K,n_max(K)) –
                              0.5*sigma_max(K);
                    end
                end
            end
        end
    end
% WHILE LOOP (from top of step 3) ends here
% If split == 1, loop again from step 3
% else if split == 0, continue with rest of step 8
if (i > I | (rem(i,2) == 0 & K_c >= 2*K))
  if (i > I)
      d_l = 0;
  end
end


% STEP 10
% Compute distances between all pairs of cluster centers, lump if too
% close.
% Compare each distance with d_l, & set lump flag to TRUE if less
% than d_l
for k=1:K_c - 1
  for r=k+1:K_c
      D(r,k) = euclid(z(r,:),z(k,:));
      if D(r,k) < d_l
          L(r,k) = 1;
      else L(r,k) = 0;
      end
  end
end


% STEP 11
% Find smallest one of distances D(r,k) less than d_l and lump the 2
% clusters.
% Repeat until either L_max lumpings are done or no more distances
% are < d_l
count = 0;
while(1)
  d_min = 999999.9;
  empty = 1;
  for k=1:K_c-1
      for r=k+1:K_c
```

```
                if L(r,k)== 1 & (D(r,k) < d_min)
                    d_min = D(r,k);
                    empty = 0;
                    r_ = r;
                    k_ = k;
                end
            end
        end
        if empty == 0
            % Lump cluster pair C_r*, C_k* together, & update clusters
            k_ = min(r_,k_);
            r_ = max(r_,k_);
            n(k_) = n(k_) + n(r_);
            % avg. center of new cluster
            z(k_,:) = 1/n(k_)*(n(r_)*z(r_,:) + n(k_)*z(k_,:));
            % update assignment index
            class_r = find(class(:,2) == r_);
            class(class_r,2) = k_;
            % update indices
            n(r_) = 0;
            z(r_,:) = 0;
            for k=r_:K_c
                n(k) = n(k+1);
                z(k,:) = z(k+1,:);
                class_k = find(class(:,2) == k);
                class(class_k,2) = k-1;
            end
            K_c = K_c -1;
        end
        count = count + 1;
        if (empty == 1 | count >= L_max)
            break;
        end
    end

      % STEP 12
      % Do another iteration (starting at step 3) or else stop
      i = i + 1;
end


function e = euclid(v1,v2)
% Euclidean distance between two vectors
e = sqrt(sum((v1 - v2).^2));
```

BIBLIOGRAPHY

BIBLIOGRAPHY

Baker, M. (1994). A model for negotiation in teaching-learning dialogues. *Journal of Artificial Intelligence in Education*, 5(2), 199-254.

Baker, M., Hansen, T., Joiner, R., & Traum, D. (1999). The role of grounding in collaborative learning tasks. In P. Dillenbourg (Ed.), *Collaborative Learning: Cognitive and Computational Approaches* (pp. 31-63). Amsterdam: Elsevier Science.

Baker, M., & Lund, K. (1997). Promoting reflective interactions in a computer-supported collaborative learning environment. *Journal of Computer Assisted Learning*, 13, 175-193.

Barnes, D., & Todd, F. (1977). *Communication and learning in small groups*. London: Routledge & Kegan Paul Ltd.

Barros, B., & Verdejo, M.F. (1999). An approach to analyse collaboration when shared structured workspaces are used for carrying out group learning processes. *Proceedings of the Ninth International Conference on Artificial Intelligence in Education*, 449-456.

Blackboard, Inc. (1999). Blackboard CourseInfo ™ 4.0 Product Overview. [On-line], Available: http://company.blackboard.com/CourseInfo/index.html.

Blumenfeld, P., Marx, R., Soloway, E., & Krajcik, J. (1996). Learning with peers: From small group cooperation to collaborative communities. *Educational Researcher*, 25(8), 37-40.

Bos, M. (1937). Experimental study of productive collaboration. *Acta Psychologica*, 3, 315-426.

Brown, A., & Palincsar, A. (1989). Guided, cooperative learning and individual knowledge acquisition. In L. Resnick (Ed.), *Knowing, learning, and Instruction: Essays in honor of Robert Glaser*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Bruckman, A., & Bonte, A. (1997). MOOSE goes to school: A comparison of three classrooms using a CSCL environment. *Proceedings of Computer-Support for Collaborative Learning (CSCL '97*), Toronto, Canada, 20-26.

Cahn, J. & Brennan, S (1999). A psychological model of grounding and repair in dialog. *Proceedings of the AAAI Fall Symposium: Psychological Models of Communication in Collaborative Systems*, Cape Cod, MA, 25-33.

Carletta, J., Isard, A., Isard, S., Kowtko, J., Doherty-Sneddon, G., & Anderson, A. (1997). The reliability of a dialogue structure coding scheme. *Computational Linguistics,* 23, 13-32.

Charniak, E. (1993). *Statistical Language Learning*. Cambridge, MA: MIT Press.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reinmann, P., & Glaser, R. (1989). Self-Explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13, 145-182.

Cohen, E. (1994). Restructuring the classroom: Conditions for productive small groups. *Review of Educational Research*, 64(1), 1-35.

Constantino-Gonzalez, M.A., & Suthers, D. (2001). Coaching collaboration by comparing solutions and tracking participation. *Proceedings of Euro-CSCL 2001*, Maastricht, The Netherlands.

Constantino-Gonzalez, M.A., & Suthers, D. (in press). Coaching web-based collaborative learning based on problem solution differences and participation. *International Journal of Artificial Intelligence in Education*, 13.

Damon, W. (1984). Peer education: The untapped potential. *Journal of Applied Developmental Psychology*, 5, 331-343.

Dillenbourg, P. (1999). What do you mean by "Collaborative Learning". In P. Dillenbourg (Ed.) *Collaborative Learning: Cognitive and Computational Approaches* (pp.1-19). Amsterdam: Elsevier Science.

Dillenbourg, P., & Traum, D. (1999). Does a shared screen make a shared solution? *Proceedings of the Computer Support for Collaborative Learning 1999 Conference (CSCL '99),* Palo Alto, California, 127-135.

Doise, W., Mugny, G., & Perret-Clermont A. (1975). Social interaction and the development of cognitive operations. *European Journal of Social Psychology*, 5(3), 367-383.

Dweck (1989). Motivation. In A. Lesgold & R. Glaser (Eds.) *Foundations for a Psychology of Education* (pp 87-136). Hillsdale, NJ: Lawrence Erlbaum Associates.

Ellis, S., Klahr, D., & Siegler, R. (1994, April). *The birth, life, and sometimes death of good ideas in collaborative problem solving.* Paper presented at the annual meeting of the American Educational Research Association, New Orleans.

Fisher, L., & van Belle, G. (1993) *Biostatistics: A methodology for the health sciences*. New York: John Wiley & Sons, Inc.

Flores, F., Graves, M., Hartfield, B., & Winograd, T. (1988). Computer systems and the design of organizational interaction. *ACM Transactions on Office Information Systems,* 6(2), 153-172.

Freitag, D., & McCallum, A. (2000). Information extraction with HMM structures learned by stochastic optimization. *Proceedings of AAAI 2000*.

Goodman, B., Hitzeman, J., Linton, F., & Ross, H. (2002). Intelligent agents for collaborative learning: Predicting the roles of dialog participants. Manuscripts submitted for publication.

Gott, S., & Lesgold, A. (2000). Competence in the workplace: How cognitive performance models and situated instruction can accelerate skill acquisition. In R. Glaser (Ed.) *Advances in Instructional Psychology: Vol. 5. Educational Design and Cognitive Science* (pp. 239-327). Mahwah, NJ: Lawrence Erlbaum Associates.

Groen, G. (1978). The theoretical ideas of Piaget and educational practice. In P. Suppes (Ed.), *Impact of Research on Education: Some Case Studies* (pp. 267-317). Washington, D.C.: National Academy of Education

Guzdial, M., Hmelo, C., Hubscher, R., Newstetter, W., Puntambekar, S., Shabo, A., Turns, J., & Kolodner, J. (1997). Integrating and guiding collaboration: Lessons learned in computer-supported collaboration learning research at Georgia Tech. *Proceedings of Computer-Support for Collaborative Learning (CSCL '97)*, Toronto, Ontario, 91-100.

Hatano, G., & Inagaki, K. (1991). Sharing cognition through collective comprehension activity. In L. Resnick, J. Levine, & S. Teasley (Eds.), *Perspectives on Socially Shared Cognition* (pp 331-348). American Psychological Society, Washington DC.

Herring, S.C. (1999). Interactional Coherence in CMC. *Hawaii International Conference on the System Sciences*. (CD-ROM). Maui, Hawai'i: Institute of Electrical and Electronics Engineers, Inc. (IEEE).

Inaba, A., & Okamoto, T. (1997). Negotiation Process Model for Intelligent Discussion Coordinating System on CSCL Environment. *Proceedings of the 8th World Conference on Artificial Intelligence in Education*, Kobe, Japan, 175-182.

Jarboe, S. (1996). Procedures for enhancing group decision making. In B. Hirokawa & M. Poole (Eds.), *Communication and Group Decision Making* (pp. 345-383). Thousand Oaks, CA: Sage Publications.

Jeong, H. (1998). *Knowledge co-construction during collaborative learning*. Unpublished Doctoral Dissertation. University of Pittsburgh, PA.

Jermann, P., & Dillenbourg, P. (1999). An analysis of learner arguments in a collective learning environment. *Proceedings of Computer-Support for Collaborative Learning (CSCL'99),* Stanford, California, 265-273.

Jermann, P., & Schneider, D. (1997). Semi-structured interface in collaborative problem solving. *Proceedings of the First Swiss Workshop on Distributed and Parallel Systems*, Lausanne, Switzerland.

Jermann, P., Soller, A., & Muehlenbrock, M. (2001). From mirroring to guiding: A review of state of the art technology for supporting collaborative learning. *Proceedings of the First European Conference on Computer-Supported Collaborative Learning*, Maastricht, The Netherlands, 324-331.

Johnson, R. & Johnson, D. (1991). *Learning Together and Alone: Cooperative, Competitive, and Individualistic Learning (3rd ed.).* Boston, MA: Allyn & Bacon.

Johnson, R., Johnson, D., & Stanne, M. (1985). Effects of cooperative, competitive, and individualistic goal structures on computer-assisted instruction. *Journal of Educational Psychology*, 77(6), 668-677.

Joiner, R. (1995). The Negotiation of dialogue focus: An investigation of dialogue processes in joint planning in a computer based task. In C. O'Malley (Ed.) *Computer-Supported Collaborative Learning*. Berlin: Springer-Verlag.

Juang, B., & Rabiner, L. (1989). A probabilistic distance measure for Hidden Markov models. *AT&T Technical Journal*, 64(2).

Kanungo, T. (1998). Hidden Markov Model (HMM) Software (version 1.02) [Computer software]. Available: http://www.cfar.umd.edu/~kanungo/software/software.html.

Katz, S., Aronis, J., & Creitz, C. (1999). Modelling pedagogical interactions with machine learning. *Proceedings of the Ninth International Conference on Artificial Intelligence in Education*, LeMans, France, 543-550.

Katz, S., Lesgold, A., Hughes, E., Peters, D., Eggan, G., Gordin, M., Greenberg., L. (1998). Sherlock II: An intelligent tutoring system built upon the LRDC Tutor Framework. In C.P. Bloom & R.B. Loftin (Eds.), *Facilitating the Development and Use of Interactive Learning Environments* (pp. 227-258). New Jersey: Lawrence Erlbaum Associates.

King, A., Staffieri, A., & Adelgais, A. (1998). Mutual peer tutoring: Effects of structuring tutorial interaction to scaffold peer learning. *Journal of Educational Psychology*, 90(1), 134-152.

Koschmann, T. (2002). Dewey's contribution to the foundations of CSCL research. Keynote discussion presented at *Computer Supported Collaborative Learning 2002*, Boulder, Colorado, 17-22.

Kruskal, J. & Wish, M. (1978). *Multidimensional Scaling*. Newbury Park, California: Sage Publications.

Lavery, T., Franz, T., Winquist, J., & Larson, J. (1999). The role of information exchange in predicting group accuracy on a multiple judgment task. *Basic and Applied Social Psychology*, 2(4), 281-289.

Lesgold, A., Katz, S., Greenberg, L., Hughes, E., & Eggan, G. (1992). Extensions of intelligent tutoring paradigms to support collaborative learning. In S. Dijkstra, H. Krammer, & J. van Merrienboer (Eds.), *Instructional Models in Computer-Based Learning Environments*. Berlin: Springer-Verlag, 291-311.

Levine, J. M. & Moreland, R. L. (1998). Small groups. In D. Gilbert, S. Fiske, & G. Lindzey (Eds.), *The handbook of social psychology* (4th ed.) Boston, MA: McGraw-Hill, 415-469.

Looney, C. (1997). *Pattern recognition using neural networks*. New York: Oxford University Press.

McManus, M., & Aiken, R. (1995). Monitoring computer-based problem solving. *Journal of Artificial Intelligence in Education*, 6(4), 307-336.

Mennecke, B. (1997). Using group support systems to discover hidden profiles: An examination of the influence of group size and meeting structures on information sharing and decision quality. *International Journal of Human-Computer Studies*, 47, 387-405.

Muehlenbrock, M. (2001). *Action-based collaboration analysis for group learning*. Doctoral Dissertation. University of Duisburg, Germany.

Muehlebrock, M., & Hoppe, H. U. (2001). A collaboration monitor for shared workspaces. *Proceedings of the International Conference on Artificial Intelligence in Education (AIED-2001),* San Antonio, TX, 154-165.

Murphy, K. (1998). MATLAB HMM Toolbox. [Computer Software]. Available: http://www.cs.berkeley.edu/~murphyk/Bayes/hmm.html.

O'Day, V., Bobrow, D., Bobrow, K., Shirley, M., Hughes, B., Walters, J. (1998). Moving practice: From classrooms to MOO rooms. *Computer Supported Cooperative Work (CSCW): The Journal of Collaborative Computing*, 7(1/2), 9-45.

Okada, T., & Simon, H. (1997). Collaborative discovery in a scientific domain. Cognitive Science, 21(2), 109-146.

Perret-Clermont, A., Perret, J., & Bell, N. (1991). The social construction of meaning and cognitive activity in elementary school children. In L. Resnick, J. Levine, & S. Teasley (Eds.) *Perspectives on Socially Shared Cognition* (pp. 41-62). American Psychological Society, Washington DC.

Rabiner, L. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257-286.

Resnick, L., Levine, J., & Teasley, S. (eds.) (1991). *Perspectives on Socially Shared Cognition*. American Psychological Society, Washington DC.

Rogoff (1991) Social interaction as apprenticeship in thinking. In L. Resnick, J. Levine, & S. Teasley (Eds.) *Perspectives on Socially Shared Cognition*. American Psychological Society, (Chapter 16). Washington DC.

Roth, D. (1999). Learning in natural language. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '99)*, Stockholm, Sweden, 898-904.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorensen, W. (1991). *Object-Oriented modeling and design*. Englewood Cliffs, NJ: Prentice Hall.

Russell, S., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall.

Salomon, G., & Globerson, T. (1989). When teams do not function the way they ought to. *International Journal of Educational Research*, 13, 89-98.

Scardamalia, M., & Bereiter, C. (1994). Computer support for knowledge-building communities. *Journal of the Learning Sciences*, 3(3), 265-283.

Schwartz, D. (1999). The productive agency that drives collaborative learning. In P. Dillenbourg (Ed.), *Collaborative Learning: Cognitive and Computational Approaches* (pp. 197-218). Elsevier Science.

Schrodt, P. (1999). Pattern recognition of international crises using Hidden Markov Models. In D. Richards (Ed.) *Political Complexity: Nonlinear Models of Politics*. Ann Arbor: University of Michigan Press.

Shepard, R. (1980). Multidimensional Scaling, Tree-fitting, and clustering. *Science*, 210, 90-398.

Shepard, R., & Arabie, P. (1979). Additive Clustering: Representation of similarities as combinations of discrete overlapping properties. *Psychological Review*, 86(2), 87-123.

Singley, M., Fairweather, P., & Swerling, S. (1999). Team tutoring systems: Reifying roles in problem solving. *Proceedings of Computer-Support for Collaborative Learning (CSCL '99)*, Stanford, California, 538-548.

Slavin, R. (1980). Cooperative Learning. *Review of Educational Research*, 50(2), 315-342.

Smyth, P. (1997). Clustering sequences with Hidden Markov Models. In M. Mozer, M. Jordan, & T. Petsche (Eds.) *Proceedings of the 1996 Conference on Advances in Neural Information Processing Systems 9* (pp. 648-654). MIT Press.

Soller, A. (2001) Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education*, 12(1), 40-62.

Soller, A., & Lesgold, A. (in press). Modeling the process of knowledge sharing. In U. Hoppe & M. Ikeda (Eds.) *New Technologies for Collaborative Learning*. Kluwer Academic Publishers.

Soller, A., Wiebe, J., & Lesgold, A. (2002). A machine learning approach to assessing knowledge sharing during collaborative learning activities. *Proceedings of Computer-Support for Collaborative Learning 2002*, Boulder, CO, 128-137.

Stasser, G. (1999). The uncertain role of unshared information in collective choice. In L. Thompson, J. Levine, & D. Messick (Eds.) *Shared Knowledge in Organizations* (pp 49-69). Hillsdale, NJ: Erlbaum.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van Ess-Dykema, C., & Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3).

Suthers, D., Connelly, J., Lesgold, A., Paolucci, M., Toth, E., Toth, J., & Weiner, A. (2001). Representational and advisory guidance for students learning scientific inquiry. In K. D. Forbus & P. J. Feltovich (Eds.), *Smart Machines in Education: The Coming Revolution in Educational Technology* (pp. 7-35). Menlo Park: AAAI Press.

Teasley, S. & Roschelle, J. (1993). Constructing a joint problem space: The computer as a tool for sharing knowledge. In S. Lajoie & S. Derry (Eds.), *Computers as cognitive tools* (pp. 229-257). Hillsdale, NJ: Lawrence Erlbaum.

Tedesco, P., & Self, J. (2000). Using meta-cognitive conflicts in a collaborative problem solving environment. *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, Montreal, Canada, 232-241.

Thagard, P. (1997) Collaborative knowledge. *Nous*, 31, 242-261.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13, 260-269.

Walker, M., Litman, D., Kamm, C., & Abella, A. (1997). PARADISE: A framework for evaluating spoken dialogue agents. *Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics (ACL '97)*, Madrid, Spain, 271-280.

Webb, N. (1992). Testing a theoretical model of student interaction and learning in small groups. In R. Hertz-Lazarowitz and N. Miller (Eds.), *Interaction in Cooperative Groups: The Theoretical Anatomy of Group Learning* (pp. 102-119). New York: Cambridge University Press.

Webb, N., & Farivar, S. (1994). Promoting helping behavior in cooperative small groups in middle school mathematics. *American Educational Research Journal*, 31 (2), 369-395.

Webb, N., & Palincsar, A. (1996). Group processes in the classroom. In D. Berlmer & R. Calfee (Eds.), *Handbook of Educational Psychology* (pp. 841-873). New York: Simon & Schuster Macmillan.

Webb, N., Troper, J., & Fall, R. (1995). Constructive activity and learning in collaborative small groups. *Journal of Educational Psychology*, 87(3), 406-423.

Wegner, D. (1987). Transactive memory: A contemporary analysis of the group mind. In B. Mullen & G. Goethals (Eds.), *Theories of Group Behavior* (pp. 185-208). New York: Springer-Verlag.

Winquist, J. R. & Larson, J. R. (1998). Information pooling: When it impacts group decision making. *Journal of Personality and Social Psychology, 74,* 371-377.

Vygotsky, L. (1978). *Mind in Society: The development of higher psychological processes*. Harvard University Press. Cambridge, MA. Collection of work edited by Michael Cole, Vera John-Steiner, Sylvia Scribner, & Ellen Souberman.

Yang, J., Xu, Y., & Chen, C. (1997). Human action learning via Hidden Markov Model. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 27(1), 34-44.