Collaborative Software and Community Building

by

Rev. Jeffrey B. Williams

B.S., Elmhurst College, 1988
M.Div., Concordia Theological Seminary, 1992

AN ABSTRACT OF A DISSERTATION

Submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Secondary Education
College of Education

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2005

ABSTRACT

How does collaborative software help in the formation of a learning community? This study looks at the experiences of students in a first level Computer Science class as they use Manhattan Virtual Classroom (MVC). Although this case study began with the assumption that a learning community would form, it quickly became obvious that student participation in the MVC was a larger issue.

The course chosen for this study was CSC-150 - Foundations of Computer Science, as taught in the Spring 2004 semester at a Midwestern university. Two traditional (face-to-face) course sections were given access to Manhattan Virtual Classroom for the purpose of discussions, comments, questions, and virtual office hours.

Many students did not take advantage of this collaborative tool. Several reasons are considered, the reluctance of freshmen to participate (Goldberg, 1997; Carlson et al., 1996), professor teaching style, and student perceptions of their own contributions to the class.

Several conclusions are drawn from this study how to increase student participation. These include better training in the use of the software, use of smaller groups within the Manhattan Virtual Classroom environment, clearly stated professor expectations, and a general adoption of this technology for other classes.

Collaborative Software and Community Building

by

Rev. Jeffrey B. Williams

B.S., Elmhurst College, 1988
M.Div., Concordia Theological Seminary, 1992

A DISSERTATION

Submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Secondary Education
College of Education

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2005

Approved by:

Major Professor:
Diane McGrath

ABSTRACT

How does collaborative software help in the formation of a learning community? This study looks at the experiences of students in a first level Computer Science class as they use Manhattan Virtual Classroom (MVC). Although this case study began with the assumption that a learning community would form, it quickly became obvious that student participation in the MVC was a larger issue.

The course chosen for this study was CSC-150 - Foundations of Computer Science, as taught in the Spring 2004 semester at a Midwestern university. Two traditional (face-to-face) course sections were given access to Manhattan Virtual Classroom for the purpose of discussions, comments, questions, and virtual office hours.

Many students did not take advantage of this collaborative tool. Several reasons are considered, the reluctance of freshmen to participate (Goldberg, 1997; Carlson et al., 1996), professor teaching style, and student perceptions of their own contributions to the class.

Several conclusions are drawn from this study how to increase student participation. These include better training in the use of the software, use of smaller groups within the Manhattan Virtual Classroom environment, clearly stated professor expectations, and a general adoption of this technology for other classes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

"O give thanks unto the LORD; for He is good:
for His mercy endureth for ever."
Psalm 136:1 KJV

Many people have helped in the preparation of this dissertation and in the process of working towards this degree. To begin the list, Dr. Diane McGrath, as the major professor and advisor, has provided constant insight, support, and guidance. The other members of my committee, Dr. Chandima Cumaranatunge, Dr. Judith Hughey, Dr. Tweed Ross, and the outside chair, Dr. Briana Goff, have also provided encouragement and assistance.

At Concordia University Wisconsin, Dr. Gary Locklair, the professor who taught the class that I studied, the department chair for Computer Science, and my mentor at CUW, provided great assistance. Dr. Angus Menuge, also in the Computer Science department, Dr. Donna Reimer-Becker, Dr. Paul Zietlow, and Rev. Dr. Roland "Cap" Ehlke also provided encouragement and suggestions in the production of this dissertation. Many students, both in the sections of CSC-150 and in the Computer Science department, provided technical support, encouragement, and a friendly ear. Specifically, however, David Locklair, Pat Delancy, Jon Auer, Chris Raasch, Mike Standfest, and Jesse Burant were particularly helpful.

Mentioned last, but certainly not least in help and sacrifice, are my daughters Dianna and Sarah, and my long-suffering spouse unit, Sally. Indeed, this research could not have been completed without Sally's excellent work as a transcriptionist, proof reader, and sounding board. As the old saying goes, "Behind every successful man is a good woman," is nowhere better proven than here. Proverbs 31 was written with such as Sally in mind.

To all these marvelous people, and to those I have forgotten to mention by name, a hearty thank you. Any sins of omission or commission during this process are my fault, and not the sins of another.

Written on this Feast of the Epiphany, 2005.

DEDICATION

To my spouse-unit of more decades than we care to count, it sure has been interesting.

Chapter 1: Setting the Stage

> It's at first difficult to see what technological paradigm follows from the community rather than delivery view of education. As each community has its own specific interests, its own way of knowing, its own central endeavors, generalizing seems out of place. But communities are made up of people, and at the heart of all social relations and practice lies human communication of one form or another. On the basis of this assumption, we suggest that learning technology should be built around a conversational paradigm. (Brown & Duguid, 1995, p. 328)

How does a learning community form? How does a random group of students, placed together by chance and circumstances, coalesce into a mutually supporting group that has the goal to learn about a given topic? What are the stages of growth in a community? How might the addition of a collaborative software package with discussion groups to a face-to-face or traditional college class setting assist in forming such a community by providing additional communication tools?

Few people would disagree with the proposition that the heart of learning is communication. As educational psychologists Bigge and Shermis (1992, p. 4) wrote: "Teaching was done by 'teachers' telling and showing students how, complimenting the learners when they did well and scolding or punishing them when they did poorly." Bigge and Shermis also wrote (1992):

> Human beings have some distinguishing characteristics that give a unique quality to a study of them. First, they talk and are *time-binding* individuals; *time-binding* means that both a past and a future enter into their present perception of things. (p. 1)

These talking, interacting teachers and students worked together so the teachers could teach and the students could learn. This interacting is communication, whether verbal or not.

Terry Anderson and Randy Garrison, both in the administration of the University of Alberta, wrote (1998):

> Educational communication in its best sense should be reciprocal (i.e., two-way), consensual (i.e., voluntary), and collaborative (i.e., shared control). ... Educational communication must facilitate the construction and negotiation of meaning, which is dependent upon critical discourse and knowledge confirmation. Educational communication must be explanatory and not just confirmatory. That is, it explains why a conception makes sense or not, as opposed to simply stating that it is right or wrong. (p. 98)

Community

What is a community? Although the term was learned in third grade social studies, where we diligently studied about people who lived in the Northwest and New Orleans and the Great Plains, the term was not really clear. Amy Jo Kim, in her book on community building on the World Wide Web (2000) wrote:

> When it was my turn to speak, I began with my definition of community: A community is a group of people with a shared interest, purpose, or goal, who get to know each other better over time. By this definition, I explained, the scattered collection of people who watch Star Trek reruns each week aren't really a community, because they have no way to communicate with each other. The Trekkies who meet up at conventions, fan sites, and mailing lists, however, could be a community, because they can get to know each other better over time.
>
> Afterwards, I was surrounded by Web-savvy conference attendees who thanked me for pointing out that building a community involves more than delivering content to a particular group. Over lunch, we debated the finer points of what makes a successful community—but we all agreed that, because members have to have a way to get to know each other, a community can't really exist without gathering places. (p. 28)

A learning community, then, can be a random group of people who agree to come together or are placed together for a length of time for the purpose of discovering new facts and gaining a new understanding of a subject. As the learning period continues, students have an opportunity to learn more about each other as they discuss the class and integrate that which is being learned into their own world view.

What started as a random group of people coalesces into a community with shared interests and goals. Often the lessons learned in a class are recalled, not from lectures or readings, but from conversations and shared projects. Meaning is constructed by each learner as a result of both planned and ad-hoc cooperation and conversation, as well as individual efforts.

Communication, Processing, and Learning

If learning is dependent on communication, then anything that overcomes barriers to communication could not but help to improve education. We are reminded by the noted educational researcher, Seymour Papert, (1993) that:

> back to Piaget's doctrine that knowledge simply cannot be "transmitted" or "conveyed ready made" to another person. Even when you seem to be successfully transmitting information by telling it, if you could see the brain processes at work you would observe that your interlocutor is "reconstructing" a personal version of the information you think you are "conveying." (p. 142)

The transmission of information from one person to another is only part of the communication process. Claude Shannon (1948, p. 379) wrote, "The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point." Although Shannon was discussing the problem of determining communication channel bandwidth, his definition has been applied to many other aspects of communication.

Shannon (1948, p. 379) continued, "Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities." In learning the message does have meaning because it relates to a conceptual entity.

Part of the learning process includes the processing of information which has been received. Processing includes the storage of information for later retrieval. Active learning, when the learner is involved in an activity, is much more effective than passive listening because active learning creates multiple pathways to the information which has been received, making retrieval easier.

The MIT professor, Marvin Minsky, (1985) wrote:

> How do we ever understand anything? Almost always, I think, by using one or another kind of analogy — that is, by representing each new thing as though it resembles something we already know. Whenever a new thing's internal workings are too strange or complicated to deal with directly, we represent whatever parts of it we can in terms of more familiar signs. This way, we make novelty seem similar to some more ordinary thing. It really is a great discovery, the use of signals, symbols, words, and names. They let our minds transform the strange into the commonplace. (p. 57)

A traditional view of education also demonstrates the importance of activity in the learning process. Donald Pratt, in an instructional techniques class taught at AT&T Bell Laboratories in 1985, told students that the instructor's job was to "explain, exercise, and evaluate." An instructor must first explain or present new material. The students then must use the material, thus internalizing the explanation. Finally both students and instructor must evaluate the learning which has taken place so to understand the effectiveness of the instruction.

In this traditional view of education, the term "exercise" means more than paper and pencil problems. The term "exercise," chosen in part for the alliteration, means any activity which causes the student to use that newly acquired information. Thus an exercise can be a role playing game, building a model, repairing a circuit, writing a program, working with other students, practicing an instrument, or any other activity.

Brown and Duguid (2002, p. 120), formerly researchers at the Xerox Palo Alto Research Center, wrote, “Knowledge is something we digest rather than merely hold. It entails the knower’s understanding and some degree of commitment.” This statement agrees with the idea that the learner must assimilate that which is taught so to be able to use that knowledge.

Peer Communication

In 1971, my first flight instructor, Al Flemming, told me the best way to truly learn about flying was to become an instructor. Only as you teach, he claimed, do you really begin to understand the subject. Other instructors, such as Dan Koen, the instructor who led me through the process of becoming a flight instructor, and who later taught me the art of flying multi-engine airplanes, added that the instructor must be able to explain a concept simply. He repeated time and again, “Unless you can explain it simply, you don’t know it yourself.”

If these and many other flight instructors are right, could not a student reinforce learning by explaining newly formed or acquired knowledge to another student? This certainly is part of the “exercise” portion of the traditional view of teaching expressed by Pratt. Such explanation reinforces knowledge, forces the student to express the relationship between various elements of information, and helps the student to construct new knowledge.

Knowledge creation includes the administration and organization of knowledge, as was mentioned by Minsky (1985).

> This explanation of the difference between older and younger children was first proposed by Seymour Papert in the 1960s, when we first started to explore society of mind ideas. Most previous theories had tried to explain Piaget's experiments by suggesting that children develop different kinds of reasoning as time goes by. That certainly is true, but the importance of Papert's conception is in emphasizing not merely the ingredients of reasoning, but how they're organized: a mind cannot really grow very much merely by accumulating knowledge. It must also develop better ways to use what it already knows. That principle deserves a name.
>
> *Papert's Principle:* Some of the most crucial steps in mental growth are based not simply on acquiring new skills, but on acquiring new administrative ways to use what one already knows. (p. 102)

Peer communication gives opportunity for such growth to happen. Students helping students reinforces and exercises newly gained knowledge, thus helping each student to solidify concepts and to organize them for further use.

A good teacher, then, would desire to use this strategy in helping students to learn. There are several reasons that peer communication, student helping student, may be done. It is a valid way of reinforcing and integrating information.

McKeachie, in his book on teaching at the university level (1978) proposes such teaching to improve student learning.

> Moreover, if students are to achieve application, critical thinking, or some higher cognitive outcomes, it seems reasonable to assume that they should have an opportunity to practice application and critical thinking and to receive feedback on the results. Group discussion provides an opportunity to do this. (p.50)

The idea of students teaching students is frequently used in the classroom. Many are the small group assignments and special reports in which students collaborate in learning. By involving the students in teaching, students are reinforcing the knowledge they have already gained, constructing new knowledge from their experience, and exercising their abilities to use that knowledge.

Collaborative Software

Collaboration is nothing new. People have always worked together to solve problems, to complete a task which cannot be finished by one person.

In 1959, according to Stewart Denenberg (1978), a research consultant, the state of Illinois in cooperation with the Control Data Corporation developed the PLATO System for Computer Aided Instruction (CAI). Denenberg noted:

> For some elusive reason, communications files are hardly ever shown, much less featured, when a PLATO system is demonstrated. This is a pity because these files, in my opinion, contribute very heavily to the unique character and power of the PLATO system. The CERL system (what PLATO users call the University of Illinois system) has over 500 and the CDC (Control Data Corporation) system over 300 of these communications files — or as they are called by the user population, "notesfiles." A notesfile can be thought of as an electronic bulletin board whereby any user can ready any other user's notes and respond to them, thus enlarging the volume of notes in the file. (p. 4)

Paul Ceruzzi (1998, p. 173) commented on PLATO and how it pointed to the future of educational computing. "In 1994 most of the predictions for PLATO came true, via the Internet and using a system called the World Wide Web."

Since the early 1970s people have developed a number of software tools for collaboration. Probably the simplest tool is the electronic mail list, which later evolved into USENET, essentially a public computer bulletin board. While mail lists may be adequate for certain tasks, other tools may be more appropriate for student collaboration.

Electronic bulletin boards (BBS) became popular as personal computers became more available. Where the first BBS systems were owned and sponsored by individuals, the concept of message groups and discussion areas quickly grew into CompuServe, GEnie, and Prodigy. Since the invention of the World Wide Web in 1992, such services have grown and consolidated (Christensen & Suess, 1989; Quarterman & Hoskins, 1986; Berners-Lee, 2000).

The World Wide Web was first designed as a collaboration tool. Tim Berners-Lee (2000), the creator of the World Wide Web, wrote:

> It was clear to me that there was a need for something like Enquire [an early database of concepts written by Berners-Lee] at CERN. In addition to keeping track of relationships between all the people, experiments, and machines, I wanted to access different kinds of information, such as a researcher's technical papers, the manuals for different software modules, minutes of meetings, hastily scribbled notes, and so on. Furthermore, I found myself answering the same questions asked frequently of me by different people. It would be so much easier if everyone could just read my database. (p. 15)

By combining the concepts of the BBS and access through the Internet, various programmers and companies developed course management systems and educational collaborative software. Systems such as WebCT and Blackboard provide means for students to interact with each other, professors to hold virtual office hours, and people widely separated by time and distance to hold discussions.

Collaboration in the Computer Science Curricula

In December, 2001, the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE) released the joint report, *Computing Curricula 2001.* This report, an update of *Computing Curricula 1991*, is designed to guide colleges and universities in developing computer science courses of study.

Computing Curricula 2001(2001) suggested:

> Few computer professionals can expect to work in isolation for very much of the time. Software projects are usually implemented by

> groups of people working together as a team. Computer science students therefore need to learn about the mechanics and dynamics of effective team participation as part of their undergraduate education. Moreover, because the value of working in teams (as well as the difficulties that arise) does not become evident in small-scale projects, students need to engage in team-oriented projects that extend over a reasonably long period of time, possibly a full semester or a significant fraction thereof.
>
> To ensure that students have the opportunity to acquire these skills as undergraduates, the CC2001 Task Force recommends that all computer science programs include the following:
>
> - Opportunities to work in teams beginning relatively early in the curriculum.
> - A significant project that involves a complex implementation task in which both the design and implementation are undertaken by a small student team. ... (p. 42)

In keeping with this philosophy, professors encourage students in the various computer science classes at Concordia University Wisconsin (the site of this research project) to work together in solving problems.

Skills in team work can be developed early in the computer science curriculum, as early as the first class in computer science.

Problem Statement

Brown and Duguid (2002, p. 126) related several instances of the need for community in learning. In one vignette, they discussed the informal interaction of computer technicians which, in fact, created a situation where technicians learned from each other. They wrote:

> As part of this common work-and-talk, creating, learning, sharing, and using knowledge appear almost indivisible. Conversely, talk without the work, communication without practice is if not unintelligible, at least unusable. Become a member of a community, engage in its practices, and you can acquire and make use of its knowledge and information. Remain an outsider, and these will remain indigestible.

Albert Bandura (1986, p. 142), the developer of Social Learning Theory, noted, "Understanding how new ideas and social practices spread within a society, or from one society to another, has important bearing on personal and social change." Knowledge spreads throughout a society by means of personal interaction.

Many new tools are available for communication, tools which had not seen widespread use before the last decade of the Twentieth Century. If we consider the explosion of computer aided communication as enabled through the invention of the Internet and the World Wide Web, we can see that technology is becoming more available. Paths of communication which did not exist before 1970 (Internet) or 1992 (World Wide Web) now are in common use. The development of fast modems, broadband communication media such as digital cable and Digital Subscriber Loop (DSL) phone service, and high speed wireless transmission have enabled high speed data communication from residences as well as from schools and businesses (Quarterman & Hoskins, 1986; Ceruzzi, 1998).

Purpose of the Study

The overall question of this study is, "How does collaborative software help in the formation of a learning community?" Slightly expanded, this question leads to several avenues of inquiry. Does collaborative software help in the formation of learning communities in which students can interact, thus helping to assimilate that information which has been imparted to them? How does a random group of students in a class become a true community?

In this study I have to studied the formation of a learning community in a first level computer science course by looking at the use of a collaborative communication tool (Manhattan Virtual Classroom) by students.

In this case study of collaborative software use I looked at the level of collaboration and types of communication used by students to interact with both the professor and other students. This was accomplished by interviewing students in CSC-150, Foundations of Computer Science, as given at Concordia University Wisconsin during the Spring semester, 2004. In addition, the students were asked about their experiences with the tool. Finally, I analyzed the conversations within the various message boards of the tool to see what types of discussions were conducted by the students in an effort to determine the growth of an online community.

As will be shown in the next chapter, very little research has been done concerning the use of collaborative software in education as it pertains to student participation. This study will help determine if collaborative software can contribute to the creation of a learning community, giving increased opportunities to students for exercising newly formed knowledge.

Certainly the results of this study will be important in designing future computer science classes which are taught in the traditional classroom. However, collaborative software may also help distance education students overcome the problems of temporal and spatial separation from instructors and peers, helping to form a learning community that spans the globe. Such collaboration may also be useful in other disciplines, such as language and the arts.

The objective in this study was to understand the growth of a learning community as influenced by the use of collaborative software. By looking at the communication patterns of the students who used Manhattan Virtual Classroom, we may be able to determine how to use this tool to facilitate learning.

Research Questions

The initial guiding question of this study was: "How does collaborative software help in the formation of a learning community?" Everything that follows is commentary, explanation, or exposition of this basic question.

Sub-questions:

1. Does the use of collaborative software help in the formation of a learning community?
2. Did a learning community actually form, or were the students essentially a group of individuals who interacted with the professor and the course material, but not with each other?
3. What seemed to be the stages of community formation, if a community did, in fact, develop?
4. What is the effect on student perceptions of their learning at the completion of the first level computer science course if they have collaborative software available and make use of the software?
5. Did the students who used the collaborative software find the course to be easier or harder because of the use of the collaborative software?
6. By collaborating, did the students integrate the concepts of the class in a more meaningful way, thus reinforcing their understanding of the material?

Limitations

Krathwohl, in his textbook on educational research, (1998) wrote:

> In addition to building a strong chain of reasoning and meeting the requirements of internal validity (LP) and external validity (GP), every study must optimize three aspects: (1) the building of audience credibility, (2) the relative weighting of internal validity (LP) with external validity (GP), and vice versa. ...
>
> In addition, these characteristics must be optimized within the limits imposed on every study: (1) what can be ethically done, (2) what an institution will permit to be done in the name of research when its primary goal is service to clients, and (3) resource limits. (p. 188)

This study was conducted under the constraints of cost, hardware and software availability, and student population. The Computer Science Department of Concordia University Wisconsin approved this study, and allowed me to use existing hardware and software so to conduct the study. All research costs were borne by the researcher, thus the need for the use of existing hardware and Open Source software.

Having the source code available for the Manhattan Virtual Classroom also gave me several advantages for subsequent archival and later use of the gathered data.

Finally, the student population does make a difference. In the literature review we will see that student usage patterns of collaborative software change as students advance in a computer science program. Because the classes in this study are geared towards the traditional freshman student, some of the conclusions may not be the same if the study were of a senior level class or a class in the Adult Education department.

Among the limitations of this study are the circumstances under which the study was conducted. The two sections of CSC-150 - Foundations of Computer Science were taught in the Spring semester of 2004. Although the professor is highly familiar with technology, and had used portions of the collaborative software package, Manhattan

Virtual Classroom, in other classes, this was the first time the package had been used by him with undergraduate students. No computer science majors were in the two sections of this class. The students were given very little instruction on how to use the collaborative software, on the benefits of collaborative software, or the use of threaded discussions in general.

The students who were interviewed for this study accepted a general invitation to be interviewed. Hence they were self selected. Although they were self selected, there were no majors which were unrepresented in the interviews.

Definitions

Classical education: According to Veith and Kern (1997, p. 11), the classical education model is the basis of a liberal arts education. There were seven liberal arts in the middle ages, the trivium of grammar, logic, and rhetoric; and the quadrivium of mathematics, music, astronomy, and geometry. "The *trivium* and *quadrivium* are not discrete subjects. They are modes of learning."

The classical education model does not necessarily mean a return to the traditional means of teaching via lecture. Veith and Kern (1997, p. 78) call classical education schools "communities of learning" where teachers are coaches and mentors of their students. The Computer Science Department at Concordia University Wisconsin is built around this model of education.

Collaboration: Palloff and Pratt (1999, p. 110) wrote: "The collaborative effort among the learners helps them achieve a deeper level of knowledge generation while moving from independence to interdependence. In their article, *Making Distance Learning Collaborative,* Ellen Christiansen and Lone Dirckinck-Holmfeld (1995) postulate that the development of collaborative skills requires a means of study and an environment for study that '(A) lets a group of students formulate a shared goal for their learning process, (B) allows the students to use personal motivating problems/interests/experiences as springboards, (C) takes dialogue as the fundamental way of inquiry.'"

Constructionism: Seymour Papert (1993, p. 142) uses the term "constructionism" in relation to learning. He contends that knowledge is constructed by the learner, and that the teacher ought to provide opportunities for the learner to immerse himself / herself in a subject. Constructionism is active learning, hands on learning, where the student gets covered in dirt and chalk dust, gets scrapes on his / her hands and knees, and comes away exhausted.

Papert noted: "Constructionism also has the connotation of 'construction set,' starting with sets in the literal sense, such as Lego, and extending to include programming languages considered as 'sets' from which programs can be made, and kitchens as 'sets' with which not only cakes but recipes and forms of mathematics in use are constructed." (p. 142)

Constructivism: Constructivism, based on the work of Jean Piaget, says the individual takes information from the environment and assimilates it by extending existing knowledge. If the new information is very different from information that is already known, it cannot be assimilated because there is nothing similar to build upon. Mayer (1992, p. 288) wrote, "All cognitive growth, according to this view, depends on our taking in information that is slightly different from what we already know and then restructuring our knowledge to integrate both the old and new information; this process produces and improved cognitive structure, which will help us survive and function better."

Constructivism is the child watching the ant hill on a summer afternoon, a student developing a program using a newly learned language, or the Wright brothers looking at the cross section of a dove's wing and relating the shape to Bernoulli's work on hydraulics.

Mark Guzdial (1997), an associate professor at the Georgia Institute of Technology who has done extensive work on collaborative software, wrote: "The confusion that I and others have about these terms stems from (a) similar looking words and (b) meaning at different levels of the word construct. Piaget was talking about how mental constructions get *formed* , philosophical constructivists talk about how these constructions are *unique* (noun construction), and Papert is simply saying that constructing is a good way to get mental constructions built. Levels here are shifting from the physical (constructionism) to the mental (constructivism), from to philosophy to method, from science to approach to practice."

Group Patricia Wallace (1999, p. 57) wrote, "One succinct definition states that a group is a collection of two or more people who are interacting with and influencing one another." Groups and communities are related, as will be seen in the definition of learning community. Especially important in the definition of group is the idea of interaction.

Learning community: Palloff and Pratt (1999, p. 21) wrote, "People seeking commonality and shared interests formed groups and communities in order to pursue the interests that distinguished them from other groups." A learning community would be a group of people who have come together for the purpose of exploring some realm of human knowledge. Said group may exist for a weekend, as with a short seminar; for a semester, as with a traditional class; or for several years, as with a learning cohort. This community, this group of like-purposed people, may be gathered in a specific place or via computer mediated communications. As the definition of learning community and group are examined, interaction is an important concept. Hence, in this paper a learning community is defined as a group of people who are brought together for the purpose of learning, who interact with each other, and so influence each other.

Open Source or Free Software: Open Source or Free Software was used in this research. According to the Free Software Foundation (www.gnu.org, 2004), Free Software preserves the users' freedoms to run, copy, distribute, and modify software. For these freedoms to exist the source code for a program must be available. They define four different freedoms.

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1).
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3).

For the purposes of this paper, the terms *Free Software* and *Open Source* will be used interchangeably.

Social cognitive theory: Albert Bandura (1977, p. 11) wrote: “In the social learning view, people are neither driven by inner forces nor buffeted by environmental stimuli. Rather, psychological functioning is explained in terms of a continuous reciprocal interaction of personal and environmental determinants. Within this approach, symbolic, vicarious, and self-regulatory processes assume a prominent role. ... The capacity to learn by observation enables people to acquire large, integrated patterns of behavior without having to form them gradually by tedious trial and error.”

Chapter 2: Communities, Communication, and Learning

> Collaborative work in groups is a complex matter, with or without electronic technology. Collaboration is about more than simply the exchange of information. Collaborative work entails cognitive aspects of communication: Group members transmit, receive, and store information of various kinds, from each other and from various other sources. Collaborative work also entails emotional and motivational aspects of communication: Group members are also transmitting, receiving, and storing the affect and influence aspect of those same messages. (McGrath & Hollingshead, 1994, p. 7)

This chapter serves both as a review of the literature concerning collaborative software and as an historical background of online communication and collaboration. It begins with an overview of various learning models which may be supported by collaborative software, followed by a discussion of community formation and life cycle. Because online communication is not possible without hardware, the availability of computers to students is an important consideration, so computer availability is discussed. A review of the history of online communication is next so the technology of collaborative software may be better understood. Online communities, which can exist only if the underlying technology exists, are examined. Collaborative software is defined and findings of earlier research are presented and discussed. A review of the research is followed by a discussion of the importance of student interaction. The chapter ends with an explanation of one form of research organization, the Novak and Gowin "V" diagram, which will form the framework for subsequent chapters.

Learning Models

How do we learn? How do our minds form? MIT's Marvin Minsky (1985) wrote:

> In ancient times it was believed that the newborn mind started out just like a full-grown mind, except for not yet being filled with ideas. Thus children were seen as ignorant adults, conceived with all their future aptitudes. Today, there are may different views. Some modern theories see a baby's mind as starting with a single Self whose problem is to learn to distinguish itself from the rest of the world. Others see the infant's mind as a place containing a horde of mind-fragments, mixed together in a disconnect and incoherent confusion in which each must learn to interact and cooperate with the others so that they can grow together to form a more coherent whole. Yet another image sees the child's mind as growing through a series of layer-like construction stages in which new levels of machinery are based and built upon older ones. (p. 164)

Certainly Minsky (1985, p. 330) is talking of the various models of cognition and learning. He defines a model as "Any structure that a person can use to simulate or anticipate the behavior of something else."

Models, by their very nature, are incomplete or limited representations of reality as can be seen in the field of mathematics. Douglas Hofstadter, in *Gödel, Escher, Bach: An eternal golden braid* (1979/1999) wrote of Gödel's "Incompleteness Theorem":

> Therefore Gödel's Theorem had an electrifying effect upon logicians, mathematicians, and philosophers interested in the foundations of mathematics, for it showed that no fixed system, no matter how complicated, could represent the complexity of the whole numbers: 0, 1, 2, 3, ... Modern readers may not be as nonplussed by this as readers of 1931 were, since in the interim our culture has absorbed Gödel's Theorem, along with the conceptual revolutions of relativity and quantum mechanics, and their philosophically disorienting messages have reached the public, even if cushioned by several layers of translation (and usually obfuscation). There is a general mood of expectation, these days, of "limitative" results — but back in 1931, this came as a bolt from the blue. (p. 19)

Loosely paraphrased, Gödel's theorem states that within a closed system there are propositions which can not be proven. Thus the only way to prove some axioms of a closed system is to step outside of the system.

As we seek to understand human cognition and learning, we are dealing with a closed system. We are thinking about thinking. Because we are in a closed system, several approaches may be made to describe the system, but no approach is, in itself, complete or provable.

John Searle (1998) wrote of the problem of studying cognition and consciousness:

> The first difficulty in getting an account of consciousness arises out of the peculiar relation in which consciousness stands to observation. We cannot observe consciousness in the way we observe mountains and oceans because the only candidate for observation is the act of observing itself. We cannot make the distinction between the observation and the thing observed for consciousness itself, as we can for other targets of observation. This point has important consequences for the doctrine of introspection, as we will see. (p. 68)

Constructivism, constructionism, social learning theory, situated cognition theory, and even the traditional theories of learning have proponents who claim the models are accurate. There are elements of truth in each of these models, but models, according to Gödel's theorem, cannot completely describe the system of human cognition.

There are two approaches to learning models. The first is theoretical, the approach of the educational psychologist. The second is pragmatic, the approach of those who see no need to debate the various theories.

The Federal Aviation Administration provides an example of the second approach in the *Aviation Instructor's Handbook* (1999).

> Over the years, many theories have attempted to explain how people learn. Even though psychologists and educators are not in complete agreement, most do agree that learning may be explained by a combination of two basic approaches: behaviorism and the cognitive theories. ...
>
> Both models of the cognitive theory have common principles. For

> example, they both acknowledge the importance of reinforcing behavior and measuring changes. Positive reinforcement is important, particularly with cognitive concepts such as knowledge and understanding. (p. 1-1, 1.2)

In their textbook on learning theories, Bigge and Shermis (1992) wrote:

> Twentieth-century systematic learning theories may be classified into two broad families, namely, S-R (stimulus-response) conditioning theories of the behavioristic family and *interactionist* theories of the cognitive family. (p. 10)

They explained, "...S-R theorists — the neobehaviorists — conceive of learning as *conditioning* or *reinforcement* of behaviors, cognitive interactionists think of it as a *development of generalized insights* — understandings — which provide a potential guide for behavior." (p. 68)

The Table 1, derived from Bigge and Shermis (1992, p. 8), gives a brief overview of several different learning theories. Although both the FAA and Bigge and Shermis speak of two twentieth-century schools of thought as regards learning theory, the table begins with a third school, namely the mental discipline models that predate the twentieth century. In some respects the "classical education" movement is built on these earlier theories, thus these theories are also relevant to this discussion.

Class	Learning Theory / Psychological System	Conception of Moral/Actional Nature	Basis of Transfer of Learning	Emphasis in Teaching
Mental discipline theories	Theistic mental discipline / faculty psychology	*Bad-Active* mind substance continues active until curbed	Exercised faculties automatic transfer	Exercise of faculties of the mind
	Humanistic mental discipline / classical humanism	*Neutral-Active* mind substance to be developed through exercise	Cultivated mind or intellect	Training of intrinsic mental power
	Natural unfoldment or self-actualization / romantic naturalism or existential humanism	*Good-Active* natural personality to unfold	Recapitulation of racial history, no transfer needed	Negative or permissive education centered on feelings
	Apperception or Harbartianism / structuralism	*Neutral-Passive* mind composed of active mental states or ideas	Growing apperceptive mass	Addition of new mental states or ideas to a store of old ones
S-R (stimulus-response) or behavioristic conditioning theories	S-R Bond / connectionism	*Neutral-Passive* organism with many possible S-R connections	Identical elements	Promotion of acquisition of desired S-R connections
	Conditioning with no reinforcement / classical conditioning	*Neutral-Passive* biological organism with innate reflexes	Conditioned responses or reflexes	Promotion of adhesion of desired responses to appropriate stimuli
	Conditioning through reinforcement / instrumental conditioning	*Neutral-Passive* biological organism with innate reflexes and drive stimuli	Reinforced or conditioned responses plus stimulus and response induction	Environmental changes to increase the probability of desired responses
Interactionist theories	Goal insight / Gestalt psychology	*Neutral-Interactive* individual in sequential relationships with environment	Transposition of tested generalized insights	Promotion of learning by aiding students in developing high-quality insights
	Linear cognitive interaction / social cognitive theory	*Neutral-Interactive* purposive person in sequential relationships with environment	Expectancies that result from reciprocal person-environment interaction	Development of observationally reinforcing components of modeled learning
	Cognitive-field interaction / field psychology or positive relativism	*Neutral-Interactive* person in simultaneous mutual interaction with environment	Continuity of life spaces, experience, or insights	Help students insightfully restructure their life spaces — contemporaneous situations

TABLE 1. Pre-Twentieth Century Learning Theories
From Morris L. Bigge & S. Samuel Shermis *Learning Theories For Teachers, 5e.*

Mental Discipline Theories

Bigge and Shermis (1992, p. 21) wrote of these theories, “All were developed as non-experimental psychologies of learning. That is, their basic orientation is philosophical or speculative.”

Richard Mayer (1992), a professor of psychology at the University of California in Santa Barbara, wrote:

> Emphasis on teaching “good habits of mind” in American schools dates back to the Boston Latin School, founded in 1712, as well as the many subsequent Latin schools that where still going strong a century ago. The premise underlying the Latin school approach was that learning well-structured subjects such as Latin, Greek, and geometry would help improve the students; minds in general, enabling them to become better problem solvers in all domains. ...
>
> This classical approach to the teaching of thinking was based on conceptions of what to teach, how to teach, and where to teach that seem to conflict with current cognitive views of thinking... (p. 363)

There has been a resurgence in interest in this particular approach to learning. Dorothy Sayers (1947), wrote:

> The syllabus was divided into two parts: the Trivium and the Quadrivium. The second part — the Quadrivium – consisted of “subjects,” and need not for the moment concern us. The interesting thing for us is the composition of the Trivium, which preceded the Quadrivium and was the preliminary discipline for it. It consisted of three parts: Grammar, Dialectic, and Rhetoric, in that order.

Veith and Kern (1997) developed the trivium model as it relates to modern education.

> The *trivium* employs the terminology of language and is comprised of grammar, logic (sometimes referred to as “dialectic”), and rhetoric. Anyone who wants to learn a language must first master its grammar: this is the structure, vocabulary, rules, and conventions that constitute language. But grammar is not enough. To use language you must also learn how to think in that language. This is the role of logic, which makes possible the “dialectical” give-and-take of conversation.

> Finally, to have a genuine facility in language, the student must develop a capacity to speak and write his own ideas in a way that is compelling and persuasive. This is the "rhetoric" of language. (p. 11)

According to Veith and Kern (1997):

> The trivium is a paradigm for the mastery of language. But it applies to far more than language. *Every subject* has its grammar, logic, and rhetoric. To be educated in any discipline, you must 1) know the basic facts (grammar); 2) be able to reason clearly about it (logic); and 3) apply it personally in an effective way (rhetoric). (p. 12)

Behavioristic Theories

Bigge and Shermis (1992) explain stimulus-response learning as follows:

> A child or youth is something to be molded in the proper fashion. Learning primarily is a process within which both verbal and nonverbal behaviors are changed. Such behaviors are inculcated by adults telling, showing, directing, guiding, arranging, manipulating, rewarding, punishing, and, at times, coercing the activities of children and youth. Accordingly, teaching is a matter of adults setting behavioristic environmental conditions — stimuli — to ensure that the students accomplish educational goals. (p. 76)

In further explanation they wrote, "Nearly all S-R condition theorists are alike in their emphasis on a mechanical treatment of stimuli and responses. Most agree that at no time is purposiveness to be attributed to human behavior" (p. 79).

Cognitive Interactionist Theories

Glover and Bruning (1990) wrote in *Educational Psychology: Principles and Applications*:

> What does modern cognitive psychology stress? According to DiVesta, "the emphasis is on the total instructional event of which the learner is a part. The situational demands, the characteristics of the learner, the task demands, the purpose of the learner, and so on *interact* to determine the quality and texture of an event such as a teaching or learning episode." Cognitive psychology emphasizes the

> active and constructive role of the student in the context of a learning situation. Students construct meaning and create their own realities, rather than responding in an automatic way to their environments. (p. 38)

Piaget and Vygotsky come to mind as theorists who espouse this model. Papert, especially in the development of the LOGO computer language, also is a proponent of this viewpoint.

What Does This Mean?

All three classes of learning theory may be found in modern classrooms. Indeed, aspects of mental discipline, behavioristic, and cognitive interaction may be found in a single classroom and advocated by a single teacher. As an example, the teacher in a "classical education" school wants students to turn in homework on time. To ensure this behavior, the teacher awards little or no credit for late homework, thus providing a stimulus (good grades) which provokes a response (timely completion of homework). Yet the same teacher encourages introspection, peer communication, and collaborative study in the pursuit of knowledge, thus demonstrating an interactionist understanding of learning.

No single theory can be used to explain the complexity of learning. Minsky (1985) explained:

> Each child learns, from time to time, various *better ways to learn* — but no one understands how this is done. We tend to speak about "intelligence" because we find it virtually impossible to understand how this is done from watching only what the child *does.* (p. 229)

A common thread which seems to run through all of the theories is the need to communicate. Teachers may simply present a concept to their students, or they may ask students to write and reflect on an event. Students respond to the teacher and ask questions or make comments. Students interact with each other in an attempt to understand those concepts first discussed in the classroom. No matter if the messages are teacher-to-student, student-to-teacher, or student-to-student, the communication process is central to learning.

On a final note, Donald Knuth (2001, p. 94) wrote, "Even in formal scientific and mathematical work, the elegance of a theorem and the 'ah ha!' insights defy explanation. In general, I don't see why there should only be one way to understand things." Each of the various learning theories which have been discussed have validity. Although different models may be used at different times depending on the situation, in the end each explains an aspect of learning.

Thus, in this study I will approach the interactive, threaded discussion tool as an adjunct for use by proponents of any of the schools of educational psychology. The tool is neutral in that it does not limit teachers or students to one mode of thinking or to one model of learning.

Communities and Small Groups

Many of the books on community formation and online communication are strangely silent on the definition of a community or group. Few mention the stages of group development or the basics of group dynamics. Yet such definitions are important for an understanding of community building using collaborative software.

A community community is a group, but a group may not become a community, as Amy Jo Kim (2000, p. 28) wrote. "A community is a group of people with a shared interest, purpose, or goal, who get to know each other better over time." Without the formation of a group, the formation of a community is impossible.

Marvin Shaw (1981, p. 9) defined a group "as two or more persons who are interacting with one another in such a manner that each person influences and is influenced by each other person." In commenting on Shaw's definition, Charles Palazzolo (1988) wrote:

> Once the importance of interaction within the group is established, it must be considered that the process of interaction involves a system of relations in which individuals become meaningful participants in a kind of association which is lasting and which is committed to the achievement of a goal of a set of goals. (p. 7)

If a group is defined as persons interacting with each other where they influence each other (Shaw) for the purpose of attaining a goal (Palozzolo), does the group have a life cycle? Various researchers (Shaw, 1981, p. 105; Mills, 1967, p. 13; Lacoursiere, 1980, p. 25) speak of the stages of group development. The stages typically are oriented around group formation, dissatisfaction, resolution, production, and termination. Thus the group is formed, either by choice or by chance, for some given purpose. As the group organizes to accomplish that purpose, some disagreement as to process and procedure, or possibly goals, is bound to occur. When the conflict is resolved the group solves whatever problem is set before it, the group accomplishes the task as they perceive it. Some groups may never get to the stage where they accomplish the group task as they perceive it, however. Finally, with the task accomplished or abandoned, the group disbands or begins the cycle anew with a new goal in mind.

Theodore Mills (1967) describes the group as a living being.

> The group is like an organism — a biological organism. It forms, grows, and reaches a state of maturity. It begins with a set of ideas, potentialities, limitations — and in the course of development evolves a particular pattern of behavior, a set of indigenous norms, a body of beliefs, a set of values, and so on. Parts become differentiated, each assuming special functions in relation to other parts and the whole;

> yet these parts are integrated by a complex set of connections, interchanges, and coordinating mechanisms. As a group approaches maturity it becomes even more complex, more differentiated, more interdependent, and more integrated. (p. 13)

Paloff & Pratt (2003) wrote of community:

> Our own definition of online community and how it forms in an online course has not varied significantly since we first presented it and is not dissimilar from that of Preece. We would add a couple of features, however, that distinguish the online learning community from an online community such as a listserv or online group where people meet to share a common interest. Engaging in collaborative learning and the reflective practice involved in transformative learning differentiate the online learning community. In addition, we suggest the following outcomes to determine whether community has formed online and become an integral part of the course:
>
> - Active interaction involving both course content and personal communication
> - Collaborative learning evidenced by comments directed primarily student to student rather than student to instructor
> - Socially constructed meaning evidenced by agreement or questioning, with the intent to achieve agreement on issues of meaning.
> - Sharing of resources among students
> - Expressions of support and encouragement exchanged between students, as well as willingness to evaluate the work of others (Palloff and Pratt, 1999, p. 32) (p. 17)

Access to Computers

Few inventions have become so quickly become a part of mainstream society as the personal computer that is connected to the Internet. The World Wide Web was invented in 1992. The first graphical browser, Mosaic, was released by the National Center for Super Computers in 1993 (Ceruzzi, 1998, p. 300; Campbell-Kelly, 2004, p. 267; Berners-Lee, 2000). Microsoft (2003) introduced the Internet Explorer browser in 1995 as an extension to Windows 95, thus beginning the push for more online information.

According to the National Center for Education Statistics (2003, Table 421), only 35% of public (K-12) schools in the United States had Internet access available in 1994, as opposed to 99% in 2002. As the students who were in grade school and high school in 1994 begin to fill college classrooms, computer usage ought to increase at the university level.

Although personal computers were introduced in the mid-1970s, they were not generally used either in schools or homes until the 1990s. Most computers, both personal and mainframe, were stand-alone units or connected only to systems in close proximity (same room). In the late 1980s and early 1990s, the transmission rates of dial-up modems jumped from 1200 characters per second to over 50,000 characters per second due to improvements in both telephone and modem technologies.

With the decreasing cost of computers, improved software, and faster access, more schools and homes had computers available. Yet in the fall of 2004, only 85% of the students at Concordia University Wisconsin had personal computers available in their dorm rooms, according to Patrick Delancy, the Concordia University Wisconsin computer lab administrator (personal communication, November 29, 2004). Resident students who do not have a computer available in their dormitory room do have access to several public access computer labs in the classroom buildings of the university. No one has determined the how many commuter students do not have computers in their homes or if they use the computer labs at the university.

According to Brown and Duguid (1995):

> The centrality of conversation helps explain why the Internet is such a significant phenomenon. Previous communications technologies—books, film, radio, television, telephones — have all supported distance. But they allowed primarily either monologues or one-to-one conversations. Communities, however, thrive on many-to-many conversations, which, even in the technologically rich twentieth century, have for the most part only been possible in face-to-face situations. So the campus and the workplace, which bring people together, have long been crucial sites for learning. Technology in general and the Net in particular now offer low cost ways to hold many-to-many conversations among people who are no longer in the same place. ...
>
> E-mail, usenets, bulletin boards, and listserve mail lists get their usefulness from the way they transmit transient comments and allow them to be captured to make up an archive. Of course, not all comments are illuminating, but an archive is helpful in showing both dead ends and possible developments. Participants see for themselves the ebb and flow of exchange and its history. (section 10)

Online data communications cannot be dated from the invention of the computer. Online data communications cannot be dated from 1883 when West Union installed an automatic telegraph (Negroponte, 1995, p. 190). Online data communications can be dated from May 24, 1844, when Samuel F. B. Morse typed, “What hath God wrought?” in the first public demonstration of the telegraph (Brown and Dugid, 2002, p. 18; US Library of Congress, 2001). However, online data communications became much more pervasive and important with the development of the ARPANET, the first packet switched computer network, in the late 1960s (Hafner & Lyon, 1996; Negroponte, 1995; Lessig, 2001).

Undergirding the discussion of electronic mail, bulletin boards, and the Internet is the gradual widening of the scope of electronic messaging. Electronic mail began with messages passed between users on a single machine, and slowly expanded to systems located in close proximity (Hafner & Lyon, 1996, p. 13). As more computers acquired the ability to receive and place telephone calls, mail and other messages passed from system to system (Salus, 1994, p. 105). Eventually, with the invention of packet switched networks, which will be discussed below, electronic information could be passed between multiple systems quickly, without the need to place individual telephone calls between systems. The speed of communications also increased greatly as networking became more prevalent.

Electronic Mail and Bulletin Boards

Several online resources have become famous in their own right. *Zen and the Art of the Internet* is one such work that helped define the intellectual bounds of the Internet, and helped in the transition from the early ARPANET to the World Wide Web. Its author, Brendan Kehoe (1992, p. 11), wrote, “The desire to communicate is the essence of networking. People have always wanted to correspond with each other in the fastest way possible, short of normal conversation. Electronic mail (or *e-mail*) is the most prevalent application of this in computer networking.” “Researchers saw each other at technical conferences and talked by phone; as early as 1965 some had even begun using a form of electronic mail to trade comments, within the very limited proximity of their mainframe computers,” wrote Hafner and Lyon (1996, p. 13) in their history of the Internet. Electronic mail was available on the time-sharing systems of the 1960s between users on the same system, but the first documented e-mail between two systems was sent in 1972 (Hafner & Lyon, 1996, p. 191).

Although e-mail is still one of the most common uses of the Internet, other applications also are gaining in popularity. "The Internet is at once a worldwide broadcasting capability, a mechanism for information dissemination, and a medium for collaboration and interaction between individuals and their computers without regard for geographic location," wrote Leiner et al. (1997, p. 102).

Closely related to electronic mail, and building on e-mail technology, is the USENET. This early collaboration tool was built by mirroring electronic mail messages which were sent to a reflector site. When a message is posted to the USENET, it is placed on a local news server. At various times during the day, as determined by the system administrator, these messages are sent to other USENET news servers. Gradually each news server receives a copy of the message as news server talks to news server.

In 1979, two Duke University grad students hooked two computers together to share information (Kehoe, 1992, p. 33; Kroll, 1989, p. 5). According to Quarterman and Hoskins (1986), in their ACM article which looked at the networks of the mid-1980s, the USENET

> ...combines the idea of mailing lists as long used on the ARPANET with bulletin-board service such as has existed for many years on TOPS-20 and other systems, adding a freedom of subject matter that could never exist on the ARPANET, and reached a more varied constituency. While chaotic and inane ramblings abound, the network is quite popular.
>
> The USENET news network is a distributed computer conferencing system bearing some similarities to commercial conferencing systems like CompuServe, though USENET is much more distributed. ... USENET is currently defined as the set of hosts receiving the newsgroup *news.announce*. There are about a dozen hosts that constitute the backbone of the network, keeping transit times low by doing frequent transfers among themselves and the hosts they feed. Since these hosts bear much of the burden of the network, their administrators tend to take a strong interest in the state of the

> network. Most newsgroups can be posted to by anyone on the network. (p. 958)

The first computer bulletin board system (CBBS) was written and went online in early 1978 (Christensen and Suess, 1989). During the Chicago blizzard of January, 1978, Ward Christensen and Randy Suess decided to write some computer code rather than shovel their driveways. Tying together the newly developed Hayes modem and a spare S-100 computer system, the two built a messaging system. Soon bulletin board systems appeared in Portland and other major cities (Quarterman & Hoskins, 1986, p. 964; Christensen & Suess, 1989).

Information Providers - CompuServe and Others

As more personal computer owners purchased modems, various entrepreneurs realized there was money to be made in selling information. One of the first public information providers was CompuServe. According to the CompuServe web site (2003):

> Founded in 1969 as a computer time-sharing service, Columbus, Ohio-based CompuServe drove the initial emergence of the online service industry. In 1979, CompuServe became the first service to offer electronic mail capabilities and technical support to personal computer users. CompuServe broke new ground again in 1980 as the first online service to offer real-time chat online with its CB Simulator.

Quarterman and Hoskins (1986) wrote of CompuServe and other commercial providers:

> Commercial networks sell services to outside users for profit. Many are in effect common carriers like the telephone system. Administration is always centralized, though execution may be delegated. Fees are usually charged to individual persons or organizations on the basis of connect or CPU time used.
>
> CompuServe, THE SOURCE, and other such services are not really networks. They consist of a few large computers closely coupled into a large distributed system and are accessed just like home personal-computer bulletin-board systems, except that users get bills. (p. 965)

Internet

Collaborative software cannot be used without a form of computer mediated communication. Even early educational systems, such as PLATO at the University of Illinois in Champaign-Urbana, relied on terminal-to-host communications. (Ceruzzi, 1998, p. 173).

The Internet started in part as a Cold War response to the threat of nuclear attack on the communications centers of the United States. (Hafner & Lyon, 1996, p. 58; Lessig, 2001, p. 31) Through the development of redundant message routing and the division of messages into small packets of data, communication among military organizations could be assured even though parts of the network might be out of service. The ARPANET, the first packet switched network, was up and running by early 1970 (Hafner & Lyon, 1996, p. 153; Quarterman & Hoskins, 1986, p. 943). As more computer networks were developed, various companies and universities acted as gateways between networks. Thus was born the Internet.

Quarterman and Hoskins (1986, p. 932) reported, "There are*internets* of smaller networks communicating with one another through the same protocols." The Internet is the combination of many networks all using TCP/IP (Transmission Control Protocol / Internet Protocol) for addressing and transmission of packets. This was described in Hunt (1998):

> The TCP/IP protocols were adopted as Military Standards (MIL STD) in 1983, and all hosts connected to the network were required to convert to the new protocols. To ease this conversion, DARPA [Defense Advanced Research Projects Agency] funded Bolt, Beranek, and Newman (BBN) to implement TCP/IP in Berkeley (BSD) UNIX. Thus began the marriage of UNIX and TCP/IP.

> About the time that TCP/IP was adopted as a standard, the term *Internet* came into common usage. In 1983, the old ARPANET was divided into MILNET, the unclassified part of the Defense Data Network (DDN), and a new, smaller ARPANET. "Internet" was used to refer to the entire network: MILNET plus ARPANET. ...
>
> The Internet has grown far beyond its original scope. The original networks and agencies that built the Internet no longer play an essential role for the current network. The Internet has evolved from a simple backbone network, through a three-tiered hierarchical structure, to a huge network of interconnected, distributed network hubs. It has grown exponentially since 1983—doubling in size every year. Through all of this incredible change one thing has remained constant: the Internet is built on the TCP/IP protocol suite. (p. 2)

There is an uncanny connection between the development of the Internet and the theorists who developed some of the concepts of constructionism, the society of mind, and artificial intelligence. According to Hafner and Lyon (1998, p. 85, 87), Marvin Minsky, John McCarthy, and Seymour Papert all worked for a time during the 1960s at Bolt, Beranek, and Newman, the company which developed packet switching. We can only speculate on the effect which the culture at BBN, at the time packet switching was being developed, had on Minsky, Papert, and McCarthy. Did the interwoven nature of the network topology spring in part from the early concepts of Minsky's society of mind, or did the network topology help clarify Minsky's ideas? Did Papert gain insight into learning from discussions of how we communicate and from watching the development of ideas, or did his theories help define the communications protocols first used in the network? Did McCarthy's work with recursion and mathematics in 1960 as he developed the LISP computer language (Pratt & Zelkowitz, 2001, p. 420) later influence the design of the packet switched network?

World Wide Web

Without the availability of the World Wide Web, collaborative tools such as WebCT (2004), Blackboard (2004), and Manhattan Virtual Classroom (Narmontas, 2003b) would not exist. The Internet made possible the development of the application known as the World Wide Web.

Lawrence Lessig (2001) wrote:

> If you're free from geekhood, you are likely not to distinguish the WWW from the Internet. But in fact, they are quite distinct. The World Wide Web is a set of protocols for displaying hyper-linked documents linked across the Internet. These protocols were developed in the late 1980s by researchers at the European particle physics lab CERN — in particular by Tim Berners-Lee. These protocols specify how a "Web server" serves content on the WWW. They also specify how "browsers" — such as Netscape Navigator or Microsoft's Internet Explorer — retrieve content on the World Wide Web. But these protocols themselves simply run on top of the protocols that define the Internet. These Internet protocols, referred to as TCP/IP, are the foundation upon which the protocols that make the World Wide Web function — HTTP (hypertext transfer protocol) and HTML (hypertext markup language) — run. (p. 41)

Open Source or Free Software

Open Source, or Free Software, as a formal idea, can be traced to the work of Richard Stallman in 1984. The same time that AT&T was broken up by Judge Harold Greene as the result of a Justice Department lawsuit, January 1, 1984, (US v. AT&T, 1982) the freely available source code for the UNIX operating system was given restricted status by AT&T. In response to this move, and to codify the "hacker ethic" which had grown around academic computing, Stallman and the Free Software Foundation developed the GNU Public License. Sam Williams (2002, p. 14), Stallman's biographer, wrote: "In simplest terms, the GPL locks software programs into a form of communal ownership—what today's legal scholars now call the "digital commons"—through the legal weight of copyright. Once locked, programs remain unremovable [programs cannot be re-licensed under a proprietary scheme and removed from the Free Software category]. Derivative versions must carry the same copyright protection—even derivative versions that bear only a small snippet of the original source code."

Free software specialist Eric Raymond coined the term "Open Source" in February, 1998, when Netscape released the Navigator 5.0 source. He wrote (1999, p. 86), "The subsequent call to the hacker culture to exploit this unprecedented opportunity and to re-label its product from 'free software' to 'Open Source' was met with a level of instant approval that surprised everybody involved."

The Free Software Foundation's (www.gnu.org, 2004) definition of free software says: "Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software." On the same web site they define four different freedoms, several of which require the source code to be available:

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1).
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3).

Much of the Internet is built on Free Software. When Tim Berners-Lee released the Hypertext Protocol, he did so under the GNU Public License, the *copyleft* in which the Free Software Foundation defines free software. The Linux operating system is also released under the GNU Public License, as is the Manhattan Virtual Classroom, the programs used for this research.

With the Internet for communications, with HTML and HTTP for serving and reading content, and with the concepts of electronic mail and discussions, the pieces are in place for the development of collaborative software. We now have the infrastructure necessary to allow students to communicate and to form online communities.

Online Communities

According to Hiltz and Wellman, writing in the *Communications of the ACM* (1997):

> Computer mediated communication can enable people with shared interests to form and sustain relationships and communities. Compared to communities offline, computer supported communities tend to be larger, more dispersed in space and time, more densely knit, and to have members with more heterogeneous social characteristics but with more homogeneous attitudes.
>
> Despite earlier fears to the contrary by those who worry about the possible dehumanizing effects of computers, online communities provide emotional support and sociability as well as information and instrumental aid related to shared tasks. (p. 44)

Ad-hoc Communities

Howard Rheingold (1993), writing about one of the first public computer bulletin board systems, the WELL, discussed the formation of various unplanned communities.

> Then several technology-savvy Deadheads started a Grateful Dead conference on the WELL. GD, as it came to be known, was so phenomenally successful that for the first several years, Deadheads were by far the single largest source of income for the enterprise. Because of the way the WELL's software allowed users to build their own boundaries, many Deadheads would invest in the technology and the hours needed to learn the WELL's software, solely in order to trade audio tapes or argue about the meaning of lyrics — and remain blithely unaware of the discussions of politics and technology and classical music happening in other conferences. Those Deadheads who did "go over the wall" ended up having strong influence on the WELL

> at large. But very different kinds of communities began to grow in other parts of the technological-social petri dish that the Deadheads were keeping in business. (p. 49)

Although this research is concerned with intentional educational communities, the topic of ad-hoc communities deserves its own study. The USENET archives are located at http://www.google.com. Google has also provided tools to help people continue to use this older communications medium.

A quick browsing of the older news groups, such as net.aviation, shows a core of participants who shared ideas, information, and encouragement. They came together because of the topic and continued to post messages because of their interest. Occasionally participants met face-to-face so to better know each other.

Quarterman (1986) wrote:

> Certain newsgroups, mailing lists, bulletin boards, and SIGs have reliable followings that form social groups. These range from groups interacting strictly in pursuit of technical goals to others interacting for the sake of interaction, to still others for whom the networked interaction is an aspect of or leads to outside interaction. (p. 966)

Intentional Educational Communities

Palloff and Pratt (2001) wrote:

> Successful learners in the online environment need to be active, creative, and engaged in the learning process. In other words, they need to be "noisy learners", or learners who are visibly engaged with one another an with the generation of knowledge. Some students who are not noisy learners in the face-to-face classroom can flourish online because they have the luxury of time for reflection and response and do not have to compete with more extroverted students in order to be heard. However, it cannot be assumed that students will engage with one another in the learning process; this must be taught. (p. 107)

Where many online communities simply happen (Kim, 2000; Rheingold, 1993), the educational community exists for the purpose of learning. Indeed, the sense of community is very important as Palloff and Pratt (1999, p. 29) also wrote, "In distance education attention needs to be paid to the developing sense of community within the group of participants in order for the learning process to be successful. The learning community is the vehicle through which learning occurs online. ... Without the support and participation of a learning community, there is no online course."

In an educational community, the role of the teacher and student change. Where in a traditional classroom the professor is seen as the expert who dispenses information, the "sage on the stage" model of learning, in computer mediated communication (CMC) the teacher gives up some power to the student. Researchers in the ACM special interest group for computer science education, Wolz, et. al. (1997), wrote:

> Work from the Calculus Reform Movement and elsewhere indicates that classes are most effective when students are actively engaged in the material. A lecture format may allow an instructor to cover a large amount of material efficiently, but this does not automatically translate into the students learning that material effectively.
>
> Collaborative learning and CMC provide two ways to address this goal. In either setting, the focus for work is on the students, and the instructor serves as facilitator and mentor. The instructor gives up much control of a class when the class format emphasizes group activity rather than lecturing. Questions from class members dictate the class format, content, and schedule. A teacher's role then is to respond quickly to questions to coach individual groups, to identify common difficulties, and to suggest new approaches. ...
>
> In a collaborative class setting that emphasizes student involvement and group participation, students are less dependent upon the teacher and they learn how to learn. When the teacher serves as coach and mentor, independent thought is encouraged, and students come to understand they can succeed. They become more proficient at reading and experimenting, and they develop effective strategies in mastering new ideas. (p. 52)

There are advantages to online learning, the use of collaborative software, but these advantages come with both experience and familiarity. Harasim, Hiltz, Teles, and

Turoff (1996) wrote:

> One of the greatest advantages of online learning networks is the opportunity they provide for students to work together. Collaborative learning means that learners will be exposed to a wide variety of points of view and that group members can improve one another's understanding by pointing out omissions or logical errors. Group learning and the obligations it entails may not be familiar or comfortable at first. Students may not be used to talking in class or may worry that their ideas will be ridiculed. They may be inexperienced or wary of working in groups, and it may seem difficult to have to cooperate with other students rather than work individually. A positive attitude and considerate behavior will generally help all students become productive group members in the online environment. (p. 207)

Collaborative Software

Wolz, et. al, (1999) remind us:

> Collaborative learning is embedded in constructivism. In the collaborative learning environment, the social context is provided by small groups, and members create a community of learners whose goal is to construct knowledge through common effort. In this new situation both teacher and student roles have to be redefined. The introduction of CMC tools in the educational setting in order to support collaborative learning will itself have an impact on the definition of these roles as new teaching/learning, technical, and social skills have to be developed. (p. 56)

The software which facilitates the communication between student and student or student and teacher is often called collaborative software. A course management system, such as BlackBoard, may include collaborative modules. Among the several collaborative software systems available is WebCT. Goldberg (1997b) describes the development of this tool at the University of British Columbia.

> In order to facilitate the construction of similarly sophisticated web-based course environments and thus further experimentation, we then went on to develop WebCT (World Wide Web Course Tool). WebCT is an authoring environment for the construction of Web-based

> courses, and requires no technical expertise on behalf of the course designer. It provides an interface that facilitates course construction (content organization, course look and feel, etc.) as well as a set of educational and administrative tools. These tools include student management and access control, grade storage and distribution, a conferencing system, a chat area, electronic mail, group presentation areas, student self-evaluation, on-line quizzes, searchable glossary, student progress tracking, course content searching, and more. (p. 127)

Other course management systems, such as Blackboard, provide a similar array of features. Narmontas (2003) wrote of the capabilities of Manhattan Virtual Classroom, the collaborative software used in this study:

> When you enter your username and password correctly on your classroom's login screen, you will be entering a private environment where you, the teacher, can:
>
> - Provide your students with handouts, notices, lecture materials, interactive self-tests, and web sites to visit.
> - Assign homework for your students to complete, receive the work they do in response to those assignments, and provide feedback.
> - Issue multiple-choice and short answer exams.
> - Exchange private messages with your students.
> - Host discussions with the entire class, or with teams of students.
> - Keep students apprised of their grades.
> - Engage in live online "chats" with your students.
> - Track which students are using the system and when.
>
> (p. 1)

Collaborative software is an outgrowth of traditional computer communications applications. Added to the online chat, the bulletin boards, and electronic mail are provisions for security and participation grouping. No one, except the administrator, has access to every area within the collaboration system. Students and faculty can use this collaborative software knowing that the discussions and comments have the same security level as do classroom discussions. Also, each module within the system has a similar look and feel which helps make the software easier to use.

Collaborative Software in Practice

The course management systems such as Manhattan Virtual Classroom, WebCT, and Blackboard have modules which use a conversational paradigm of communication. Students and instructors engage in a dialog concerning the issues raised elsewhere within the class.

Marlene Scardamalia and Carl Bereiter (1993, p. 37, 1996, p. 36), professors at the Centre for Applied Cognitive Science, Ontario Institute for Studies in Education, have been at the forefront of developing "computer-supported intentional learning environments" (CSILE). They (1996) described CSILE as:

> Knowledge building ... can be distinguished from most of school learning that focuses on individual assignments and various other individual displays of knowledgeability. In CSILE, the focus is on collective responsibility and continual advancement of ideas in a student-generated database. (p. 36)

According to Scardamalia (2004), in 1995 the concept of CSILE was re-engineered and renamed Knowledge Forum.

> The heart of CSILE/Knowledge Forum is a multimedia community knowledge space. In the form of notes, participants contribute theories, working models, plans, evidence, reference material, and so forth to this shared space. The software provides knowledge building supports both in the creation of these notes and in the way they are displayed, linked, and made objects for further work. (p. 2)

The CSILE paradigm is focused on creating various knowledge representations (text, graphics, etc.) so to add to a knowledge base. In this respect the CSILE paradigm seems to be related to other online collaboration systems such as the Wiki, a web based environment were anyone can add or edit content, and the similar CoWeb as described by Carlson, et. al. (1996).

Carlson chronicled the creation and use of a collaborative tool, the CoWeb, from the design through the first few years of its operation. Carlson et al. (1996) wrote:

> Cognitive science has shown that the "transmission model" of learning is wrong — students do not learn by simply receiving information. Rather, people learn *constructively,* which is to reflect on material, interact with it, and create an understanding. Thus, for courseware to be effective on the WWW, it must allow and even encourage students to be engaged and reflect on interesting situations. (p. 290)

They proceeded to discuss the construction of a collaborative WWW based learning environment known as a Wiki, which is a web site which allows any user to add or modify pages. In 2000, Guzdial, Rick, and Kerimbaev (2000) described the operation of the collaborative tool, Co-Web, and discussed the first two years of operation. Over 120 classes had used the tool. They wrote of the CoWeb:

> By constantly having user feedback, we were able to develop the software to fit the needs of the actual users, some of whom we would not have envisioned without this development cycle. This iterative design methodology allowed us to evolve a flexible open-ended collaboration tool that is quite popular and works in a variety of contexts. (p. 261)

Manhattan Virtual Classroom, WebCT, and other course management systems do not support the CSILE or Wiki model of interactive learning. Instead they use threaded discussion groups or internal electronic mail, in much the same way as a Bulletin Board System.

Participation

Two groups seem to be at the forefront of early collaborative software research using the conversational model. The University of British Colombia and Georgia Tech both figure prominently in the literature.

Goldberg (1997) documented the experiences of the Computer Science Department in using WebCT for teaching an introductory class. He wrote:

> One of the new WebCT-based courses is a first-year course that introduces computer science and uses C++ as the language of experimentation. This is a lecture-based course and has an enrollment of approximately 650 students. The on-line environment for the course contains a complete set of course notes, tutorials and assignments, an extensive glossary, and uses many of WebCT's features including the conferencing system and the chat area. Students in the first year at the University of British Columbia are, in many ways, quite different from their more mature third year counterparts. Our experience has shown that they tend to be less likely to participate in class and take advantage of the resources available to them. The reason for this could be a combination of the fact that they are new to the university environment, and that the class sizes tend to be much larger in the first year than in third. Regardless of the reasons, we wanted to know:
>
> 1. whether the availability of a web-based resource would affect their experience in the same way that it did for the third year students we studied, and
> 2. whether the students would make as much use of the web-based resource as did the third year students. (p. 127)

Goldberg (1997, p. 128) also reported that 82% of the first-year students did not make any postings to the course web based collaboration tool. Of the 18% who posted, they averaged 1.7 postings per student, yet 62% of the first year students surveyed felt a stronger sense of community was formed through the use of the online communication tools.

Rita-Marie Conrad, an online course designer for Nova Southeastern University, and J. Ana Donaldson, assistant professor of educational technology at the University of Northern Iowa, wrote of participation (2004):

> The student's role as an engaged learner develops over time. Interaction and collaboration is not intuitive to many adult learners who have been educated in a predominantly lecture-based environment. Initially, a learner may be more comfortable in a passive student role and will need guidance and the opportunity to become more involved in an online learning environment. (p. 9)

Luca and McLoughlin (2004) reported on the use of collaborative software at the

Edith Cowan University in Perth, Western Australia. In their study of final year multimedia students enrolled in the "Industry Project Development" course, 73 students posted 1437 messages during 13 weeks. They reported the use of collaborative software "appears to have been quite successful in terms of student participation." Fifteen percent of the course grade was based on bulletin board participation.

Other Research Concerning Collaborative Software

In the past decade as personal computers became more available and threaded discussion groups were adapted for education, there have been a number of studies and books about using WebCT and similar programs. The bibliography lists a number of books and articles which deal with collaborative software or the electronic classroom.

Aside from the three studies previously quoted (Carlson, et. al. (1996), Goldberg (1997), Luca & McLoughlin (2004), few of the books or articles discuss the quality or quantity of student discussion. These books and articles give suggestions for enhancing participation (Conrad & Donaldson (2004), Wolz, et. al. (1997), Palof & Pratt (2001)) but do not speak of the evolution of participation as did the studies previously cited.

The V Diagram: A Framework for Research

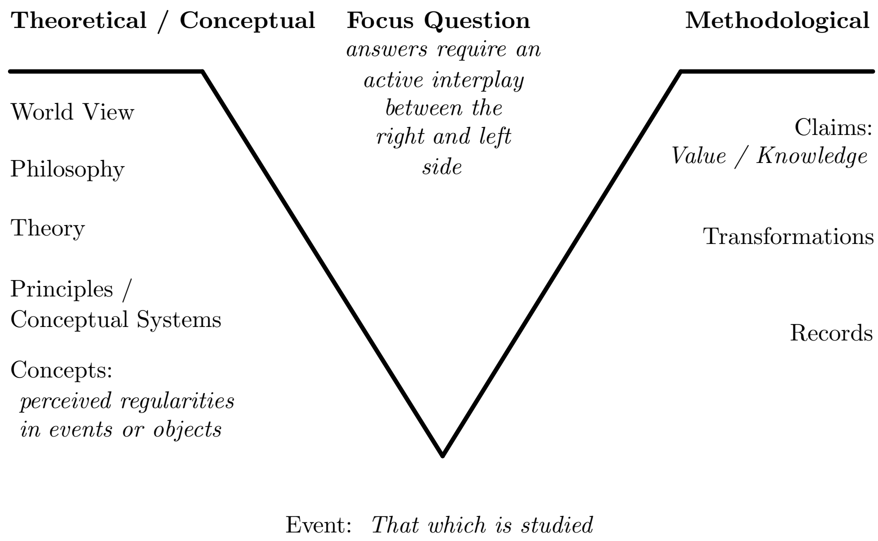

Figure 1. V Diagram - based on Novak / Gowin (1984)

Qualitative researchers, Novak and Gowin (1984), discussed an organization scheme for research which uses a Vee shaped diagram which is surrounded by questions or statements. They explained:

> Concepts operate in an explicit way to select the events or objects we choose to observe and the records we choose to make. If our concepts are inadequate or faulty, our inquiry is already in difficulty. If our records are faulty, then we do not have the *facts* (valid records) to work with and no form of transformation can lead to valid claims. The Vee helps us to see that although the meaning of all knowledge eventually derives from the events and/or objects we observe, there is nothing in the records of these events or objects that tells us what the records mean. This meaning must be constructed, and we must show how all elements interact when we construct new meanings. (p. 55)

The V diagram is divided into two major areas, the theoretical and the methodological. All of the elements are centered on the *Focus Question*, the research which provides the reason the research is conducted. Before a researcher can decide on a methodology, the elements of the left hand column, world view, philosophy, and the like, must be examined. The research question is examined in light of a given event. Only after the focus question, theoretical background, and event have been determined can the researcher begin to develop the methodology of the study.

As the research plan is developed and executed, the V diagram can serve as both overview and sequencer. The research question, research theories, and the research methodology are given in one place. By following the V from left to right, the researcher can define the theoretical underpinnings, from *World View* to *Concepts*. Only then does the researcher determine what *Records* exist or need to be created during the course of the research. Those records are then *Transformed* by encoding, and analysis. At the end of the research the researcher will make *Claims* which may then be verified or refuted by other researchers.

Subsequent chapters of this study will present the V diagram for this study and build on the basic elements of the diagram.

Impact on This Study

Although traditional and more modern approaches to education seem to be widely divergent, in fact both traditional and modern theories expect the student to take an active part in their own learning. While the traditional model may call the reflective portion of learning "exercise," the student is still applying that which was presented and integrating it into their own knowledge base.

This study of the growth of a learning community will help shed some light on various aspects of education. Specifically, I will look at the interaction of students, their interaction with each other and with the professor, and their rationale behind using or not using this collaborative tool. Especially in light of Goldberg's findings, as previously noted, I'll be very interested in learning why students do not take advantage of all the communication tools which have been made available to them.

Although this study is strictly focused on the first level computer science course, a later study of some of the same students as juniors and seniors could be very revealing concerning their growth as scholars. As they continue in their educational careers, and as they have more opportunities to interact with others, does the familiarity with the tools of communications and the learning process enable them to better utilize collaborative software? These further questions may be addressed in a followup study. We can only speculate on the answers at this time.

Chapter 3: Methodology

> In professional practice, the design process usually begins with the clarification of the client's objectives (the content) and continues through an analytical phase in which the objective is further clarified and detailed. The process progresses through a visualization phase in which the overall look and feel of the piece is determined. ... An *objective* is the desired result, or goal of any course of action. In design, our course of action is to project a message to a specified audience in the hope of obtaining a desired response from them. A clear definition and understanding of the problem at hand are essential to a successful project. (Resnick, 2003, p. 17)

In the previous chapter I introduced the Novak and Gowin (1984) V Diagram which I have chosen to organize this research project on collaborative software. What Resnick, previously quoted, said about graphic design is also true about research design. To explain the design of this research project, I shall proceed through the V diagram which was developed as one of the first steps in planning.

In this chapter you will find a short discussion of each element of the V diagram, followed by various factors which weighed in the design of the study. You will see the steps taken at the beginning, middle, and end of the semester. The chapter ends with an item-by-item analysis of the interview questions, including rationale for each question.

In brief, this study looks at the formation of a learning community in a one-semester Foundations of Computer Science class. There were 42 students who initially had signed up for the class, and 38 finished. Of the 42 who signed up for the class, 25 were male, 16 were female. None of the females dropped the class. The students were encouraged to discuss the course materials using an online tool, Manhattan Virtual

Classroom. As the semester ended, 21 students (10 male, 11 female) were interviewed. Both the interviews and the threaded discussions from the Manhattan Virtual Classroom were printed and archived.

Project V Diagram

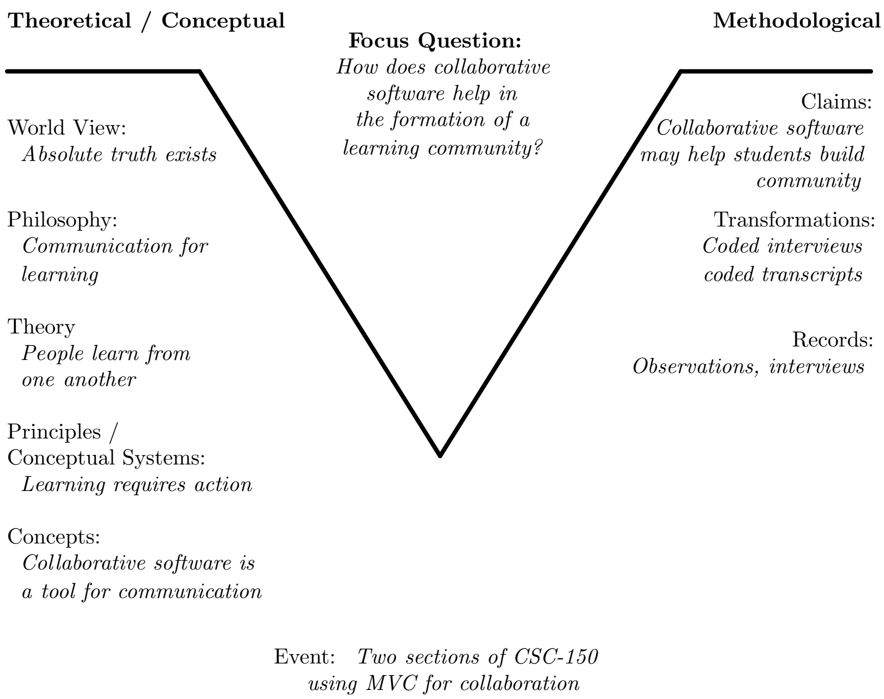

Figure 2. V Diagram - Collaborative Software and Community Building

Focus Question

The Rev. Dr. Donald Deffner, in his preaching techniques or homiletics classes, would often repeat the dictum, "You must be able to state your sermon theme in 15 words or less." In so doing, he would argue, you and your hearers would know the direction of the sermon. Every paragraph, every sentence, must relate back to the theme or it should not be part of the sermon.

I have applied the same principle when defining the overall question of this study, "How does collaborative software help in the formation of a learning community?" Everything that follows is commentary or explanation or exposition of this basic question.

Event

For every research project there is an event, a subject which is studied. In this study, the event was the class Foundations to Computer Science, a first level computer science course taught at Concordia University Wisconsin. The conclusions drawn from the data gathered, from the interviews conducted, and from an analysis of the observations made may be applied to other classes or other universities at least in a general sense. However, in this case study as in all case studies, there are unique elements which mean the final conclusions may not be universally applicable over all populations in all time and in all places.

The event which this study examines took place over a single semester. It was two sections of the same course taught by the same professor who was not the researcher. Although the course is required for several majors outside of computer science, it is not part of the general core, thus it did not attract a complete cross section of the student population. Normally, about half of each CSC-150 class are computer science majors, while the rest are language students, art students, and students of other disciplines. In the two class sections studied, however, there were no computer science majors and very few computer science minors.

The two sections of CSC-150 which were studied were taught during the day, thus most of the students are "traditional," meaning they were in their late teens or early twenties, full time students, most of whom lived on campus. Because this class is a survey of computer science, many of the students were in their first year of post-secondary education.

Theoretical / Conceptual Elements

Within these elements are several subtopics concerning world view, philosophy, theories, principles, and concepts. Each of these are filters which affect our understanding of the data which was collected and how they were analyzed. No observer is impartial, for each observer's knowledge and world view is shaped by prior events and prior learnings. Among the biases and filters which plague an observer is language. Linguistic researcher Steven Pinker (2002, p. 22) wrote:

> True, many non-Western languages lack the means to express certain abstract concepts. They may have no word for numbers higher than three, for example, or no word for goodness in general as opposed to the goodness of a particular person. But those limitations simply reflect the daily needs of those people as they live their lives, not an infirmity in their mental abilities.

In this section of the Vee diagram, Novak and Gowin describe the various filters and elements that the observer brings to a project. Thus, briefly, I shall review some of the conceptual elements which I bring to this study which impact the methodological design.

World view. I believe that there is an absolute truth. I also believe there are various views of that truth which, on the surface, may seem to be in conflict but are actually complementary. The classic story of the four blind men discovering an elephant is an appropriate illustration of this point. One, upon touching the trunk, declared the elephant was like a snake. Another, feeling a tusk, suggested the elephant was hard and smooth. The third, feeling the side of the elephant, suggested it was like a large, rough wall, while the fourth, feeling the leg, declared the elephant was like a tree. Each of the four is correct, but none was aware of the big picture.

So, too, our search for the truth. Each element of an observation adds to our storehouse of knowledge and clarifies the picture of reality. We argue for our view point, and slowly construct an understanding of the underlying structure of the true nature of the universe, or at least a portion thereof. We constantly seek to find if our model of the universe is, in fact, accurate, if the model can be used to predict future events. The more the model can be verified as correct, the more we can say we understand the truth.

Philosophy. In discussing our filters, we also need to examine our philosophy of learning. We seem to construct knowledge by integrating new information with which we are presented into our existing knowledge base. Events, information, and concepts are incorporated within a given framework within our minds. Where there is no framework, and nothing closely related upon which to build a new framework, the information presented is lost or unusable.

Because of my long association with computers, artificial intelligence, database theories, and computer networking, I tend to look at both communication and learning from an information processing viewpoint.

In the classical model of education (Sayers, 1947; Veith & Kern, 1997), the first part of the trivium is the learning of grammar, the second is logic, and the third is rhetoric. Students ought to be given many opportunities to use the words of a discipline. By using the words of a discipline, students also exercise the actions of that discipline. The more opportunities a student has to exercise newly gained vocabulary and associated skills, the more firmly entrenched will be the knowledge they have acquired.

The schema theory of cognition seems to say the same thing but in different words. F. C. Bartlett (1967, p. 201) wrote, "'Schema' refers to an active organization of past reactions, or of past experiences, which must always be supposed to be operating in any well-adapted organic response."

Another working definition of schema theory, though not by that name, is given by Jerry Fodor (1975).

> It has been a main argument of this book that if you want to know what a response to a given stimulus is going to elicit, you must find out what internal representation the organism assigns to the stimulus. Patently, the character of such assignments must in turn depend upon what kind of representational system is available for mediating the cognitive processes of the organism. (p. 163)

Theory. The hypothesis which I will examine in light of the research question: People learn from one another. Those who are part of a group may have an easier time learning than the person who is in isolation. Social cognitive theory suggests that learning from others is necessary.

Principles / Conceptual Systems. The construction of knowledge requires action. Passive learning is no learning at all. One can not simply sit and listen but must integrate that which is heard by linking it to other bits of knowledge and experience.

Concepts. Collaborative software is one of the many tools available to help students communicate, to interact with one another or with the professor. It certainly is not the only tool, especially in a traditional face-to-face type of class, but it may be an important adjunct to learning.

Methodological

Records (Data Collection). The collaborative software used in this study, Manhattan Virtual Classroom, collects discussions into threads and provides the mechanism to print these discussions. It also provides reports on individual activity, such as the date and time of access, the date and time of posting messages, and the date and time of reading messages.

In addition to the data provided by the Manhattan Virtual Classroom program, I also interviewed 21 of the 38 students who completed the class. The interviews were transcribed and placed into a database for further analysis.

Transformations (Analysis). The MVC records were printed and coded. Interview transcripts were coded using a Perl program which allowed for grouping various responses by topic. The documentation and source code for the Perl programs used to enter and encode the interview transcripts are in Appendix 4.

Claims. Claims that I make as a result of this study will be discussed in Chapters 4 and 5.

Design of the Study

This research project is a case study which focuses on the patterns of communication between students who use collaborative software. Because this is a case study, I shall report on the course, with special focus on the use of collaborative software.

Why do a case study on the use of collaborative software in a first level computer science course? First, the results reported by Goldberg (1997) and Carlson, et. al. (1996) suggest that first and second year computer science students do not take advantage of the course materials and communication opportunities made available to them. Second, Concordia University Wisconsin is starting to use WebCT and has integrated that program with the student registration and administration program, Banner. As the university seeks to use this type of software in more classes, they need evidence as to its acceptance by students. Third, the Computer Science Department has had a policy for several years, at least since the Fall, 2000, semester, of using class web sites for distributing class notes, assignments, and readings. A case study will show how this type of collaborative software can help in meeting department goals. Finally, this is a unique case, the first use of collaborative software in this introductory computer science course. This case, especially, can show how the software may be introduced and used by other instructors and students.

John Creswell (1998), a professor of educational psychology at the University of Nebraska, wrote:

> ... a *case study* is an exploration of a "bounded system" or a case (or multiple cases) over time through detailed, in-depth data collection involving multiple sources of information rich in context. This *bounded system* is bounded by time and place, and it is the *case* being studied — a program, an event, an activity, or individuals. (p. 61)

This case study is bounded by time and place: the Spring, 2004, semester at Concordia University Wisconsin. It is limited to the students of two sections of one class, CSC-150, Foundations of Computer Science.

Sharan Merriam (1988) wrote:

> The case study can be further defined by its special features. ... While the number of characteristics and the terminology may differ from source to source, a review of these and other writings suggests that the following four characteristics are essential properties of a qualitative case study: particularistic, descriptive, heuristic, and inductive. (p. 11)

Boundaries - Venue and Environment

A case study is bounded by time and place, as previously mentioned. This is what Merriam called *particularistic*. The time and place may be considered the venue of the study. Because of the technology involved in this study, the other important factor is the computer software environment, the programs used and the methods available for the student to access the collaborative software. In order to accomplish this study of the effect of collaborative software on community development, I needed to have a proper venue and environment.

University Venue. Concordia University Wisconsin (2002) is a Lutheran liberal arts institution which was established in 1881 to help train pastors and teachers for the Lutheran Church—Missouri Synod. Over the years the course offerings have expanded to include many majors beyond the training of church professionals.

Professors are expected to integrate their Christian faith into each subject area. As the Catalog (2002) says:

> Concordia University seeks to develop mature Christians in whom knowledge and understanding of the Holy Scriptures, the inspired, inerrant Word of God and the source and norm of Christian truth, are united with personal faith in Jesus as God and Savior. Concordia's spiritual resources are directed to the development of Christian faith and practice of Christian virtues so that the student can form value judgments and arrive at ethical principles required for purposeful living. (p. 7)

Department Venue. The Computer Science Department has, on the average, 50 students associated with it either as majors or minors. No computer science course is considered part of the core curriculum, but several majors, such as Business, require one or more computer science courses.

Because the department is one of the smaller ones in the school, the department chair, Dr. Gary Locklair, has tried to establish a sense of belonging within the department. Each semester he sponsors a "Computer Science Get-Together" which combines skits, food, and prizes. Each month the department also hosts "Geek Night at the Movies," which provides an opportunity for students to gather and discuss technology as shown in various films.

By developing a computer science community on campus the department is trying to accomplish several goals. First is to expand the program so to better utilize the school's investment in people and facilities. Second is to provide students who are willing to help each other in classes inside and outside the department. Third is to provide students who are trained in computer science for the many jobs on campus. The collaborative software provides another means by which these departmental goals can be met.

During the Spring 2004 semester there were 2.5 FTE faculty positions in the department. The half-time professor also teaches half time in the Philosophy Department. He is also a director of the Cranach Institute which studies Christian vocation and educational issues. With the connection between the Cranach Institute and the department, the Classical Education model is very important. The trivium permeates the course sequence, with CSC-150 being the "grammar" level of teaching.

Course Venue. Several classes were available at Concordia University Wisconsin for this study. Dr. Paul Zietlow has volunteered two sections of an advanced communications course, and Dr. Donna Reimer-Becker volunteered two sections of an introductory writing course. A researcher building on this study could easily take advantage of such offers.

The course I chose, however, was CSC-150, Foundations of Computer Science. There are several reasons this course was the best choice for this study. First, Dr. Gary Locklair, the professor who teaches this course, fully documented the course and objectives in his dissertation *Foundations of Computer Science: A Survey of Computer Science via its Grand Ideas.* Two sections of this course are offered each semester. Both sections are on the same schedule, that is to say they are taught in an identical manner by the same professor using the same course time line.

In addition, Dr. Locklair encourages students to collaborate, and has designed portions of this course to require collaboration between students. The course web site provides not only daily outlines of materials, but the presentation graphics use for each class period and suggestions from former students on how to achieve the maximum success in the course.

Finally, Dr. Locklair understands the underlying technology of the collaborative software. As the head of the Computer Science Department, he is aware of the problems of networking, data security and backup, and server administration.

Dr. Locklair (2001, p. 3) described the course in his dissertation:

> The philosophy of this text is to present concepts and fundamentals along with some practical skills which will not only serve you at the

> present, but also provide valuable insights into the foundational grand ideas of computer science.
>
> Throughout the text are various chapters marked (lab). These chapters are meant to serve as a guide through an actual "hands on" session with a computer system. The lab chapters are designed to develop problem solving and critical thinking skills as they will purposively **not** list every detail necessary for successful completion of the lab objectives. You may need assistance from an instructor or other knowledgeable person to complete the labs.

In the syllabus Dr. Locklair (2003) explains part of his course design philosophy.

> There are three important components to CSC 150: Theory, Practice and Application. Theory deals with computer science concepts. For example, the various techniques of high-level to machine language translation will be studied with an emphasis on: what it is, how it works, and why it is used. Practice deals with computer system use, that is, hands-on exposure to working systems. For example, a number of application packages will be introduced and used in laboratory sessions. Application deals with putting concepts to use in order to solve problems.

The student syllabus (Locklair, 2003) also gives additional information on both the scope of the homework assignments and the scope of student collaboration.

> Homework is assigned at various times throughout the semester. You may use any resource you desire to aid in answering the questions. You may discuss the questions with your classmates; however, plagiarism will not be tolerated. The answers to the questions should be primarily your own work - not a fellow student's. Most homework involves carrying out assignments using the computer systems. You may work together on the assignments as long as the finished project is primarily your own work. If plagiarism is detected, all parties involved may receive a 0 for that assignment (see the section on academic ethics, in the student handbook). I will not provide "procedural" help for homework assignments, but I will gladly answer questions dealing with specifications and semantics.

Dr. Locklair has been influenced greatly by the writings of Dorothy Sayers and Gene Edward Veith in regard to teaching philosophy. With CSC-150 being the grammar level of the trivium in computer science, most of the class is taught with lecture. The textbook, Schneider and Gersting (1999), is used to provide various illustrations of the grand ideas of computer science, but is not followed chapter by chapter. Many times the students do not use the text at all but rely on lecture notes and handouts.

Even at the grammar level of the trivium, students are encouraged to exercise their use of that grammar. There are ten assignments which build on the "grand ideas" (Locklair, 2001) of computer science that give an opportunity to expand into the logic and rhetoric portions of the trivium.

Sections of CSC-150, Foundations of Computer Science, have been taught at Concordia University Wisconsin by the same professor since 1987. Although in some semesters other professors will also teach sessions of this class, there have been no semester where the designer of the course has not taught the course. The course is well documented with clear goals and objectives, in part because it was the topic of the designer's dissertation (Locklair, 2001).

The class has a consistent history so data collected during the semester could be compared to data from previous semesters.

Computing Environment - Server and Software. Many different collaborative software systems were available for this study. Concordia University Wisconsin uses WebCT, and allocates server space for each class section. WebCT is well known, well documented, and readily available for use with any browser. However, WebCT is centrally administered at Concordia. It runs on its own, highly secured server, a server to which I have no administrative rights.

Thus I determined to use an Open Source collaborative software system called Manhattan Virtual Classroom. Manhattan Virtual Classroom provides most, if not all, of the functions of WebCT. In addition, it runs under a number of operating systems, thus both the server and the subsequent web pages are cross platform capable.

Manhattan Virtual Classroom ran on the Computer Science Department Linux server (pascal.cuw.edu). As the faculty administrator of this computer, I had complete access to the system and all data thereon. We used the most recent version of Manhattan Virtual Classroom, version 2.1.1, which was released in December, 2003.

Manhattan Virtual Classroom was developed by Steven Narmontas, the manager of the Educational Technology Center at Western New England College in Springfield, Massachusetts. The introduction of the Teacher's Manual (Narmontas, 2003, p. 7) says:

> Dubbed "The Manhattan Project," because it was largely developed in secret, the software enabled teachers to post files to a web site for their students to read. The earliest version of "Manhattan" also supported a few discussion groups and private messaging. Several pioneering professors made good use of the system immediately and Steve [Narmontas] continued to develop the software. Today "The Manhattan Virtual Classroom" is very much a normal part of teaching and learning at the College supporting over 300 course sections per year.

The Manhattan Virtual Classroom distribution includes extensive documentation for teachers, students, and administrators, all released as Open Source under the GNU Public License or GNU Free Documentation License. As an Open Source project, the source code is available, which means the Manhattan Virtual Classroom can be recompiled on another computer system if, in the future, the data from this project need to be examined. Manhattan Virtual Classroom is written in C and has been ported to a number of different operating systems, including several UNIX variants, Windows, and Mac OS-X. It is available from http://manhattan.sourceforge.net.

There are a number of modules available in Manhattan Virtual Classroom, but this project used only the message boards. By looking at the message boards, which are threaded discussion groups, I was looking for conversations about the course content as well as other information which would show the formation of a learning community.

If there are truly conversations in these threaded discussion groups, rather than one-sided posts with brief answers in which there are more than two participants, then a learning community may be forming. If the posts are merely simple answers to questions posed by the instructor, and if there is no interaction between the students, then a learning community is probably not forming.

Computing Environment - Student. Manhattan Virtual Classroom is accessible via Hypertext Transfer Protocols as a web page. The server, pascal.cuw.edu, has an HTTP port outside the university firewall, so the system is available to students who live off campus as well as those who live on campus. For those students who do not have a computer in their dormitory rooms, both the university and department provide computer laboratories. The university computer lab is available 24 hours a day, and the computer science lab is available from 6:00 a.m. to 10:00 p.m. each day.

Descriptive

Merriam (1988, p. 11) defined *descriptive*, "...the end product of a case study is a rich, 'thick' description of the phenomenon under study." As a case study, this research project will describe CSC-150 in detail, paying particular attention to the use of the Manhattan Virtual Classroom as part of the course. Because the course content was described in detail by Locklair (2001), and the content has not changed except in details, there is no need to discuss each class or laboratory session.

Heuristic

According to Merriam (1988, p. 13), "*heuristic* means that case studies illuminate the reader's understanding of the phenomenon under study. They can bring about the discovery of new meaning, extend the reader's experience, or confirm what is known."

This particular study is enlightening because it concerns the first use of collaborative software in the CSC-150 course. Although Dr. Locklair had previously used Manhattan Virtual Classroom for several masters level classes, the Spring 2004 semester was the first time he used it for an undergraduate class.

Inductive

"For the most part, case studies rely on inductive reasoning," according to Merriam (1988, p. 13). This leads to the discovery of new relationships, new concepts, and new understandings as the case study unfolds.

Procedure

In the Spring 2004 semester, 42 students initially signed up for CSC-150. There were 21 students per section, giving an over-subscription of one student per section. Of the 42, 38 finished the course, and of those 21 consented to be interviewed.

Beginning of the Semester.

At the beginning of the semester, Dr. Locklair made the Manhattan Virtual Classroom software available to the students via a link on the course home page and the Computer Science Department home page. The students of each section received a letter, shown in Appendix 1, concerning the study.

During the Semester.

During the semester I monitored the use of the collaborative software to determine usage patterns, topic strings, and the like. Specifically I took the role of system administrator so to ensure that the software was available and properly working.

I did not, as system administrator, take part in discussions or in any way influence the students of these course sections with the exception of answering one question answered for Dr. Locklair when he was out of town. I also made several announcements about interview scheduling and replied to student questions about the interviews (15 posts total).

End of the Semester.

This study was approved by both the Kansas State University and Concordia University Wisconsin Institutional Review Boards which determined the research was conducted within the guidelines of both institutions. These guidelines are both procedural and ethical in nature.

At the end of the semester I archived all the discussions for later analysis and interviewed many students from each section. The interview process began week ten in the semester. By the tenth week of the semester, students knew how they were doing in the class, had definite opinions about the use of the collaborative software, and yet were not hampered by the pressures of the end of the semester.

Both the Manhattan Virtual Classroom (software and discussions) and interviews were archived on a CD-ROM which I will retain in my files. Because the Manhattan Virtual Classroom is in a self-contained directory on the server, taking a snapshot of the directory for archival purposes was not difficult. Interviews were done using both a digital recorders and a tape which allows the interviews to be preserved on the CD-ROM.

Because student names are available on this CD-ROM, I will maintain student confidentiality. The Family Educational Rights and Privacy Act (2002) says that schools can not release information from a student's educational record unless they have written permission. Directory information, such as names and addresses, may be released without permission unless the student specifically requests the directory not be released.

So to follow the dictates of this act in both the spirit and the letter of the law, the archive of the Manhattan Virtual Classroom sessions and other data collected which may identify students will not be released to other researchers until such identifying information is removed.

During the interview and analysis process, I established a web site which was used to show the progress of this study. Everyone involved, the students, Dr. Locklair, and the dissertation committee, has the right to know how the research is proceeding and what the preliminary findings were.

Non-Participation

Students had the right and ability not to participate in this study. They refrained from participating in two ways. First, some simply decided not to use the software. Second, some who used the software elected not to be interviewed. Only the students who were interviewed signed the consent forms.

I invited all of the students in both sections of CSC-150 to be part of the interview process. All who accepted the invitation were interviewed. My rationale for having a large number of students participate was to obtain a good cross-section of the student body.

Interview Questions and Rationale

The interview questions were divided into three sections, questions about the student being interviewed, questions about the class in general, and questions specific to Manhattan Virtual Classroom. These questions were open-ended. Students were encouraged to use the questions as a springboard for discussion rather than simply providing a one or two word answer.

Questions About the Student

The first questions established the student's reason for taking the class, initial expectations, and the realization of those expectations. Later questions gave the student the opportunity to comment on procedures, peer interaction, and community involvement.

1. *What is your major? What is your minor?* Because this class is required for several majors, not all students are going to continue in computer science. By asking this question I determined the course audience.

Questions About the Class

These questions are specific to the class as regards homework and study groups.

1. *What did you expect to learn in this class?* Some students believe that the study of computer science is the study of programming, others understand the study to be the use of computers in problem solving. Later qualitative analysis of student perceptions of the class took into account student expectations as opposed to the

reality of the class.

2. *What did you learn in this class?* With such an open-ended question, I hoped to determine that the students met the stated objectives of the class, and see if the collaborative software helped them to integrate and use the information presented in the class.

3. *CSC-150 has quite a bit of homework. What assignments did you find easy? What assignments did you find difficult?* Student success in CSC-150 depends on completing homework so to demonstrate mastery of various concepts. Much of the homework involves the use of computer programs for problem solving and information gathering. By asking these homework related questions, I was able to determine the student's perspective of the difficulty of the course.

4. *Where did you go, or what resources did you use, to help complete the homework assignments?* With this question we began to explore the idea of community building.

5. *Did you discuss homework with other students in your course section or in the other course section? If so, what were your reasons for so doing?* Again, this question deals with community building and collaboration.

6. *Did you discuss homework with other computer science students not in CSC-150? If so, what category of student did you seek (i.e., senior CS major, sophomore CS major, CS minor)?* There is a larger community of computer science students who can influence the CSC-150 students. No class section is taught in isolation, but is part of the department. We encourage students to interact with other students at all levels of the computer science program, and provide several social

events to help this interaction take place.

7. *Did you discuss homework with other people, such as lab assistants, others who are not in the CS program who had taken the course, etc.?* The students are part of the University community, which may or may not be a good influence. I was looking for the student's willingness to ask for help in general, and their connection to the University community.

8. *Did you form study groups for any or all of the material or tests?* This was another way to discover the details of community formation within a course section.

9. *Did you take advantage of the study materials to be found on the course web site? If so, what was most helpful and why?* Part of the ongoing nature of a community was the collecting and dissemination of shared experiences. The CSC-150 web site has suggestions from former students to help current students succeed.

10. *Did you use any electronic means (electronic mail, chat rooms, instant messaging services) to communicate with any of your peers or others concerning this course? If so, what and to what extent? Was this helpful?* Because this is a computer science class, we would hope that students would take advantage of computer networks, electronic mail, and other online communications tools. Because members of both sections used these tools, we can gauge the perceived importance of group formation to the students.

Manhattan Virtual Classroom

The following questions are specific to the Manhattan Virtual Classroom. Students may have decided not to participate in online conversations because of the difficulty of using the MVC tools or a general mistrust of computers.

1. *How would you rate the usability of Manhattan Virtual Classroom (easy to use, difficult to use, neutral) and why?* If the online collaboration tool is difficult to use or is prone to crash, the students will not use it. Therefore, this question is important so to determine if the tool, itself, has affected the outcome of the study.
2. *What features of Manhattan Virtual Classroom did you use?* Manhattan Virtual Classroom has a number of features beyond threaded discussion bulletin boards. Online chat and electronic mail are available, as is an anonymous discussion area.
3. *How often did you access Manhattan Virtual Classroom? For what purposes?* The more frequently the Manhattan Virtual Classroom is accessed, the more effect it may have on student grades and satisfaction. Students could access the Manhattan Virtual Classroom for the purpose of posting a question, gathering replies, or simply lurking to see what others are saying. These data were also available in the Manhattan Virtual Classroom archives.
4. *Did other students answer questions or respond to your comments?* No collaboration tool is worthwhile if questions or comments are left unanswered. The student's perception of the value of the tool can be determined from this question.
5. *Did you answer questions or respond to comments? What were your criteria for*

responding? When tied with the preceding question, I determined the student's commitment to helping others, or to forming a community.

6. *How much time did you spend, typically, on online collaboration?* By this question we determined the additional time required to use such a collaborative tool.
7. *What is your opinion concerning Manhattan Virtual Classroom, its usefulness to you as a student?* If students don't consider a tool to be useful, they simply will not use it. Although the teacher or professor believes the tool to be an important resource, students ultimately will apply themselves to that which the perceive will help them succeed.
8. Please finish the following sentence:*Manhattan Virtual Classroom should / should not be used in CSC-150 because:* Based on these sentence completions I can determine if such a tool is perceived as adding value for enhancing student outcomes. If the tool meets student needs, it should be offered for other classes in the department.

Data Sources

The following table shows the sources of the data collected and the purpose of each resource.

Data Sources		
Item	Source	Purpose
CSC-150 class list	CUW Registrar	List of participants, majors, academic year
CSC-150 course materials	Dr. Gary Locklair	Review the materials taught at a given time, review the "grand ideas" of computer science
Dissertation about CSC-150	Dr. Gary Locklair	Obtain insight into the class, why certain topics were taught and the order in which they were taught
CSC-150 grades	CUW Registrar	Historical records, compare present class with past classes
Manhattan Virtual Classroom threaded discussions	Manhattan Virtual Classroom	Analyze content of discussions
Manhattan Virtual Classroom statistics	Manhattan Virtual Classroom	Compare student reported use with actual use
Student interviews	Students	Determine student perceptions of CSC-150 and Manhattan Virtual Classroom

TABLE 2. Data Sources for *Collaborative Software and Community Building*

Item Analysis

At the end of the semester I had the interview transcripts and the message archives to analyze. The interview analysis was done using several programs to encode and annotate the transcripts. The message archives, due to the nature of the data capture, were analyzed by hand.

Interviews

The transcripts of the interviews had been typed and loaded into a MySQL database. I wrote and used the ENCODE.PL program (see Appendix 4 for the source code) to annotate the transcripts. As I annotated the interviews, I selected passages based on the following 17 codes. These codes were developed based on the interview questions.

Code id	Code description	Question
1	Initial expectations of CSC-150	Class 1
2	Topics learned in CSC-150	Class 2
3	Difficult homework	Class 3
4	Outside resources	Class 4, 5, 6, 7, 8, 9, 10
5	Study materials	Class 9
6	Frustrations	Class 3
7	Satisfactions	Class 3
8	MVC features	MVC 2
9	Posting guidelines / reasons MVC 4, 5	
10	Time using MVC at each use	MVC 6
11	Nature of MVC - useful or not	MVC 7, 8
12	Continued use of MVC for CSC-150	MVC 7, 8
13	MVC compared to WebCT / Blackboard, etc.	MVC 7
14	Use of instant messenger	Class 10
15	Understands importance of posting questions so others have the benefit of their knowledge	Class 10; MVC 4, 5
16	Community in CS	Class 5, 6, 7
17	Student's Major or Minor	Student 1

TABLE 3. Interview Transcripts Coding

Each transcript was reviewed a minimum of four times. The first review was for basic form and content, correcting transcriptionist errors when students used terms which were unknown to the transcriber. The second review was after the transcript was uploaded into the database so to check the accuracy of the upload process. The third review was to encode comments and to add annotations to various lines of the transcript. Some interview lines were not coded, especially the comments about the weather, sports, or the Spring play in which one of the interviewees had participated.

Once the third review was finished, I printed each line of the interview to which a code was attached, thus being able to easily see comments from students about a given topic. I then reviewed those comments within the context of each interview looking for additional, relevant information.

Following the printout of the coding report, I read each interview at least one more time to ensure that I did not miss an important piece of information.

MVC Archives

As previously mentioned, the Manhattan Virtual Classroom archives were of such a format that analysis was easiest if done by hand. Although I had complete access to the archives as the system administrator, I determined that figuring out the compression scheme and internal database scheme for the messages would not be necessary.

For the analysis I printed the message threads and MVC logs. Message threads were classified as informational, procedural, short answer questions, virtual office, and student requests for help. I noted who had started the thread and how many replies were given on the thread.

The Manhattan Virtual Classroom program collects a number of statistics about each participant in a given class. Aside from the name and identification number for a participant, MVC notes the number of unread messages, the number of times someone has accessed the MVC system, the date and time of each access, and when messages arrive or are read. With this information I can see who accessed MVC and how many messages they posted and read.

Other Resources

As previously noted, the Registrar's Office and Dr. Locklair provided information about each class section and previous class sections. The grade information for previous classes was processed to obtain a grade distribution which I used to compare the Spring 2004 semester with previous semesters.

Although not truly a part of the study, I nonetheless had a chance to observe some of the CSC-150 students in the Computer Science Department labs and at various social events. These observations were valuable in demonstrating the formation, or lack of formation, of a relationship between the CSC-150 students and the rest of the department.

Peer Review

During the course of the semester and subsequent analysis, I discussed my findings with Dr. Locklair and several other professors. In addition, I placed each chapter of the research report on a web site which is available to students so to receive their comments. Several students (not members of the two class sections which were studied) did read the various chapters and agreed that I had captured the essence of both CSC-150 and the Computer Science department.

During the Spring 2004 and Fall 2004 semesters, I presented the progress of my research to a group of interested faculty members at a monthly"dissertation support group" that was sponsored by the Faculty Development Committee. They commented on the ongoing analysis and gave insights as to the mechanics of the research.

The final review of the dissertation was conducted by Dr. Locklair and several other faculty members, as well as students and interested staff members. They agreed with the results of the research and the conclusions which were drawn.

Chapter 4: Observations

> When I say that the question "finite or infinite?" is a red herring, I don't mean simply that philosophers and theologians have often been arguing about an unimportant issue. I also mean that physicists and other scientists fail to realize this. For example, take the literature of chaos theory: Hundreds of papers have been written about the behavior of solutions to unstable recurrences, by people who assume that *real* numbers are *real*. (Knuth, 2001, p. 174)

In this chapter I begin with an analysis of the student population for CSC-150. Following the discussion of the student population, I will note the resources available to the CSC-150 students, provided both by the professor and by former CSC-150 students. I then will quantify the online discussions as to topic, posting party, and the like. Finally, I will discuss the interview process.

CSC 150 Students: Demographics and Grades

Number

In the Spring 2004 semester, 38 students finished CSC-150, Foundations of Computer Science at Concordia University Wisconsin. There were two sections of this class, both taught by Dr. Gary Locklair on Monday, Wednesday, and Friday mornings.

Grades

The students earned the grades shown in Table 3:

Twenty-five of the 38 students who completed the course received a B- or better as their final grade. Only three students students received less than a C-, and no student failed the class. Thus, as measured with grades, the class seems to have been successful

CSC-150 Grades	
Grade	Number
A	7
A-	5
B+	6
B	5
B-	2
C+	5
C	3
C-	2
D+	0
D	2
D-	1
F	0

TABLE 4. Spring 2004 CSC-150 Grades

in that students gave evidence of learning the presented materials.

CSC-150 had been taught in 43 different sections since the Fall 1988 semester with 1942 students finishing the class. When the grades from the Spring 2004 semester are compared to the overall grades historically, a higher percentage of students (31.57% vs. 23.11%) received an A or A-, but the distribution of grades is consistent with the historical grades. The grade percentages are shown in Table 4.

Grade	Historic	Historic Percent	Spring 2004	Spring 2004 Percent
A	294	15.13	7	18.42
A-	155	7.98	5	13.15
B+	232	11.94	6	15.78
B	368	18.94	5	13.15
B-	187	9.62	2	5.26
C+	144	7.41	5	13.15
C	224	11.53	3	7.89
C-	97	4.99	2	5.26
D+	45	2.31	0	0
D	72	3.70	2	5.26
D-	38	1.95	1	2.63
F	86	4.42	0	0
Total	1942		38	

TABLE 5. Historic Grades vs. Spring 2004 Grades

The box-plot graph in Figure 3 shows the range of grades for each class session. Each session is represented by a notched box. Minimum and maximum values are shown by the dotted "whiskers" extending from the box, while the 25% and 75% values are represented by the top and bottom of the box. The median is noted with the notch on the box.

The Spring 2004 grades are not inconsistent with grades from previous classes.

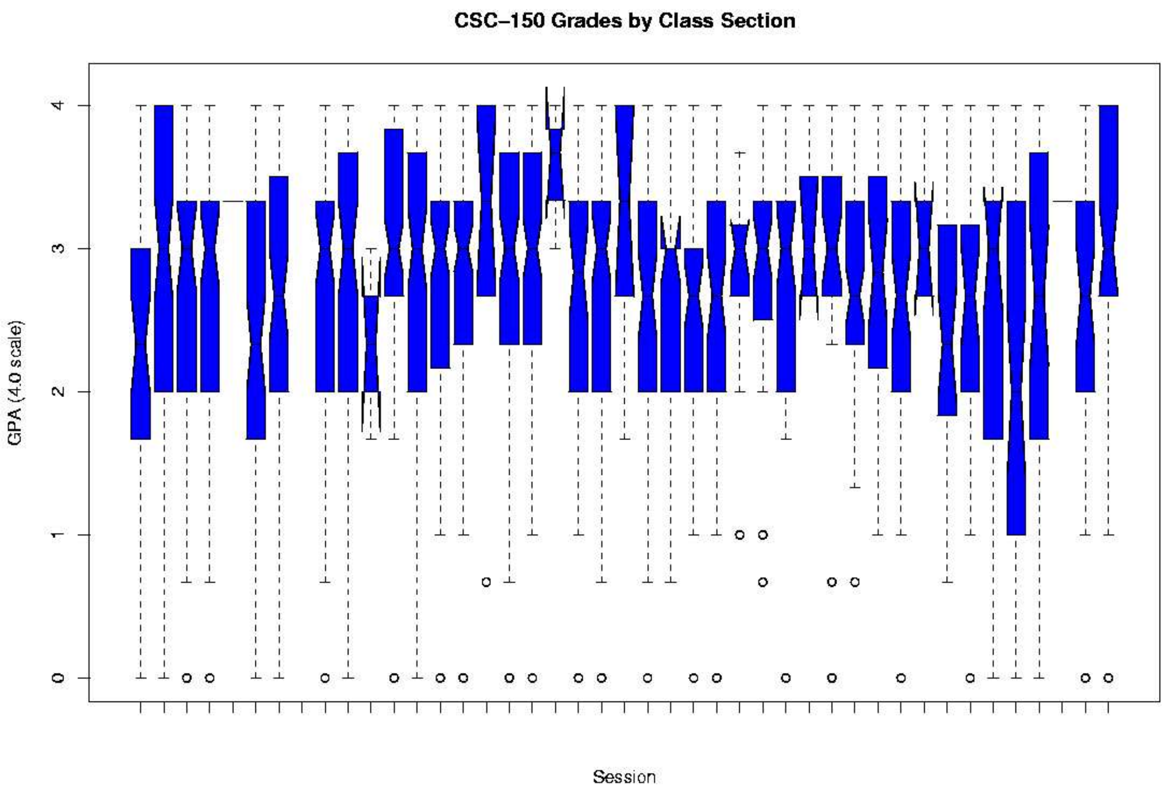

Figure 3. CSC-150 Student Grades by Class Section

The course grades, according to the syllabus (Locklair, 2003), are determined as follows:

There are 500 points possible in CSC 150, distributed as:

3 unit examinations	200 points (100 points each; drop lowest)
Homework	150 points total
Term Project	50 points
Final Exam	100 points

I do not put letter grades on exams or homework - only point scores. To determine your grade at any time in the semester, divide your accumulated points by the total points possible to date. The resulting percentage score can be used to determine a letter grade.

The following will serve as a tentative grading guide:

90	100 percent	A range
79	89 percent	B range
68	78 percent	C range
59	67 percent	D range
	less than 59 percent	F (let's try it again next semester)

CUW grades with + and - letter grades. Scores in the top two percentage points of any range will receive a plus while scores in the bottom two points of any range will receive a minus.

Majors

Twenty-one students were interviewed for this project. Of those 21, no one was a computer science major, and only two were computer science minors. Twelve were in some business area, including accounting, management, and marketing. Five were in education, either secondary or elementary. There were individuals who were theology, communications, graphic design, and interior design majors. Only one was undecided. According to the Registrar's office, there were no computer science majors in CSC-150 during this semester. Table 5 shows the distribution of majors and minors for the Spring 2004 sections of CSC-150.

Dr. Locklair (July 19, 2004, personal correspondence) suggested most of the computer science majors take CSC-150 in the Fall semester because of the sequence of

Academic Area	Majors	Minors
Accounting	2	
Broad Field Science - Secondary Education	1	
Business	6	2
Communications	3	
Computer Science		2
Education	3	
Education K-12	1	
Finance	1	
Graphic Design	1	
History - Secondary Education	1	
Interior Design	3	
Lay Ministry	1	
Management	3	
Mathematics		1
Marketing	1	1
Parish Teaching		
Secondary Education	4	
Spanish		1
Theology	2	1
Psychology	2	
Public Relations		1
Social Work	1	
Undeclared	1	
Youth Ministry		1

TABLE 6. Spring CSC-150 Majors / Minors

classes within the major. "I don't remember if there were any CS majors in CSC150 during the Spring. I usually have several each Fall, but not too many in the Spring (because of scope/sequence)." This particular semester's numbers give credence to this observation.

CSC 150 Web Resources

The Computer Science Department at Concordia University Wisconsin has a number of web servers for courses, student projects, and other tasks. For at least the last five years all course materials have been available on the Web. Some professors prefer to place visuals on the web after the class section, but provide worksheets before class. Others place the visuals online and encourage students to print them so that student notes can be placed with a given presentation graphics slide.

The department first used Manhattan Virtual Classroom (MVC) in the Spring 2003 semester for several classes. Now most of the classes have an MVC component. There are still several unexplored modules in MVC, but all the classes which use MVC take advantage of the discussion areas.

Professor Provided Web Resources

Before each day's class, Dr. Locklair would provide a study guide for the day's discussion. These would be available in both HTML and Microsoft .DOC format. Instructions for using the web resources were provided to the students and are reproduced in Appendix 2. A sample study guide is provided in Appendix 3.

One student said of these study guides, "I used the study guide or outline that was provided which I thought helped a lot because during class you weren't trying to write everything down and could pay more attention to what he was saying and he had the blank fill in the words to follow along."

Aside from the daily study guides, each day's presentation graphics were available after class. Test study guides, homework assignments, lab exercises, and streaming audio of each class session were also available on the web. The students appreciated all of the materials. One student, commenting on how Dr. Locklair provides a lot of material on the web site, said:

> [The material is] very useful, obviously for the outlines and day to day note taking... The test study sheets was [sic] also very important because it was a nice summarization of definitions and things that he would more use than our own definitions and it kind of gave you a

> technical display of what was to be expected as far as what you should be thinking of as far as answering those questions. ...but between the outlines and tests and potential test material that was the most beneficial.

Student Authored Web Resources

Over the years a number of students have given Dr. Locklair their study materials for the exams. These are available on the course web site. Current students do not place materials on the course web site, however. They are free to post comments and questions on the Manhattan Virtual Classroom page.

MVC Postings

Manhattan Virtual Classroom provides several different discussion areas. Dr. Locklair selected three types of discussion area for CSC-150, but only one was actually used. Students were provided with a general discussion area, the area that was actually used, a chat area, and a "Team Discussion" area. There were two teams assigned, one for each class section, but no discussions took place in the team area.

Palloff and Pratt (1999) wrote:

> If a synchronous meeting or seminar is being conducted, the group should be small enough to allow for full participation and to prevent information overload. Groups that are too large can be overwhelming for the instructor and the participants; five to ten is an ideal number. Asynchronous groups, however, can be much larger. As many as twenty or more participants can have a successful experience in an asynchronous setting. However, the success of a large group depends on the skill of the instructor as facilitator, his or her knowledge of the electronic medium, the content being discussed and explored, and the means by which that exploration occurs. (p. 55)

With 38 students completing the course, and all of the students posting in one large discussion area, we could expect that many would become overwhelmed and stop reading and posting. Later we shall examine some of the numbers and see that this did happen.

Although Manhattan Virtual Classroom was used for several classes in the Computer Science Department in previous semesters, this is the first semester it has been used for CSC-150. There were approximately 433 messages posted in the Classroom. Only one student read all the messages, while most of the students read fewer than half. Seventeen students accessed MVC the equivalent of once a week or less. Only five students posted the equivalent of one message per week.

The Table 6 gives the number of unread messages, the number of messages sent, the number of times the student accessed Manhattan Virtual Classroom, and their final grade for the class.

Table 7 is the same information but sorted by grade rather than unread messages.

When the average of unread messages, posted messages, and logins are computed, the numbers are quite interesting. The average student logged into the system slightly over 19 times during the semester. He or she sent 6 messages, read approximately 100, and left 330 unread.

In Table 8, the grades are displayed with the number of messages sent by all the students receiving that grade, and the average number of messages sent by each student receiving that grade. Within a given grade band (A, B, C), the higher the grade, the higher the average number of messages posted per student. We can only speculate as to the reason for this distribution of grades vs. messages. A larger sample may show different results.

Messages, Logins, and Grade			
Unread	Posted	Logins	Grade
0	15	86	A
21	14	25	A
22	17	39	A
89	6	38	B+
231	4	29	A-
247	13	31	A-
273	2	15	B-
283	3	11	B
286	13	50	A-
291	14	63	B
300	8	16	A
301	6	34	B+
307	13	13	C
311	10	19	C+
319	13	25	A
347	10	25	A-
357	1	11	B+
358	5	11	B
359	5	2	A
364	15	20	B+
369	12	13	C+
378	9	19	C+
382	9	41	B+
383	0	14	C-
399	2	29	C
401	0	6	A-
410	4	10	B
415	4	13	D-
415	2	7	B
416	4	7	A
417	1	11	D
418	2	10	B-
420	3	4	C-
425	1	3	C+
430	1	2	C+
432	0	2	B+
433	0	1	D
433	0	1	C
433	0	0	C-

TABLE 7. Message Activity vs. Final Grade

Grades, Logins, and Messages			
Grade	Unread	Logins	Posted
A	0	86	15
A	21	25	14
A	300	16	8
A	319	25	13
A	359	2	5
A	416	7	4
A	22	39	17
A-	286	50	13
A-	231	29	4
A-	347	25	10
A-	247	31	13
A-	401	6	0
B+	89	38	6
B+	382	41	9
B+	301	34	6
B+	432	2	0
B+	357	11	1
B+	364	20	15
B	291	63	14
B	415	7	2
B	410	10	4
B	358	11	5
B	283	11	3
B-	418	10	2
B-	273	15	2
C+	369	13	12
C+	430	2	1
C+	311	19	10
C+	425	3	1
C+	378	19	9
C	399	29	2
C	307	13	13
C	433	1	0
C-	383	14	0
C-	420	4	3
C-	433	0	0
D	417	11	1
D	433	1	0
D-	415	13	4

TABLE 8. Grade vs. Message Activity

Topics Initiated per Week

Table 9 shows the number of topics initiated during a given week of the semester, who initiated the topic (professor (P) or student (S)), the total number of posts per topic by professor or student, and type of post. Topics have been divided into informational, procedural, short answer questions, virtual office, and student requests for

Grade	Students	Messages	Average
A	7	76	10.8571
A-	5	40	8.0000
B+	6	37	6.1667
B	5	28	5.6000
B-	2	4	2.0000
C+	5	33	6.6000
C	3	15	5.0000
C-	2	3	1.0000
D	2	1	0.5000
D-	1	4	4.0000

TABLE 9. Average Logins by Grade Earned

help.

In addition to the 42 topics initiated by either the professor or a student, I initiated three topics with a total of 15 posts by me and 9 posts by students. These posts were concerning the procedure for scheduling interviews.

Informational (I): An informational post does not necessarily require a reply. It includes a welcome to the Manhattan Virtual Classroom, and various other announcements.

Procedural (R): A procedural message deals with the logistics of the class, such as how to contact the professor.

Short Answer Questions (Q): These are similar to test questions where the students responded to the professor but not to each other.

Virtual Office (O): Several times students used the Manhattan Virtual Classroom as a simple way to contact the professor to ask questions.

Student Requests for Help (H): These student requests were for suggestions on how to study for an exam or for help in solving computer system problems.

Week Number	Dates	Initiated		Posts		Type				
		P	S	P	S	I	R	Q	O	H
1	01/25-01/31	1		1	1	2				
2	02/01-02/07	2		13	35			2		
3	02/08-02/14	4		40	38	1		3		
4	02/15-02/21									
5	02/22-02/28	2	1	9	25	1	1		1	
6	02/29-03/06		1	1	1				1	
7	03/07-03/13	4	2	24	21		1	3		1
Break	03/14-03/20									
8	03/21-03/27	1	2	19	19		1	1	1	
9	03/28-04/03	4	2	14	21		2	3		2
10	04/04-04/10	1		7	3			1		
11	04/11-04/17	1	3	15	19			1		3
12	04/18-04/24	3	4	15	29	2	1			4
13	04/25-05/01a	2		13	16			2		
14	05/02-05/08		1	1	2				1	
Finals	05/09-05/15	1		1		1				

TABLE 10. Topics Initiated by Week

Professor Initiated

Of the 45 topics within the classroom discussion board, Dr. Locklair initiated 26. These mostly were questions concerning the Grand Ideas of computer science, such as possible algorithms for face recognition. In addition, several were informational, such as posting a notice that final grades were online. Some were seeking information about student attitudes and procedures on test taking, the grading scale, and class participation.

For an example, on February 2, 2004, one week into the Spring semester, Dr. Locklair posted: "Is knowing how to effectively use a computer system to solve problems as important as knowing how to read and write? Post your thoughts and comments."

There were 24 messages in that particular discussion thread. Students responded to Dr. Locklair's question, but did not respond to each other. Some observations by students showed they were trying to relate to their own experience. "McDonald employees today already have to utilize computer skills whether it is to take orders on a computer based system, cook food on a grill that is preset, or (if you are a manager) enter sales data and other information at the end of the day. ... It almost seems like we've already entered the era where computer skills are necessary to some extent." "I personally don't think so, however that could be only because I am not really into computers."

By the end of the semester the posts had changed, as had the answers and participation. Here the students did respond to each other, and there was a dialog between Dr. Locklair and two students. There were only six postings on this thread, three from Dr. Locklair.

> In class, we discussed how the switches inside the CPU allow the computer to process. I know this is a difficult concept, and I'd like your help.
>
> If you would post an explanation of how this happens in your own words, that would help me learn how to better explain this concept.
>
> Or, if you can post what part of the explanation was ambiguous (unclear), I can learn from that also.
>
> Thanks.

One student replied:

> This is how I view switches inside the computer. I think of them as a bunch of on or off switches. When an instruction is read that tells one of the switches to turn on, the information that is created by the turning on the switch outputs information that is relevant to that particular switch. Obviously this contributes to billions of possible combinations.

Another student wrote:

> This [the above post which was reproduced in the body of the reply] is similar to how I understand it along with how [student initials] explained it being controlled by electricity to form the different combinations based on instructions.

There were 4 discussion threads that were truly conversational, the earliest starting on April 15. Three were asking specific problem solving questions, such as “What causes a System 32 crash?” One was a quick review and 3 message conversation about the movie *2001: A Space Odyssey.* A total of 21 messages were posted in these 4 threads.

Although the students typically did not directly respond to each other, and the tenor of the question almost suggested they didn’t need to, clearly they were reading each others’ posts. One post built on another, thus giving all the students positive feedback.

Student Initiated

Students initiated 17 of the conversation threads during the semester. Eight of these threads were about the logistics of the course, essentially an asynchronous virtual office visit. These threads typically had two elements, the question and the simple answer.

One virtual office visit led to a conversation about the C++ programming assignment. According to the interviews, students perceived this assignment as being the most difficult in the course. All of the students interviewed had difficulty with this assignment.

Because the interviews were open-ended, some of the questions to which the students responded were of an ad-hoc nature. One such question was, “What did you find difficult about the C++ assignment?” The problem was not in the C++ language, but in a lack of familiarity with programming. As one student said, “I didn’t know where to put different symbols and stuff.” Another admitted, “Because I have never really understood programming.” A third remarked, “Probably the C++ assignments because that was foreign to me. It was like another language."

One thread begun by a student was for technical support on an operating system issue. This post led to several responses about pop-up advertisements, viruses, and other problems which exist because of data communications.

Only during the last six weeks of the semester did any student initiated post elicit a conversation or response from someone other than a professor. One of the threads that did elicit comments was begun by a student who asked,

> I study for the test from reading the book and looking over the notes. I study really hard and I still do not get a good grade. Is there any advice that people could give to me on how to do better on the tests or how to study better for the tests?

She received several replies:

> I've found that when trying to learn and study so much new information, it is much easier to review and study several times throughout the unit instead of shortly before the exam. I take a few minutes sometime during the day after class and review the information so that it is fresh in my mind. By doing so after each class period it is much easier to retain the information.
>
> Here a few things that seem to work well for me in this class. I print off the notes for the day but don't fill them out in class; I use them to follow along while I take my own notes in detail. Then I go back and fill in the handout notes from my own. I find this to be a good way to review. When I start to really focus on an upcoming test, I read through my own notes and then (before moving on to the next day's notes) read through the handout notes for that day. This helps to really reinforce it for me as I am reading the notes twice. When I read things in the text book that I find confusing I try to find a corresponding concept in our notes (lab notes have helped with this); I try to focus on how we used a concept and let the text help reinforce that. I also find the handout notes in lab to be helpful for other concepts.

Interviews Process

During the last three weeks of class, 21 students agreed to be interviewed concerning the use of Manhattan Virtual Classroom and their perceived success in CSC-150. These interviews were recorded both on tape and captured as MP3 files, thus ensuring that no single point of failure would cause an interview to be lost. The recordings were transcribed into a text file and loaded into a database for coding and analysis.

Main Research Question — Collaborative Software and Community

As we look at the formation of a learning community, we do well to review both the definition of a community or group and the stages of a group's existence. Patricia Wallace (1999, p.57) defines a group as "a collection of two or more people who are interacting with and influencing one another." To this is added the concept of interaction.

About the stages of group and community development, Palloff and Pratt (1999) wrote:

> Writers in the areas of group and organizational behavior have referred to these stages as forming, norming, storming, performing, and adjourning. First people come together around a common purpose. This is the forming stage. Then they reach out to one another to figure out how to work toward common goals, developing norms of behavior in the process. It is not uncommon as this occurs for conflict to begin as members grapple with the negotiation of individual differences versus the collective purpose or objective. However, in order to achieve group cohesion and to perform tasks together, the group needs to walk through that conflict. If attempts are made to avoid the conflict, the group may disintegrate or simply go through the motions, never really achieving intimacy. Just as in face-to-face groups, the conflict phase is an essential element that the group must work through in order to move on to the performing stage. (p. 26)

As the analysis of the postings in the Manhattan Virtual Classroom and the student interviews continued, several questions emerged. Is the class actually a group in which the members influence each other? Does this group display the stages of group formation and life?

If community implies communication, and if communication leads to a change of attitude, ideas, or ability, then the quality of communication within the class is important. Did the students interact in the classroom? Dr. Locklair (July 19, 2004) wrote in a private electronic mail message:

> I would say that each section of CSC150 was similar when it came to in-class discussion. As you know, for a 100-level class we can't really do much discussion as we're setting the foundational (grammar) knowledge of the discipline. However, students do ask and answer questions, etc. I would say that the sections were about average when it came to class participation. Over the course of the semester, everyone either asked or answered a question in class.

If the students did not actively participate in lengthy classroom discussions, they certainly may not feel the need to participate in an online discussion. Class participation has much to do with the teacher as well as the student. If the teacher does not encourage participation, the students will not participate. Yet, even if the teacher does ask for participation, students may not wish to expose their lack of knowledge or understanding to their peers.

Several students, when discussing why they did not post in the Manhattan Virtual Classroom, said, essentially, they did not wish to appear as if they did not know an answer. Although the interviewer did not specifically ask about classroom participation, the same motivation is probably at work.

> If I felt that I had a good understanding of what the question was pretty much my thing. If I wasn't exactly sure I wouldn't post it.

> If it was just like an opinion or if I knew what I was talking about otherwise I didn't want to say anything because I didn't want to be wrong.
>
> Probably if I knew the answer or something.

Q1: Did the use of collaborative software help in the formation of a learning community?

A prior question needs to be asked under the circumstances, "Did a learning community actually form?" Based on the interviews of students and the posts in the Manhattan Virtual Classroom, there was little interaction between students. Most of the posts in MVC were direct answers to professor-initiated questions, with very few posts being directed from student to student.

In addition, the students who were interviewed suggested they did not, for the most part, collaborate in completing homework assignments or discussing the class materials. One student commented, "No, I don't really learn from other students." Another student remarked,

> Well, I offered one of the students who posted online that they needed help studying for an exam, and I talked to him outside of class one time, it was the night before the exam, and said that if you need any help just holler because I am an A student and he obviously needed help and I would help. So I was offering, but he never contacted me.

Yet a few students did work together. A student said she studied with three or four people occasionally. "Yes, I worked with a girl in the same 150 section, not on every assignment, but when they started getting more difficult we decided to work together so we could pool our resources."

Some of the students mentioned receiving help from other computer science students who were farther along in the program. "There was an older guy, I don't know his name, but like he is a computer science major and he was telling me, and I was doing an assignment and was having trouble in the lab, and he was in there and he just helped me through it a little bit." "Actually I went to [a junior computer science major] on that one. He is the one that helped me with that." "I talked to<another computer science junior> every now and then just when I was like doing studying and understanding a term project."

Of no surprise, several students went to roommates who had previously taken the course. "My roommate helped me a little bit with stuff she did in class last semester. She did help me with the programming a little bit."

Was there enough interaction between the members of the CSC-150 class sections to truly suggest the formation of a learning community? Probably not. The students in CSC-150 did not, for the most part, take part in any of the other computer science activities which are designed to build a community. With only two computer science minors in the class sections, it is not difficult to see that, aside from passing the course, there was little motivation to form relationships with other class members. There is also no evidence that students with other majors worked together or formed any type of community in this class.

Q2: What were the stages of community formation?

As earlier stated, a group or community exists when people come together to interact and to solve a particular problem. The group known as the CSC-150 student body came together for the purpose of learning about the foundations of computer science. Most of the students were part of this group because their major or minor required they take this course. As one student said, "I had to take it, I really didn't want to take it, and I really didn't think I would like it."

Because of the purpose of the group, an introductory level class, the professor dominated the group. His agenda became their agenda, his standards became their standards. Students were encouraged to ask questions and to participate in discussions, but these discussions were typically instigated by the professor.

The stages of the group life cycle, as previously mentioned, may be called "forming, storming, norming, performing, and adjourning." Did the messages in MVC and the interviews show these stages of the group development?

Forming of this group happened several months before the first class session as the students registered. Students were aware of the existence of this class and were encouraged to enroll in this class. Because of a history of student involvement in this class, the university and the professor who taught the class agreed to form, sponsor, and teach this group.

Aside from a quick "ice breaker" on the first day of class, the students did not have much opportunity to introduce themselves. The forming stage was formal and somewhat cold.

Because the class formed for the purpose of learning about the basics of computer science, there was no time spent on agreeing to an agenda. The professor submitted the syllabus to the group, and the "storming" or goal setting portion was done. Certainly the goal-setting happened long before the first class meeting when the course was initially designed, reviewed, and entered into the university catalog.

During the interviews, the students related how they shed their preconceived notions about the course and settled into a routine. As one student said:

> I thought it was going to be basically educating me on how to use certain programs like most obviously most people know how to use Microsoft Windows and things like that. That is what I thought it was going to be, not really so much the view that you are learning about algorithms and things like that or what computer science is. I guess when I entered the class I had a misconception as to what computer science actually was, but yeah I mean that is what I thought I was going to be learning computer programs and stuff like that that I would potentially use you know in either education or other parts of my life. That is basically what I thought it was.

Another student reflected, "I expected a lot more of learning how to fix computers... It would be like cool if I learned how to fix it on my own instead of the more foundational stuff we have been learning."

Students, during the semester, performed various tasks which were designed to enhance learning. As one student remarked:

> Computers are a lot more than just a big box. It is a tool for problem solving. It has a lot of capabilities but it helps to educate yourself about what it is capable of doing and to familiarize yourself with a lot of different concepts and things like that because it really helps you to utilize it to its fullest potential.

After 14 weeks of class, and a week of finals, the semester came to an end. The students went on their ways, some to other schools, some to drop out of college for a while, some to the military, and some to return for later classes. The class adjourned.

> That I learned a lot more then I expected. There is a lot more I didn't know about but it was explained to me. That was good.

> I got a lot from the class just through the lectures and the tests and homework. I learned a lot more stuff.

> I realize that I learned all this stuff I just didn't think I could learn all of it and I would get there and be successful on the tests and I was like wow I know how everything works now.

> Well I really enjoyed it and I think computer science is better than I expected I absolutely love the way he taught the Christianity in the class also it is something a lot of professors don't take the time to do.

> That was nice that he did take the time to do it you know. I told my friends at school that he was able to tie all that in into and I found Manhattan to be very useful I mean I thought it was and that some people didn't like it as much but I thought it was a nice resource for the class, especially when I was working on the term paper it was the most useful. I think I said it all.

Learner Perceptions

In order to learn, a student needs to take ownership of the topic and needs to be engaged with the class. Without active participation the student will retain very little of what is presented. One tool for active participation is the collaborative software which was provided. Do the students of CSC-150 believe their learning experience was enhanced by the use of Manhattan Virtual Classroom?

Q4: What was the effect of collaborative software on learner perceptions of learning?

For the most part, students thought that Manhattan Virtual Classroom was a positive addition to the course. One student remarked, "It was helpful when Dr. Locklair posted study questions and we could ask questions for the exam and other information."

According to another student:

> I guess because a lot of the questions other people had were the questions I had also. They asked the questions so I didn't have to ask the questions. It would have been a lot of email to Dr. Locklair with questions anyway if Manhattan had not been on there and he responded in a timely manner so that wasn't a problem. I would ask a question and two hours later there would be a response to it. It was quick and I thought it was a nice part of the class.

A different student commented, "In my opinion Manhattan was really useful for the class, and it helped you to stay connected with other students and understand what they were thinking." This comment shows that having collaborative software available does make a positive impact for at least one student.

Finally, a student suggested, "It would be nice to see sometime, maybe, if the entire school could access something like that (MVC) for other programs to incorporate."

Of course, not every student agrees. In answer to the question, *What is your opinion concerning Manhattan Virtual Classroom, its usefulness to you as a student?* one student bluntly said, "Not really useful." Another replied, "I don't think so, to tell you the truth, I really don't."

Q5: Was the course easier or harder because of the collaborative software?

Manhattan Virtual Classroom was seen as a good resource for holding virtual office hours. As one student said, "It is just another source when Professor Locklair isn't there. It is just another means of consultation." "I thought it was [useful] just because it is available and I could tell that Dr. Locklair was good at accessing it often to answer questions, and it is an easy way to answer a question."

Q6: Did using collaborative software reinforce understanding?

Several students spoke of an improved understanding which came from reading posts in the Manhattan Virtual Classroom.

> I think the way Dr. Locklair used it was very useful because he would take the concepts we learned in class and pose a question or two every few weeks, and then give us the option to reply to that or dig into it and answer it based on maybe outside knowledge or something we learned in class that week. I thought that solidified the understanding because it didn't make you sit through class and take notes it made you apply it and write a response to it. I thought it was very useful the way it was used.

On the last day of class a student posted:

> I wanted to say thank you for always being so readily available to answer any questions that we as students have had, and also for doing so in terms we can understand. It has been very helpful as it helped me understand the computer science terms and concepts much easier than if I'd ever have to do it on my own.

Having the Manhattan Virtual Classroom available to this student gave an opportunity to give feedback to the Professor.

Chapter 5: What Does This Mean?

> If, unwarned by my example, any man shall undertake and shall succeed in really constructing an engine ... upon different principles or by simpler mechanical means, I have no fear of leaving my reputation in his charge, for he alone will be fully able to appreciate the nature of my efforts and the value of their results. (Charles Babbage, quoted in Swade, 2000, p. 318)

Did the use of Manhattan Virtual Classroom lead to enhanced community, improved grades, and more retention of the materials presented in CSC-150, Foundations of Computer Science? Based on the one-sided flow of messages, the lack of participation, and the unread messages, the impact of the tool on community formation may have been minimal. Does this mean that the tool ought not be used to enhance classroom discussion? No, it means the way the tool is used ought to be studied and modified so to make it more valuable to the students for forming a learning community.

Did the random group of students collectively taught in the sections of CSC-150 become a true learning community? From all indications, they did not. With very few exceptions, they came in to the class as individuals, interacted with the professor as individuals, and left the class at the end of the semester as individuals.

Is Manhattan Virtual Classroom, or other collaborative software, appropriate for all classes and all professors? Again the answer is “No.” A professor who does not encourage classroom participation will probably not see participation in a web-based tool such as Manhattan Virtual Classroom. A class that is designed to teach the grammar of the trivium via lectures rather than experimentation may also not benefit greatly from such a tool.

In this chapter I shall discuss how the unfolding data changed the research emphasis. I will also review the data gathered about the Spring, 2004, classes of CSC-150 and relate these observations to the experience of other researchers. I will end this chapter by making several recommendations about further research and some practical suggestions for increasing participation as teachers begin to use collaborative software tools such as Manhattan Virtual Classroom.

Changes in the Study

When I had initially proposed this research, it was to compare the final grades of two sections of a given class. One section would use the collaborative tool, the other section would not. At the urging of the committee, and properly so, the research focus changed to one of community building. In the end, however, the qualitative data collected showed the underlying question was not even that of community building, but of class participation.

Without class participation and interaction, no community can be developed. The advantages of class participation, the exercise of newly gained skills, the integration of new concepts into pre-existing schemata, and the reinforcing of knowledge, are lost if students remain passive. I remain convinced that the goal ought not to be merely passing a class, but to actually learn something in the long run. I remain convinced that passive students are not learning the material of a class, but are retaining enough to pass a test.

In analyzing this data, I have already made changes in my own teaching style and my own use of the Manhattan Virtual Classroom. Several of the students in the sections of CSC-150 which were studied have subsequently been in sections of CSC-175, Advanced Computer Applications, which I have taught. I've seen several make more use of the Manhattan Virtual Classroom then they did in CSC-150, and several use it in the same way as in CSC-150.

Changes to be Tried Because of the Study

Classroom Changes

I've become much more aware of the need to involve students in discussions and classroom activities. A lecture is both safe and easy, and on days when the students do not wish to participate, a lecture allows them to be passive. Passive learning is no learning at all.

From the beginning of the semester I make it a point to learn each student's name and some trivia about them. During the semester I use that trivia when framing questions, both to show they are important and to connect them to the class. I also comment in class about postings the students have made in the Manhattan Virtual Classroom, showing that I read these posts and find them relevant.

In several courses I have made more use of small group activities which allow students an opportunity to use newly gained knowledge. In the upper division courses I ask students to prepare and teach segments of each unit. Students also submit test questions which are then incorporated into the unit exams.

MVC Changes

Since reading several books about small groups and seeing the problems of using a single large group in the discussion area of MVC, I have made much more use of the "Team Discussion" option. At the beginning of the semester I divide each class section into groups of 7 to 10 students. I then pose questions for them to discuss with their team mates. Several of these discussions have continued in the classroom as the students meet face-to-face.

Participation Levels

The experiences of other researchers, in particular Murray Goldberg (1997) of the University of British Columbia, suggested that first year computer science students would be less likely to use a collaborative software tool such as Manhattan Virtual Classroom. He suggested:

> Students in the first year at the University of British Columbia are, in many ways, quite different than their more mature third year counterparts. Our experience has shown that they tend to be less likely to participate in class and take advantage of the resources available to them.

Students at Concordia University Wisconsin acted in a way consistent with this observation. As shown in the preceding chapter, the number of login activities, postings, and messages read was quite small.

Group Size

Mills (1967, p. 64), reports the observation, "as the [group] size increases, the consequences of alienating a single member becomes less and less severe." This agrees with Shaw (1981, p. 169):

> More significantly, the distribution of participation varies with group size. Group members report that they have fewer chances to speak in larger groups, and their feelings are reflected in the pattern of communication in the group. As the size of the group increases, larger and larger proportions of group members participate in less than their "fair share." A few members tend to dominate the discussion, with others participating relatively less as size increases.

The CSC-150 Manhattan Virtual Classroom had one large group of 38 members, plus the professor and observer. Although the software allowed for the formation of "teams," the professor chose not to use this particular function.

Student participation levels were consistent with the findings of Mills, Shaw, and other researchers. This was a large group, and even the students who were most active posted typically less than one message per week, and the class as a whole averaged one posting every two weeks of the semester.

Active vs. Lurkers

Only student read all the postings within the Manhattan Virtual Classroom, while two additional students left fewer than 25 messages unread. A fourth student read all but 89, while the next closest student left 231 of the 433 messages unread. Dr. Locklair posted 173 messages, and the system administrator posted an additional 15. This means that students posted 245 messages.

For the most part, students lurked (were non-participants) even when they read the messages. Without student postings, and without students reading the posts, the tool can not be effectively used.

What can a professor do to increase communication and motivate students to use collaborative software? Palloff and Pratt (2001, p. 30) suggest several guidelines for improving participation in the collaborative classroom, including:

- Teach students about online learning.
- Be a model of good participation by logging on frequently and contributing to the discussion.
- Set limits if participation wanes or if the conversation heads in the wrong

direction.

- Contact non-participating students and invite them to participate.
- Create a warm and inviting atmosphere that promotes the development of a sense of community.

Did this happen in CSC-150? Yes, in some respects. Learning to use collaborative software requires time and effort on the part of both student and faculty. This being the first time the tool was used in the class, posting guidelines, and even the formal use of the tool had to be defined. Does the tool become a major part of the class, or is it an adjunct to the classroom, an optional item? Certainly in the case of the traditional class using the tool to enhance communication, participation is not as critical as when the tool is the only means of communication between students or students and faculty.

Required Postings vs. Voluntary

The goal is to motivate students to use collaborative software to enhance their learning by forming a learning community. However, students need to be enticed to use the tool. A blanket, "You shall post three messages a week," does not truly give direction to the student. Mandatory posts are good, but just posting for the sake of posting does not enhance learning nor contribute to the formation of a learning community..

The online classroom is still a classroom. The same techniques which work to begin and continue discussions in a traditional classroom should work in an online classroom. According to McKeachie (1978, p. 38), discussions may begin with a shared experience, a question, or a controversy. Merely asking, "Any questions?" suggests to the student that the professor does not desire an answer.

Dr. Locklair did provide good discussion questions for the class. The first one he posed, "Is knowing how to effectively use a computer system to solve problems as important as knowing how to read and write? Post your thoughts and comments." Questions such as this should encourage students to think about the course material and go beyond merely cramming for a test.

What the CSC-150 student needed were directions and expectations for using the tool and guidelines for participation. These directions, expectations, and guidelines were not specifically spelled out in either the initial course materials or in subsequent postings in the Manhattan Virtual Classroom. Students as well as faculty must be taught how to use the tool (Palloff and Pratt, Porter, Conrad and Donaldson). Without such training, the students will be reluctant to post or read messages.

One way of providing such directions and expectations is to specifically give the completion criteria in the question. The first question could have been stated as, "Is knowing how to effectively use a computer system to solve problems as important as knowing how to read and write? Post your thoughts and comments. Respond to at least two other students (which may mean you will make multiple posts in this topic thread)."

Finally, trying to keep track of thirty or more active participants in a web-based discussion is a daunting task. Discussions could be held in smaller groups or "teams" using other Manhattan Virtual Classroom modules. Each class could have been split into two or three groups thus making participation both easier and more subject to peer encouragement.

Future Research

Longitudinal Study

The Goldberg (1997) research suggested that upper division students, because of their greater maturity or familiarity with university life, tend to make greater use of collaborative software. A future study could track collaborative software usage patterns for a group of students from their freshman through senior years. Might the same group of students make more use of the tool as they both gain familiarity with the tool and increase in knowledge, wisdom, stature, and maturity?

Modifying the Population

This study dealt with a computer science class which was taught to non-computer science students. Would students in the computer science major react in the same manner as students outside the major? Would a class in an entirely different major area, such as in Communications or English, experience the same participation levels?

Modifying Student Training

Initially students were given a short demonstration of the Manhattan Virtual Classroom software, including a session during which they verified their userid and passwords. They were given encouragement to use the forum to ask questions and to respond to other student's posts, but were not directly told how or what to post. What would happen to student participation levels if they were given more training in the software and were taught about active learning (constructionism, Problem Based Learning, Social Cognitive Development, etc.) as part of the course introduction?

Modifying Instructor Factors

The course instructor had little practical training in using a collaborative classroom tool, although he had great theoretical knowledge. How would instructor training affect student participation. What training ought to be given beyond the basic "nuts and bolts" of uploading materials, posting messages, and course configuration? Does the online classroom essentially differ from the traditional classroom as regards purpose, that is to clearly communicate new information to students?

What would happen to student participation if the instructor also taught in a more active style, including using a framework of constructionism, Problem Based Learning, and other collaborative modes of teaching? Would the students, as encouraged by the instructor, become more involved in the class and, as they helped each other, form a learning community?

I believe that a group of students would actually form a learning community if the classroom was run in a more active style. From the changes I made in my classes which encourage teamwork, I have seen more of a community develop. This community is seen not only in classroom activities, but in participation levels for departmental activities such as *Geek Night at the Movies* .

Modifying the University Environment

As of the Spring 2004 semester, every class at Concordia University Wisconsin has a web presence available using WebCT. Most professors have not taken advantage of WebCT because of training and familiarity issues. The College of Education and the Information Technology Department have provided some seminars on this tool, but the seminars were poorly attended. The Faculty Development Committee provided a

"Lunch and Learn" session for 24 faculty members concerning the use of technology in the classroom, including a section on WebCT, but there was no follow-up to that presentation.

The early adopters of this technology need to mentor their peers as later adopters begin to use this type of program. Some of the training and mentoring would be informal and on an ad-hoc basis, but, possibly at the Department level, such training could be formalized.

Τετελεσται *(Tetelestai): It is Finished*

CSC-150, Foundations of Computer Science, has come to an end for the Spring semester of 2004. There are several lessons learned from the experience of using collaborative software. First, students must be engaged and motivated to use the resource. They need training on both the software, and more importantly, on the reason for using the software. Second, even in a class where the software is not the primary means of communication, it still provides for virtual office hours and consultation between students and professor. Many students who did not participate to a great extent felt the tool had value. Third, the professors and instructors need training as to the best way to utilize the tool. The same techniques used in the traditional classroom need to be modified for use in the online classroom. Professors must encourage students to take an active role in their own learning.

The experience of using Manhattan Virtual Classroom in CSC-150 was positive. It will be used again as another pathway for communication between students and professor, to help build knowledge and to foster a sense of community.

Bibliography

ACM & IEEE (2001). *Computing Curriculum 2001* Retrieved January 15, 2002, from http://www.computer.org/education/cc2001/final/index.htm

Anderson, T., & Garrison, R. (1998). Learning in a networked world: New roles and responsibilities. In C. C. Gibson (Ed.)*Distance learners in higher education: Institutional responses for quality outcomes*(pp. 97-112). Madison, WI: Atwood Publishing.

Bandura, A. (1977). *Social learning theory.* Englewood Cliffs, NJ: Prentice-Hall.

Bandura, A. (1986). *Social foundations of thought & action: A social cognitive theory.* Edgewood Cliffs, New Jersey: Prentice-Hall.

Bartlett, F. C. (1967). *Remembering: a study in experimental and social psychology* London: Cambridge University Press.

Berners-Lee, T. (2000). *Weaving the web: The original design and ultimate destiny of the world wide web.* New York: Harper Collins.

Bigge, M. L., & Shermis, S. S. (1992). *Learning theories for teachers (5th. ed.)* New York: Harper Collins.

Blackboard (2004). Blackboard [Computer software]. Retrieved November 15, 2004 from www.blackboard.com

Brookshear, J. G. (2005). *Computer science: An overview (8th. ed.)* Boston: Pearson Addison Wesley.

Brown, J. S., & Duguid, P. (1995). *Universities in the digital age.* Retrieved July 15, 2003, from http://www2.parc.com/ops/members/brown/paper/university.html

Brown, J. S., & Duguid, P. (2002). *The social life of information*(2nd ed.). Boston: Harvard Business School Press.

Campbell-Kelly, M. & Aspray, W. (2004). *Computer: A history of the information machine (2nd. ed.)* Boulder, CO: Westview Press.

Carlson, D., Guzdial, M., Kehoe, C., Shah, V., & Stasko, J (1996). WWW interactive learning environments for computer science education. *Proceedings of the Special Interest Group for Comptuter Science Education, 1996, Philadelphia* (p. 290 - 294). New York: ACM Press.

Carroll, J. B. (1964). *Language and Thought*. Edgewood Cliffs, New Jersey: Prentice-Hall.

Ceruzzi, P. E. (1998). *A history of modern computing.* Cambridge, MA: MIT Press.

Christiansen, E., & Dirckinck-Holmfeld, L. (1995). *Making distance learning cooperative.* http://www-cscl95.indiana.edu/cscl95/chritia.html

Christensen, W., & Suess, R. (1989). *The birth of the BBS*. Retrieved September 28, 2003, from http://www.chinet.com/html/cbbs.html

Coates, K. E., Dvorak, D. L., & Watts, R. M. (1981). *An overview of BLN: A Bell Laboratories computing network*. Retrieved October 27, 2004, from http://www.acm.org.

CompuServe (2003). *CompuServe interactive services overview.* Retrieved August 14, 2003, from http://webcenters.compuserve.com/compuserve/menu/about.jsp

Concordia University Wisconsin (2002). *Catalog*. Mequon, WI: Author.

Conrad, R. & Donaldson, J. A. (2004). *Engaging the online learner: Activities and resources for creative instruction*. San Francisco: Jossey-Bass Publishers.

Creswell, J. W. (1998). *Qualitative inquiry and research design: Choosing among five traditions*. Thousand Oaks, CA: SAGE Publications.

Denenberg, S. A. (1978). *A personal evaluation of the PLATO system ACM SIGCUE Bulletion*, vol. 12 no. 2, 3-10.

Family Educational Rights and Privacy Act (FERPA) (20 U.S.C. § 1232g; 34 CFR Part 99). Retrieved November 12, 2003, from http://www.ed.gov/policy/guid/fpco/ferpa/index.html

Federal Aviation Administration (1999). *Aviation instructor's handbook* (FAA-H-8083-9). Washington, DC: U.S. Government Printing Office.

Free Software Foundation (2004). *The free software definition* Retrieved November 13, 2004, from http://www.gnu.org/philosophy/free-sw.html

Fodor, J. A. (1975). *The language of thought.* Cambridge, MA: Harvard University Press.

Glover, J. A. & Bruning, R. H. (1990). *Educational psychology: Principles and applications (3rd. ed.)* New York: Harper Collins.

Goldberg, M. W. (1997). WebCT and first year: Student reaction to and use of a web-based resource in first year computer science. *Proceedings of the 2nd conference on integrating technology into computer science education, Uppsala, Sweden* (pp. 127-129). New York: ACM Press.

Guzdial, M. (1997). *LBD conceptual: cognition: theory* Retrieved August 8, 2003, from http://www.cc.gatech.edu/deutech/LBD/constructivism.html

Guzdial, M., Rick, J., & Kerimbaev, B. (2000). Recognizing and supporting roles in cscw. *Proceedings of the Computer Supported Cooperative Work 2000, Philadelphia.* New York: ACM Press.

Hafner, K., & Lyon, M. (1996). *Where wizards stay up late: The origins of the Internet.* New York: Simon & Schuster.

Harasim, L., Hiltz, S. R., Teles, L., & Turoff, M. (1996). *Learning networks* Cambridge, MA: MIT Press.

Hiltz, S. R., & Wellman, B. (1997). Asynchronous learning networks as a virtual classroom. *Communications of the ACM*, vol. 40 no. 9, 44-49.

Hofstadter, D. R. (1979/1999). *Gödel, Escher, Bach: An eternal golden braid.* New York: Basic Books.

Hunt, C. (1998). *TCP/IP network administration (2nd. ed.).* Sebastopol, CA: O'Reilly and Associates.

Johnson-Laird, P. N. (n.d.). *Philip N. Johnson-Laird's homepage* Retrieved October 27, 2004, from http://webscript.princeton.edu/˜psych/PsychSite/˜phil.html

Johnson-Laird, P. N. (1988). *The computer and the mind: An introduction to cognitive science*. Cambridge, MA: Harvard University Press.

Kehoe, B. P. (1992). *Zen and the art of the internet: A beginner's guide to the internet* (1st ed.). Retrieved September 1, 2003, from http:www.eff.org/Net_culture/Net_info/Guidebooks/Zen_and_Internet.

Kernighan, B. W., & Pike, R. (1999). *The practice of programming*. New York: Addison Wesley Longman.

Kim, A. J. (2000). *Community building on the web: Secret strategies for successful online communities.* Berkeley, CA: Peachpit Press.

Knuth, D. E. (2001). *Things a computer scientist rarely talks about.* Stanford, CA: CSLI Publications.

Krathwohl, D. R. (1998). *Educational & social research: An integrated approach* (2nd ed.). New York: Addison Wesley Longman.

Kroll, E. (1989). *RFC1118 - Hitchhikers guide to the Internet.* Retrieved September 2, 2003, from http://www.rfc-editor.org

Lacoursiere, R. B. (1980). The life cycle of groups: Group development stage theory. New York: Human Sciences Press.

Leiner, B. M., Cerf, V. G., Clark, D. D., Kahn, R. E., Kleinrock, L., Lynch, D. C., Postel, J., Roberts, L. G., & Wolff, S. S. (1997). The past and future history of the Internet. *Communications of the ACM*, vol. 40 no. 2, 102-108.

Lessig, L. (2001). *The future of ideas: The fate of the commons in a connected world.* New York: Random House.

Locklair, G. H. (2001). *Foundations of computer science: A survey of computer science via its grand ideas*. Unpublished doctoral dissertation, Nova Southeastern University.

Locklair, G. H. (2003). Course syllabus for CSC-150: Foundations of computer science. Retrieved September 2, 2003, from http://www.cs.cuw.edu/csc/csc150

Luka, J., & McLoughlin, C. (2004). Using online forums to support a community of learning. *AACE Journal*, Vol. 12, Iss. 4. Retrieved December 18, 2004 from: http://www.aace.org/pubs/aacej/current.cfm

Mayer, R. E. (1992). *Thinking, problem solving, cognition (2nd ed.).* New York: W. H. Freeman and Company.

McGrath, J. E, & Hollingshead, A. B. (1994). *Groups interacting with technology.* Thousand Oaks, CA: Sage Publications.

McKeachie, W. J. (1978). *Teaching tips: A guidebook for the beginning college teacher [7th ed].* Lexington, MA: D. C. Heath and Company.

Microsoft (2003). *Internet Explorer.* Retrieved on December 20, 2004, from: http://www.microsoft.com/windows/WinHistoryIEmspx

Mills, T. M. (1967). *The sociology of small groups.* Englewood Cliffs, NJ: Prentice-Hall.

Minsky, M. (1985). *The society of mind.* New York: Simon & Schuster.

Narmontas, S. (2003). Manhattan Virtual Classroom (Version 2.0) [Computer software]. Springfield, MA: Western New England College. Retrieved August 1, 2003, from http://manhattan.sourceforge.net

Narmontas, S. (2003). *The manhattan virtual classroom 2.0 teacher's reference.* Springfield, MA: Western New England College. Retrieved August 1, 2003, from http://manhattan.sourceforge.net

National Center for Education Statistics (2003). Table 421. Public schools and instructional rooms with access to the Internet, by selected school characteristics: 1994 to 2002 *Digest of Education Statistics, 2003.* Retrieved December 20, 2004, from http://nces.ed.gov/programs/digest/d03/list_tables4.asp

Negroponte, N. P. (1995). *Being digital.* New York: Alfred A. Knopf.

Novak, J. D., & Gowin, D. B. (1984). *Learning how to learn.* New York: Cambridge University Press.

Papert, S. (1993). *The children's machine.* New York: Basic Books.

Palazzolo, C. S. (1988). The social group: Definitions. In R. S. Cathcart, & L. A. Samovar (Eds.). *Small group communication: A reader* (5th Ed., pp. 6-19). Dubuque, IA: William C. Brown Publishers.

Palloff, R. M., & Pratt, K. (1999). *Building learning communities in cyberspace: Effective strategies for the online classroom.* San Francisco: Jossey-Bass Publishers.

Palloff, R. M., & Pratt, K. (2001). *Lessons from the cyberspace classroom: The realities of online teaching.* San Francisco: Jossey-Bass Publishers.

Palloff, R. M., & Pratt, K. (2003). *The virtual student: a profile and guide to working with online learners.* San Francisco: Jossey-Bass Publishers.

Pinker, S. (2002). *The blank slate: The modern denial of human nature.* New York: Penguin Books.

Porter, L. R. (1997). *Creating the virtual classroom: Distance learning with the Internet* New York: John Wiley & Sons.

Pratt, T. W., & Zelkowitz, M. V. (2001). *Programming languages: Design and implementation* (4th. ed.). Upper Saddle River, NJ: Prentice Hall.

Pressman, R. S. (2003). *Software engineering: A practitioner's approach* (6th. ed.). New York: McGraww-Hill Book Company.

Quarterman, J. S., & Hoskins, J. C. (1986). Notable computer networks. *Communications of the ACM*, vol. 29, no. 10, 932-971.

Raymond, E. S. (1999). *The cathedral & the bazaar: Musings on linux and open source by an accidental revolutionary.* Sebastopol, CA: O'Reilly and Associates.

Rheingold, H. (1993). *The virtual community: Homesteading on the electronic frontier.* Reading, MA: Addison-Wesley Publishing Company.

Resnick, E. (2003). *Design for communication: Conceptual graphic design basics.* Hoboken, NJ: John Wiley & Sons, Inc.

Sayers, D. L. (1947). *The lost tools of learning* Retrieved on December 20, 2004 from: http://www.gbt.org/text/sayers.html

Scardamalia, M. (2004). CSILE/Knowledge Forum. In *Education and technology: An encyclopedia* (pp. 183-192). Santa Barbara: ABC-CLIO. Retrieved on November 10, 2004 from: ikit.org/fulltext/CSILE_KF.pdf

Scardamalia, M. & Bereiter, C. (1993). *Technologies for knowledge-building discourse. Communications of the ACM*, vol. 36, no. 5, 37-41.

Scardamalia, M. & Bereiter, C. (1996). *Student communities for the advancement of knowledge. Communications of the ACM*, vol. 39, no. 4, 36-37.

Schneider, G.M., & Gersting, J.L. (1999). *An invitation to computer science* (2nd ed.). Pacific Grove: PWS Publishing.

Searle, J. R. (1998). *Mind, language and society: Philosophy in the real world*. New York: Basic Books.

Shannon, C. E. (1948). *A mathematical theory of communication. The Bell System Technical Journal*, vol. 27, 379-423, 623-656.

Shaw, M. E. (1981). *Group dynamics: The psychology of small group behavior* (3rd. ed.). New York: McGraww-Hill Book Company.

Staley, H. A. (1977). *The tongue and quill: Communicating to manage in tomorrow's Air Force [3rd ed.].* Montgomery, AL: Air Command and Staff College.

Stallings, W., Van Slyke, R. (1998). *Business Data Communications*. Upper Saddle River, NJ: Prentice Hall.

Swade, D. (2000). *The difference engine: Charles Babbage and the quest to build the first computer.* New York: Penguin Books.

United States v. American Telephone and Telegraph, 552 F. Supp. 131, 150 (DDC 1982)

United States Library of Congress (2001). *The Samuel F. B. Morse Papers at the Library of Congress*. Retrieved August 1, 2003 from http://memory.loc.gov/ammem/sfbmhtml/sfbmhome.html

Veith, G. E. & Kern, A. (1997). *Classical education: Toward the revival of American schooling*. Washington, DC: Capital Research Center.

Vygotsky, L. S. (1962). *Thought and Language*(E. Hanfmann & G. Vakar, Ed. and Trans.). Cambridge, MA: MIT Press. (Original work published 1934)

Wallace, P. M. (1999). *The psychology of the Internet.* New York: Cambridge University Press.

Watts, R. M. (2001). *The slingshot syndrome: Why America's leading technology firms fail at innovation.* Lincoln, NE: Writers Club Press, an imprint of iUniverse.

WebCT (n.d.). WebCT [Computer software]. Retrieved November 15, 2004 from http://www.webct.com

Williams, S. (2002). *Free as in freedom: Richard Stallman's crusade for free software.* Sebastopol, CA: O'Reilly and Associates.

Wolz, U., Palme, J., Anderson, P., Chen, Z., Dunne, J., Karlsson, G., Laribi, A., Mannikko, S., Spielvogel, R., & Walker, H. (1997). Working Group Report. ACM: ITiCSE 97, pp.51-69.

Appendix 1 - Project Letter for CSC-150 Students

Spring Semester 2004
Greetings,

I am Professor Jeff Williams, one of the Computer Science professors. As part of my Ph.D. studies, I am conducting research on educational technology and the tools available for students.

Two sections of CSC-150, Foundations of Computer Science, will be involved in a study of software which can be used by students for discussing concepts learned in class, homework, and projects.

This study will look at the effect of the use of Manhattan Virtual Classroom the development of a learning community by encouraging communication. You will find that discussing CSC-150 with your fellow students will help you to better understand the materials presented. Only as you use the information you have received can you truly call it your own, apply it to other classes, and develop a true appreciation of the grand ideas of computer science. Manhattan Virtual Classroom should be used in addition to the other methods of communication between you and other students.

Beginning about the tenth week of class, I will interview each member of both class sections to get your reactions to both CSC-150 and Manhattan Virtual Classroom. You may decline to participate if you so desire, however your experiences and opinions are important and will add to the study.

Any information I obtain which is of a personal nature, such as your student ID number (F00 number) or any other identification will be kept confidential. I will not use your name or any other identifier within the research report. Although I will keep an archive of the Manhattan Virtual Classroom discussions, I will not release the archives to anyone else without first removing any identifying information.

Your participation in this study will not affect your grades in CSC-150 or any other class except to the extent that any improved collaboration between you and other students will help your understanding of the material.

You will be able to monitor the progress of the research which is taking place, and will be able to read the resulting dissertation if you so desire.

Appendix 2 - CSC-150 Web Resources

Course notes will be available on a day-by-day basis on the CS Department web server (http://www.cs.cuw.edu). You may access the web server from any computer with Internet access: at school, work or home.

To obtain the day's course outline, visit the appropriate unit link (eg, Unit 1) and then visit the appropriate day.

If you receive a "not found" error, that day's outline has not yet been uploaded to the web server. Check back again later. (But, see the following note...)

Note: due to the "proxy server" utilized by CUW's IT department - it may be necessary to "refresh" a page for it to be shown correctly. If a page looks "old" or does not display correctly, use the [Refresh] button on the browser. If you believe a page should be on the server, yet you receive a "not found" error, try the [Refresh] command.

Each day's notes will be available either as a Web .html file or as a Word .doc file. Follow the appropriate link for each unit. Web .html notes are quick and accessible whether or not you have MS Word on your computer. The Word data files are best for printing. CSC 150 students should print out the Word notes for use in class.

Streaming audio of each class period is also available in the Real format. To hear the audio, your computer must have the free Real player.

If your computer has Microsoft Word and PowerPoint installed, using Microsoft's Internet Explorer will allow you to easily access both the Word and PowerPoint files. Using another browser will result in a prompt to either open or save the Word and PowerPoint files. If the proper software is installed, selecting open should place the data file in the appropriate application for viewing.

The Computer Science Department would appreciate it if you would turn on the double-side printing option in order to conserve paper.

If you have questions, feel free to ask a lab assistant (in S109) or the CS system administrator (in S118C) for help.

Appendix 3 - CSC-150 Sample Worksheet

Introduction to Computer Systems
III. Putting Computer Systems to Use
A computer is a tool for solving problems; how is it used to solve problems?
?

A. User Perspective
1. [preliminary] Power up computer
a. self test
b. Windows is run; you know Windows is running by the Win desktop.
2. [preliminary] Network Login
a. use File Server's, or shared, resources (eg ...
3. Enter a ___________ to _______ a S/W application
Ex: [START | Programs | Applications |GeoClock |Geoclock]
Easy to do; need to know concepts underlying this also (B. & C.)

B. Computer Perspective (‘‘behind the scenes - after A.3.’’)
1. H/W must _______ S/W
copy S/W instructions from storage to memory
Ex: GeoClock instructions are copied from file server's storage to
my PC's memory (via the LAN)
2. H/W must _________ S/W
CPU follows the software instructions (algorithms) within memory
Ex: My PC follows the GeoClock instructions in its memory

C. Problem solving (CS Perspective)
1. Programmer created algorithms (software)
2. User interacts with software by entering data and issuing commands
3. Ex: note algorithms, abstraction, and automation in GeoClock
algorithm
abstraction
automation

IV. Computer Systems (Again)

A. What Do Computers Do?
1. Input Data
2. Perform __________ Operations (+, -, *, /)
3. Perform __________ Operations - comparisons (>, <=, etc)
together 2. & 3. are ‘‘processing’’
4. Output Information
5. Store ‘‘Stuff’’
6. Follow _____________

Anything unique, special or magical about what computers do? ?
B. Advantages of computers
Why have a computer solve this problem instead of people?
Computers possess no special capabilities compared to people
1.
Ex: ...
2.
Ex: ...

C. Definitions
1. Data - raw facts input to the computer
2. Information - useful, processed data used by people to make decisions
Computers convert data into information; this is ‘‘processing’’
Good jobs ...
Bad jobs ...
3. Software - instructs the H/W _____ to convert data into information
S/W explains to the H/W how to process using arithmetic and logical ops
4. Command - issued by user
a. specifies which _____ to use for processing (Ex: ...
b. specifies which _______ to use during processing (Ex: ...
Don't confuse S/W and Command
S/W is created by programmers and tells the computer ...
Commands are entered by users and tell the computer ...

```
      5.  Byte - a measure of memory or storage capacity
          one byte can ``hold'' one character of data
      6.  Small Computer
          Supports only one user at a time  (Ex: ...
      7.  Large Computer
          Supports multiple, simultaneous users  (Ex: ...
      8.  Network  (see pages 1-13ff in C.C)
          ...

   D. Software is key to the ____________ ______ of a computer system
      H/W is obvious and visible; but S/W is more important
         Analogy: Law & Gospel.  Law (like H/W) is more visible [Romans 2:14];
            while Gospel (like S/W) is more important! [Romans 1:16]
      S/W ``drives'' the decision-making process
      1.  Why - what problem do I need to solve?  [people]
      2.  What S/W will solve this problem?  [S/W]
      3.  What H/W will run this S/W?  [H/W]

   E. Capabilities and Limitations
      ...

A computer is put to use by running S/W applications
Computers are used to solve problems for productivity and efficiency reason
```

Appendix 4 - Perl Source Code

These Perl programs were written to manage and analyze the interview transcripts. TRANSPOP.PL is used to load the interview transcript database from the ASCII file produced by any editor or word processor. ENCODE.PL, the program used to annotate the interview transcripts, uses the Perl/Tk module for manipulating screens. TRANSTEST.PL and CODETEST.PL are used to print the transcripts or coded lines from each transcript. MySQL is the underlying database engine.

TRANSPOP.PL

```
#!/usr/bin/perl -w

# transpop.pl
# jeff williams
# 5/17/2004
# transcript database population

# describe transcript;
# +----------+-------------+------+-----+---------+----------------+
# | Field    | Type        | Null | Key | Default | Extra          |
# +----------+-------------+------+-----+---------+----------------+
# | int_id   | int(4)      |      | PRI | NULL    | auto_increment |
# | int_name | varchar(30) | YES  |     | NULL    |                |
# | speaker  | char(1)     | YES  |     | NULL    |                |
# | response | blob        | YES  |     | NULL    |                |
# | line_no  | int(4)      | YES  |     | NULL    |                |
# +----------+-------------+------+-----+---------+----------------+
#
# describe coding;
# +---------+--------+------+-----+---------+-------+
# | Field   | Type   | Null | Key | Default | Extra |
# +---------+--------+------+-----+---------+-------+
# | int_id  | int(4) | YES  |     | NULL    |       |
# | code_id | int(4) | YES  |     | NULL    |       |
# | line_no | int(4) | YES  |     | NULL    |       |
# +---------+--------+------+-----+---------+-------+
#
# describe codes;
# +-----------+--------------+------+-----+---------+----------------+
# | Field     | Type         | Null | Key | Default | Extra          |
# +-----------+--------------+------+-----+---------+----------------+
# | code_id   | int(4)       |      | PRI | NULL    | auto_increment |
# | code_desc | varchar(255) | YES  |     | NULL    |                |
# +-----------+--------------+------+-----+---------+----------------+

system('clear');
print "Enter database password: ";
$dbf_pass = <STDIN>;
chomp($dbf_pass);

use DBI;
$dbh = DBI->connect('DBI:mysql:interviews','cfiaime',"$dbf_pass");

system('clear');

$dbs = $dbh->prepare("select distinct int_id,int_name from transcript
        order by int_id");
$dbs->execute;

print "Current transcripts in database0;
while (@row = $dbs->fetchrow_array) {
        print ("$row[0]$row[1]0);
```

```
}
print "0;
print "Interview database population0;
print "Enter interview file: ";
$infile = <STDIN>;
chomp($infile);
print "Enter interview ID number: ";
$int_id = <STDIN>;
chomp($int_id);

open (INFILE, "$infile");

$line_no = 0;
while (<INFILE>) {
        if (length($_) > 0 && substr($_,0,1) ne 'A' && substr($_,0,1) ne 'Q') {
                print "$_ 0;
                $int_name = $_;
                chomp($int_name);
        } else {
                @line = split //,$_;
                $dbs = $dbh->prepare("insert into transcript
                        set int_id='$int_id',
                        int_name='$int_name',
                        speaker='$line[0]',
                        response='$line[1]',
                        line_no='$line_no'");
                $dbs->execute;
                $line_no = $line_no+1;
        }
}
```

ENCODE.PL

```
#!/usr/bin/perl -w

# program: encode.pl
# author: jeff williams
# date:   5/29/2004
# Interview transcript markup / commentary system
# revision history:
#       5/29/2004 - initial program

# This code released under GPL 2.0

# describe codes;
# +-----------+--------------+------+-----+---------+----------------+
# | Field     | Type         | Null | Key | Default | Extra          |
# +-----------+--------------+------+-----+---------+----------------+
# | code_id   | int(4)       |      | PRI | NULL    | auto_increment |
# | code_desc | varchar(255) | YES  |     | NULL    |                |
# +-----------+--------------+------+-----+---------+----------------+
#
# describe coding;
# +----------+--------+------+-----+---------+-------+
# | Field    | Type   | Null | Key | Default | Extra |
# +----------+--------+------+-----+---------+-------+
# | int_id   | int(4) | YES  |     | NULL    |       |
# | code_id  | int(4) | YES  |     | NULL    |       |
# | line_no  | int(4) | YES  |     | NULL    |       |
# | comments | blob   | YES  |     | NULL    |       |
# +----------+--------+------+-----+---------+-------+
#
# describe transcript;
# +----------+-------------+------+-----+---------+-------+
# | Field    | Type        | Null | Key | Default | Extra |
# +----------+-------------+------+-----+---------+-------+
# | int_id   | int(4)      |      |     | 0       |       |
# | int_name | varchar(30) | YES  |     | NULL    |       |
# | speaker  | char(1)     | YES  |     | NULL    |       |
# | response | blob        | YES  |     | NULL    |       |
# | line_no  | int(4)      | YES  |     | NULL    |       |
```

```
# +----------+------------+------+-----+--------+-------+

use strict;
use Tk;
use DBI;

use vars qw( $mw $dbh $dbs $dbs1 $dbs2 $dbs3 $dbs4 $int_id $int_name
        $trans_line $trans_txt
        $trans_speak $trans_resp $coding_line $coding_id
        $coding_comments @code_id @code_desc
        @comm @comm_id @comm_line @comm_desc
        @code @trans_id @trans_name @trans $lineout);
use vars qw( $i $rowcnt @row $lb $tb1 $tb2 $frm @row1 $tag_indent $tag_lineno );
use vars qw( $lineno $frm1 $frm1_wide $frm2 $tb2a $tb2a_msg );
use vars qw( $frm1a $frm1b $frm1c $frm1d $frm1e $frm1f );
use vars qw( $tb2a1 $tb2a2 $tb2b1 $tb2b2 $tb2b3 $tb2c1 $tb2c2 );
use vars qw( $tb2d $tb2d1 $tb2e1 $tb2e2 );
use vars qw( $tb2f1 $tb2f2 );
use vars qw( $save_int_line $save_comments $save_code );
use vars qw( $frm3 $frm3a $frm3a1);
use vars qw( $frm4 $frm4a $frm4b);
use vars qw( $dbs5 @analyzed $analyzed_cnt );
use vars qw( $menubar $reports $report1 $report2 );
use vars qw( $passwd );

open(PASSWD,'passwd.txt');
$passwd=<PASSWD>;
chomp($passwd);
close(PASSWD);

# set int_id for initial screen population;
$int_id = -1;

int_list();

$lineout = "no interview selected";

$mw = MainWindow->new(
        -title=>"Encode Transcript System",
        -background=>'beige');

# # create REPORTS menu entry
# $mw->configure(-menu=>$menubar = $mw->Menu);
# $reports = $menubar->cascade(-label=>'Reports');
# $report1 = $reports->cascade(
#        -label=>'Interview transcript',
#        -command=>\&report1);
# $report2 = $reports->cascade(
#        -label=>'Transcripts by coding',
#        -command=>\&report2);


# listbox of possible interviews
$mw->Label(
        -text=>'Interview Subjects',
        -background=>'ivory1',
        ) ->pack(-side=>'top', -anchor=>'n');

$lb = $mw->Scrolled('Listbox',
        -background=>'ivory1',
        -scrollbars=>'e',
        -selectmode=>"single",
        -height=>3,
                ) ->pack(-side=>"top",-expand=>0,-fill=>'x',
                -anchor=>'n');
$lb->delete(0,'end');
$lb->insert('end',@trans );

# bottom menu frame
$frm2 = $mw->Frame(
        -relief=>'raised',
        -background=>'white',
```

```
        ) ->pack(-side=>'bottom');
# activate search of list box
        $frm2->Button(
                -background=>'lightblue',
                -text=>'Load Interview',
                -activebackground=>'lightblue',
                -activeforeground=>'red',
                -command=> \&lbsearch,
                ) ->pack(-side=>"left",-anchor=>'n');

# report1 - print transcripts
        $frm2->Button(
                -background=>'purple',
                -text=>'Print Transcript',
                -activebackground=>'purple',
                -activeforeground=>'red',
                -command=> \&report1,
                ) ->pack(-side=>"left",-anchor=>'n');

# report2 - print coded transcript lines
        $frm2->Button(
                -background=>'purple',
                -text=>'Print Coded Transcript',
                -activebackground=>'purple',
                -activeforeground=>'red',
                -command=> \&report2,
                ) ->pack(-side=>"left",-anchor=>'n');

# activate coding selection
        $frm2->Button(
                -background=>'khaki1',
                -text=>'Select CODE',
                -activebackground=>'khaki1',
                -activeforeground=>'red',
                -command=> \&code_get,
                ) ->pack(-side=>"left",-anchor=>'n');

# get comments / populate comments database
        $frm2->Button(
                -background=>'Yellow',
                -text=>'Grab comments',
                -activebackground=>'yellow',
                -activeforeground=>'red',
                -command=> \&s_comment,
                ) ->pack(-side=>'left');

# provides a way out of this mess
        $frm2->Button(
                -background=>'Red',
                -text=>'Quit',
                -activebackground=>'red',
                -activeforeground=>'white',
                -command=>sub{exit},
                ) ->pack(-side=>"right",-anchor=>'n');

# place result of selection in text box
# $tb1 = $mw->Text(
#       -spacing1=>2,
#       -spacing3=>2,
#       -background=>'lightblue',
#       -wrap => 'word',
#       )
#       ->pack(-side=>'left',-expand=>1,-fill=>'both');
$tb1 = $mw->Scrolled('Text',
        -scrollbars=>'e',
        -spacing1=>2,
        -spacing3=>2,
        -background=>'lightblue',
        -wrap => 'word',
        )
        ->pack(-side=>'left',-expand=>1,-fill=>'both');
# $tb1->insert('end',$lineout);
```

```
# right side comments / coding box
$frm1 = $mw->Frame(-relief=>"raised",-background=>'Yellow')
        ->pack(-side=>'right',-expand=>1,-fill=>'both',-anchor=>'n');
$frm1_wide = $frm1->cget(-width);
# $tb2 = $mw->Entry(-textvariable=>\$coding_comments,-background=>'Yellow')
                ->pack(-side=>'right',-expand=>1,-fill=>'both');
        $tb2a_msg = "Left click on line number upon which you desire to
                comment.\n";
        $tb2a_msg = "$tb2a_msg Center click to load line number and
                        interview id into comment window.\n";
        $tb2a_msg = "$tb2a_msg You may either select CODE and click
                        'Select CODE' or enter code number.\n";
        $tb2a_msg = "$tb2a_msg To enter a new CODE, simply enter CODE number
                         and COMMENTS in the yellow area.";
        $tb2a = $frm1->Label(
                -background=>'RoyalBlue',
                -foreground=>'white',
                -text=>"$tb2a_msg",
                -width=>"$frm1_wide",
                ) ->pack(-side=>'top',-expand=>1,-fill=>'both',-anchor=>'n');
        $tb2b3 = $frm1->Label(
                -background=>'orange',
                -text=>'Interview Comment Database Population',
                ) ->pack(-side=>'top',-anchor=>'n');
        $frm1b = $frm1->Frame(
                        -background=>'Yellow',
                        -height=>'.25i',
                        ) ->pack(-side=>'top',-anchor=>'nw');
                $tb2b1 = $frm1b->Label(
                        -background=>'Yellow',
                        -text=>'Interview name:   ',
                        ) ->pack(-side=>'left',-anchor=>'nw');
                $tb2b2 = $frm1b->Label(
                        -background=>'Yellow',
                        -textvariable=>\$int_name,
                        ) ->pack(-side=>'right',-anchor=>'nw');
        $frm1c = $frm1->Frame(
                        -background=>'Yellow',
                        -height=>'.25i',
                        ) ->pack(-side=>'top',-anchor=>'nw');
                $tb2c1 = $frm1c->Label(
                        -background=>'Yellow',
                        -text=>'Interview number: ',
                        ) ->pack(-side=>'left',-anchor=>'nw');
                $tb2c2 = $frm1c->Label(
                        -background=>'Yellow',
                        -textvariable=>\$int_id,
                        ) ->pack(-side=>'right',-anchor=>'nw');
        $frm1a = $frm1->Frame(-background=>'Yellow')
                        ->pack(-side=>'top',-anchor=>'nw');
                $tb2a1 = $frm1a->Label(
                        -background=>'Yellow',
                        -text=>'Line number:        ',
                        ) ->pack(-side=>'left',-anchor=>'nw');
                $tb2a2 = $frm1a->Entry(
                        -background=>'PaleGreen',
                        -textvariable=>\$save_int_line,
                        -width=>5,
                        ) ->pack(-side=>'right',-anchor=>'nw');
        $frm1f = $frm1->Frame(-background=>'Yellow')
                        ->pack(-side=>'top',-anchor=>'nw');
                $tb2f1 = $frm1f->Label(
                        -background=>'Yellow',
                        -text=>'Code ID:              ',
                        ) ->pack(-side=>'left',-anchor=>'nw');
                $tb2f2 = $frm1f->Entry(
                        -textvariable=>\$save_code,
                        -background=>'PaleGreen',
                        -width=>5,
                        ) ->pack(-side=>'left',-anchor=>'nw');
        $frm1e = $frm1->Frame(-background=>'Yellow')
                        ->pack(-side=>'top',-anchor=>'nw');
```

```
                $tb2e1 = $frm1e->Label(
                        -background=>'Yellow',
                        -text=>'Comments:           ',
                        ) ->pack(-side=>'left',-anchor=>'nw');
                $tb2e2 = $frm1e->Entry(
                        -textvariable=>\$save_comments,
                        -background=>'PaleGreen',
                        -width=>45,
                        ) ->pack(-side=>'left',-anchor=>'nw');

# Existing comments frame
        $frm4a = $frm1->Text(
                -spacing1=>2,
                -spacing3=>2,
                -background=>'cyan',
                -wrap => 'word',
                ) ->pack(-side=>'bottom',-expand=>1,-fill=>'both');
        comm_pop();

# Coding frame
        $frm3a = $frm1->Scrolled('Listbox',
                -background=>'khaki1',
                -scrollbars=>'se',
                -selectmode=>'single',
                ) ->pack(-side=>'bottom',-expand=>1,-fill=>'both',
                        -anchor=>'n');
        code_pop();

MainLoop;

sub int_list {
        # determine who the interview subjects are
        $dbh = DBI->connect('DBI:mysql:interviews','cfiaime',"$passwd");
        $dbs = $dbh->prepare("select distinct int_id,int_name from
                         transcript order by int_id");
        $dbs->execute;

        # determine which interviews have been analyzed
        $dbs5 = $dbh->prepare("select distinct coding.int_id,int_name
                         from coding,transcript
                where coding.int_id=transcript.int_id order by int_id");
        $dbs5->execute;
        $analyzed_cnt = $dbs5->rows;

        my $j = 0;
        while (my @row5 = $dbs5->fetchrow_array) {
                $analyzed[$j] = $row5[0];
                $j++;
        }

        $rowcnt = $dbs->rows;
        $i = 0;
        while (@row = $dbs->fetchrow_array) {
                $trans_id[$i] = $row[0];
                $trans_name[$i] = $row[1];
                for ($j=0; $j<$analyzed_cnt; $j++) {
                        if ($trans_id[$i] == $analyzed[$j]) {
                                $trans[$i] = "$trans_id[$i]  $trans_name[$i]
                                        (analyzed)";
                                $j=99;
                        } else {
                                $trans[$i] = "$trans_id[$i]  $trans_name[$i]";
                        }
                }
                $i++;
        }
}

sub lbsearch {
        $tag_indent = $tb1->tagConfigure('indent',
                -lmargin1=>55,
                -lmargin2=>112,
```

```
                );
        $tag_lineno = $tb1->tagConfigure('line_no',
                -foreground=>'navy',
                -background=>'white',
                );
        $tb1->tagBind('line_no','<Button-2>', \&lbcomment);
        my @selected = $lb->curselection;
        foreach (@selected) {
                $int_id = $_ + 1;
                $dbs1 = $dbh->prepare("select line_no,speaker,response
                        from transcript
                        where int_id='$int_id' order by line_no");
                $dbs1->execute;
                $int_name = $trans_name[$int_id-1];
                $tb1->delete('1.0','end');
                while (@row1 = $dbs1->fetchrow_array) {
                        $tb1->insert('end',"$row1[0]",'line_no');
                        $tb1->insert('end',"\t$row1[1]\t");
                        $tb1->insert('end',"$row1[2]",'indent');
                }
        }
        comm_pop();
        int_list();
}


sub lbcomment {
        $tb1->selectWord;
        $lineno = $tb1->getSelected();
        $tb1->unselectAll;
        $tb2a2->insert('end',"$lineno");
}


sub s_comment {
        if (not $save_comments) {
                $save_comments = ' ';
        }
        print "Int name: $int_name\tInt ID: $int_id\tInt line:
                $save_int_line\tSave code:
                $save_code\tSave comments: $save_comments\n";
        if ($code_id[$save_code - 1]) {
                $dbs3 = $dbh->prepare("insert into coding set
                        int_id=$int_id,
                        code_id=$save_code,
                        line_no=$save_int_line,
                        comments='$save_comments'");
                $dbs3->execute;
                comm_pop();
        } else {
                $dbs3 = $dbh->prepare("insert into codes set
                        code_desc='$save_comments'");
                $dbs3->execute;
                code_pop();
        }
        $save_int_line = '';
        $save_comments = '';
        $save_code     = '';
        $tb2a2->delete(0,'end');
        $tb2e2->delete(0,'end');
        $tb2f2->delete(0,'end');
}

sub code_get {
        my @code_sel = $frm3a->curselection;
        foreach (@code_sel) {
                my $code_get = $_ + 1;
                $tb2f2->insert('end',"$code_get");
        }
}

sub code_pop {
```

```
# populate coding window
        $frm3a->delete(0,'end');
        $dbs2 = $dbh->prepare("select * from codes order by code_id");
        $dbs2->execute;
        $i = 0;
        while (@row = $dbs2->fetchrow_array) {
                $code_id[$i] = $row[0];
                $code_desc[$i] = $row[1];
                $code[$i] = "$code_id[$i]     $code_desc[$i]";
                $i++;
        }
        $frm3a->insert('end',@code);
}

sub comm_pop {
# populate comments window
        $frm4a->delete('1.0','end');
        $dbs4 = $dbh->prepare("select * from coding
                where int_id=$int_id
                order by line_no,code_id");
        $dbs4->execute;
        $i = 0;
        while (my @row1 = $dbs4->fetchrow_array) {
                $comm_id[$i] = $row1[1];
                $comm_line[$i] = $row1[2];
                $comm_desc[$i] = $row1[3];
                $comm[$i] = "Line:  $comm_line[$i]\tCode:
                        $comm_id[$i]\t$comm_desc[$i]\n";
                $i++;
        }
        foreach (@comm) {
#               $frm4a->insert('1.0',$_);
                $frm4a->insert('end',$_);
        }
}

sub report1 {
        print "This will evenutally print the transcripts\n";
}


sub report2 {
        print "This will evenutally print the coded transcripts\n";
}
```

CODETEST.PL

```
#!/usr/bin/perl -w

# codetest.pl
# jeff williams
# 05/17/2004
# transcript printing test

# describe codes;
# +-----------+--------------+------+-----+---------+
# | Field     | Type         | Null | Key | Default |
# +-----------+--------------+------+-----+---------+
# | code_id   | int(4)       |      | PRI | NULL    |
# | code_desc | varchar(255) | YES  |     | NULL    |
# +-----------+--------------+------+-----+---------+
#
# describe coding;
# +----------+--------+------+-----+---------+
# | Field    | Type   | Null | Key | Default |
# +----------+--------+------+-----+---------+
# | int_id   | int(4) | YES  |     | NULL    |
# | code_id  | int(4) | YES  |     | NULL    |
# | line_no  | int(4) | YES  |     | NULL    |
# | comments | blob   | YES  |     | NULL    |
```

```
# +----------+--------+------+-----+---------+
#
# describe transcript;
# +----------+------------+------+-----+---------+
# | Field    | Type       | Null | Key | Default |
# +----------+------------+------+-----+---------+
# | int_id   | int(4)     |      |     | 0       |
# | int_name | varchar(30) | YES  |     | NULL    |
# | speaker  | char(1)    | YES  |     | NULL    |
# | response | blob       | YES  |     | NULL    |
# | line_no  | int(4)     | YES  |     | NULL    |
# +----------+------------+------+-----+---------+

system('clear');
print "Enter database password: ";
$dbf_pass = <STDIN>;
chomp($dbf_pass);

use vars qw( @codes @code_desc );

use DBI;

$dbh = DBI->connect('DBI:mysql:interviews','cfiaime',"$dbf_pass");
$dbs=$dbh->prepare("select * from codes order by code_id");
$dbs->execute;

my $code_cnt = 0;
while (@row = $dbs->fetchrow_array) {
        $code_id=$row[0];
        $code_desc=$row[1];
        $dbs1 = $dbh->prepare("select coding.int_id, coding.code_id, coding.line_no,
                coding.comments, transcript.speaker, transcript.response
                from coding, transcript where
                coding.int_id=transcript.int_id and coding.line_no=transcript.line_no
                and coding.code_id='$code_id' order by coding.int_id");
        $dbs1->execute;

        $codeout = "codeout.$code_id.txt";

        $- = 0;
        $% = 0;

        open(CODEOUT,">$codeout");

        select CODEOUT;

        while (@row1 = $dbs1->fetchrow_array) {
                $int_id = $row1[0];
                $code_id = $row1[1];
                $line_no = $row1[2];
                $comments = $row1[3];
                $speaker = $row1[4];
                $response= $row1[5];

                write(CODEOUT);
        }
        close(CODEOUT);
#       system("lpr $codeout");
}


format CODEOUT_TOP =
     Page @<<  Code: @<  Desc: @<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
$%, $code_id, $code_desc

     Int  Line  Speaker  Response
     ---  ----  -------  ---------------------------------------------------


format CODEOUT =
     @>>  @>>>    @      ^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
$int_id, $line_no, $speaker, $response
```

```
~~                       ^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
$response
          Comment        ^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
$comments
~~                       ^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
$comments
```

TRANSTEST.PL

```
#!/usr/bin/perl -w

# transtest.pl
# jeff williams
# 5/17/2004
# transcript printing test

# describe codes;
# +-----------+--------------+------+-----+---------+
# | Field     | Type         | Null | Key | Default |
# +-----------+--------------+------+-----+---------+
# | code_id   | int(4)       |      | PRI | NULL    |
# | code_desc | varchar(255) | YES  |     | NULL    |
# +-----------+--------------+------+-----+---------+
#
# describe coding;
# +----------+--------+------+-----+---------+
# | Field    | Type   | Null | Key | Default |
# +----------+--------+------+-----+---------+
# | int_id   | int(4) | YES  |     | NULL    |
# | code_id  | int(4) | YES  |     | NULL    |
# | line_no  | int(4) | YES  |     | NULL    |
# | comments | blob   | YES  |     | NULL    |
# +----------+--------+------+-----+---------+
#
# describe transcript;
# +----------+-------------+------+-----+---------+
# | Field    | Type        | Null | Key | Default |
# +----------+-------------+------+-----+---------+
# | int_id   | int(4)      |      |     | 0       |
# | int_name | varchar(30) | YES  |     | NULL    |
# | speaker  | char(1)     | YES  |     | NULL    |
# | response | blob        | YES  |     | NULL    |
# | line_no  | int(4)      | YES  |     | NULL    |
# +----------+-------------+------+-----+---------+

use DBI;

system(’clear’);
print "Enter database password: ";
$dbf_pass = <STDIN>;
chomp($dbf_pass);

$dbh = DBI->connect(’DBI:mysql:interviews’,’cfiaime’,"$dbf_pass");
$dbs=$dbh->prepare("select * from transcript order by int_id,line_no");
$dbs->execute;

select TRANSOUT;

my $old_id = -1;
while (@row = $dbs->fetchrow_array) {
        $int_id = $row[0];
        $int_name=$row[1];
        $speaker=$row[2];
        $response=$row[3];
        $line_no=$row[4];

        if ($int_id != $old_id) {
                $transout = "transout.$int_id.txt";
                open(TRANSOUT,">$transout");
                $- = 0;
```

```
                $% = 0;
                $old_id = $int_id;
                select STDOUT;
                select TRANSOUT;
        }

        write(TRANSOUT);
}


format TRANSOUT_TOP =
     Interview @<    Interview of: @<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<  Page @<<
$int_id, $int_name, $%

     Line  Speaker  Response
     ----  -------  --------------------------------------------------------


format TRANSOUT =
     @>>>    @      ^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
$line_no, $speaker, $response
~~                  ^<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
$response
```