COMPSOFT – A PLATFORM FOR ONLINE PROBLEM-BASED LEARNING FOR HEALTH CARE PROFESSIONALS

by

Chun Wai Chan
B.Sc., University of British Columbia, 1999

PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

In the
School
of
Interactive Arts and Technology

SIMON FRASER UNIVERSITY

Summer 2006

APPROVAL

Name: **Chun Wai Chan**

Degree: **Master of Applied Science**

Title of Project: **COMPSOFT – A Platform for Online Problem-Based Learning for Health Care Professionals**

Examining Committee:

Chair: **Dr. Tom Calvert**

____

Dr. Vive Kumar
Senior Supervisor

____

Steve DiPaola
Supervisor

____

Dr. David Kaufman
Supervisor

____

Dr. Jane Fee
External Examiner

Date Defended/Approved: May 29/06

ABSTRACT

PBL has been widely adopted by medical schools for years. However, implementing PBL curricula in online environments pose specific challenges to educators. Although existing technologies can alleviate some of the challenges, none of them comprehensively address the key aspects of a PBL curriculum. COMPSOFT, a software platform, has been designed to support key aspects of online PBL. COMPSOFT consists of a number of software components. Each of them supports or enhances a specific aspect of PBL. They work cooperatively, complementing each other to provide a rather complete online PBL environment. This project details the rationale, design and implementation of such a software environment. A usability test was performed where several students worked through a medical PBL case, online, under the guidance of a facilitator using COMPSOFT. The usability test results, addressing what is working well, what is not working well, and how to improve the PBL environment, have been encouraging and the participants rated the platform positively.

ACKNOWLEDGEMENTS

I would like to acknowledge the supervision of Dr. Vive Kumar as my senior supervisor. I thank Steve DiPaola for his guidance on the technical side and his help on my paper publication. I dedicate my sincere thanks to Dr. David Kaufman for his general advice on this research, his help on the experiment and on various administrative works.

I am grateful to everyone in the SAGE-SIM team, every staff in LIDC in Simon Fraser University, and every staffs in Simon Fraser University Surrey for their supports and helps. From the team member, Robyn Schell is specially thanked for her care and attention.

Finally, I am forever indebted to my parents and Waisheep for their understanding, endless patience and encouragement when it was most required.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Medical education has undergone tremendous changes this decade. From the traditional biomedical perspective, medical education has been broadened to a more holistic approach that emphasizes all aspects of medical learning experiences including the physical, environmental, emotional, social, and lifestyle values. Problem-based learning (PBL) has been widely adopted by medical schools as a powerful education methodology that addresses many of the needs associated with modern medical curriculum. Although PBL is the preferred methodology for medical education, it poses a large burden to schools in terms of the resources needed to implement a PBL curriculum. Educators and students also face problems in their roles and responsibilities, and in assessments.

Technologies have been used in PBL to resolve some of the challenges faced by educators. However, none of these existing technologies covers all aspects of PBL. One solution to the problem is to deploy several distinct systems at the same time but this creates new problems in terms of system interoperability and user experience.

The Collaborative Online Multimedia Problem-based Simulation Software (COMPSOFT) platform (Arya, DiPaola, Jefferies, & Enns, 2006; DiPaola, Chan, & Arya, 2005; , "Simulated Gaming for Medical Learning", 2005, June), formerly called COMPS, is the solution to the problems mentioned above. COMPSOFT is developed under the Collaborative Online Multimedia Problem-based Simulation (COMPS) project by the Simulation and Advanced Gaming Environments (SAGE) for Learning (*SAGE:*

Simulation and Advanced Gaming Environments for Learning, n.d.) initiative. SAGE for Learning is a Pan-Canadian initiative funded by the Social Sciences and Humanities Research Council (SSHRC) to explore the potential of simulations and games to support learning in the health domain. The SAGE-SIM team, one of the teams in the SAGE initiative, is responsible for the design and development of COMPSOFT. The SAGE-SIM team includes people from Simon Fraser University (SFU) and the British Columbia Institute of Technology (BCIT).

The next chapter explains PBL in detail: the process, the rationale, the advantages, and the challenges faced by educators implementing PBL curricula. Chapter 3 outlines the kinds of technologies that are available and how they are used to enhance and resolve some of the challenges of PBL. The chapter also discusses the reasons for the inadequacies of the existing technologies. COMPSOFT, the software platform, is introduced in Chapter 4. The chapter examines the platform's functionalities, its user interfaces, and how it supports and enhances PBL. Chapter 5 presents COMPSOFT's architecture, implementation details, and deployment issues. Chapter 6 describes a study that assesses the usability of the COMPSOFT platform, followed by an analysis of the results obtained from the study. Finally, Chapter 7 summarizes this research.

2 PROBLEM-BASED LEARNING (PBL)

2.1 Definitions

There are various definitions of PBL in the literature (Barrows & Tamblyn, 1980; Boud, 1985; Boud & Feletti, 1998; Duch, Groh, & Allen, 2001; A. Kaufman, 1985). A popular and straightforward definition for PBL is "the learning that results from the process of working toward the understanding or resolution of a problem. The problem is encountered first in the learning process" (Barrows & Tamblyn, p. 18).

Another popular definition was proposed by Duch (2001) as an instructional method that challenges students to learn to learn, to work cooperatively in groups to come up with solutions to real world problems. These problems engage students' curiosity to initiate the learning of the subject matter, and thereby act as a context for students to enhance their critical thinking and problem solving skills. In addition to learning the essential concepts of the course, students acquire the ability to locate and use appropriate learning resources for life-long learning.

Jones (1996) of San Diego State University's Distributed Course Delivery for Problem Based Learning Institute described PBL as student-centred, faculty-facilitated and that:

> In problem based learning, students collaborate to study the issues of a problem as they strive to create viable solutions. Unlike traditional instruction, which is often conducted in lecture format, teaching in problem based learning normally occurs within small discussion groups of students facilitated by a faculty tutor.

> Because the amount of direct instruction is reduced in problem based learning, students assume greater responsibility for their own learning. The instructor's role becomes one of subject matter expert, resource guide, and task group consultant. (para. 1)

2.2 History and Origins

Problem-based learning is a relatively recent innovation in higher education. However, its root can be traced back to ancient days (Rhem, 1998). The president of Samford University, Thomas Corts, viewed PBL as a newly recovered style of learning. He related PBL to Socrates (469–339 B.C.) who believed that self-education or self-discovery was the only true way to learn. In 1909, Dewey said (1909, chap. 16):

> If, nevertheless, the situation as a whole presents something novel and hence uncertain, the entire response is not mechanical, because this mechanical operation is put to use in solving a problem. There is no end to this spiral process: foreign subject matter transformed through thinking into a familiar possession becomes a resource for judging and assimilating additional foreign subject-matter.(p. 223)

As John Cavanaugh pointed out, people had known about PBL but had not brought it into practice. Because of advances in cognitive science and in technology, they were then able to fill in the details of the abstract framework built by Dewey (Rhem, 1998).

In the 1950s, Case Western Reserve University in Ohio is said to be the first to incorporate PBL methods and strategies in a multidisciplinary laboratory. However, the Faculty of Health Sciences at the McMaster University in Ontario, Canada is considered to be the first to offer a complete PBL curriculum (Neufeld, Woodward, & MacLeod, 1989). From then on, numerous medical schools have incorporated PBL into their curricula either completely or partially (Armstrong, 1998; J. Des Marchais, 1990;

Edwards, 1990; Kantrowicz, Kaufman, Mennin, Fulop, & Guilbert, 1987; A. Kaufman et al., 1989; Mann & Kaufman, 1995; H. G. Schmidt, Dauphinee, & Patel, 1987).

From medical and professional schools, PBL is slowly adopted by schools of science in general and later by schools of humanities (Boud, 1985; Boud & Feletti, 1998). Recently, PBL has been experimented in elementary schools, middle schools and high schools and are reported to induce positive changes in the learning process (Finkle & Torp, 1996).

2.3 Process

There are many variations of the PBL process. A typical process may be summarized as follows (Albanese & Mitchell, 1993; Barrows, 1986; Barrows & Tamblyn, 1980; Boud & Feletti, 1998; Finkle & Torp, 1995):

1. Present the problem statement. An ill-structured problem or scenario, in the form of a case, a research paper, or a video tape, for example, is presented to the students before any prior studies and preparation have been made, as it would be in reality. The problem will be designed in such a way that the students do not have enough knowledge to solve the problem. This requires that the students gather information or learn new concepts, principles or skills as they try to solve the problem. Students (in groups) then attempt to define the broad nature of the problem by organizing their ideas and previous knowledge related to the problem.
2. Identify the learning issues. Throughout discussion, areas of learning requiring attention and aspects of the problem that the students do not understand are identified. These learning issues are recorded by the group. Students are

continually encouraged to define what they know - and more importantly - what they do not know.

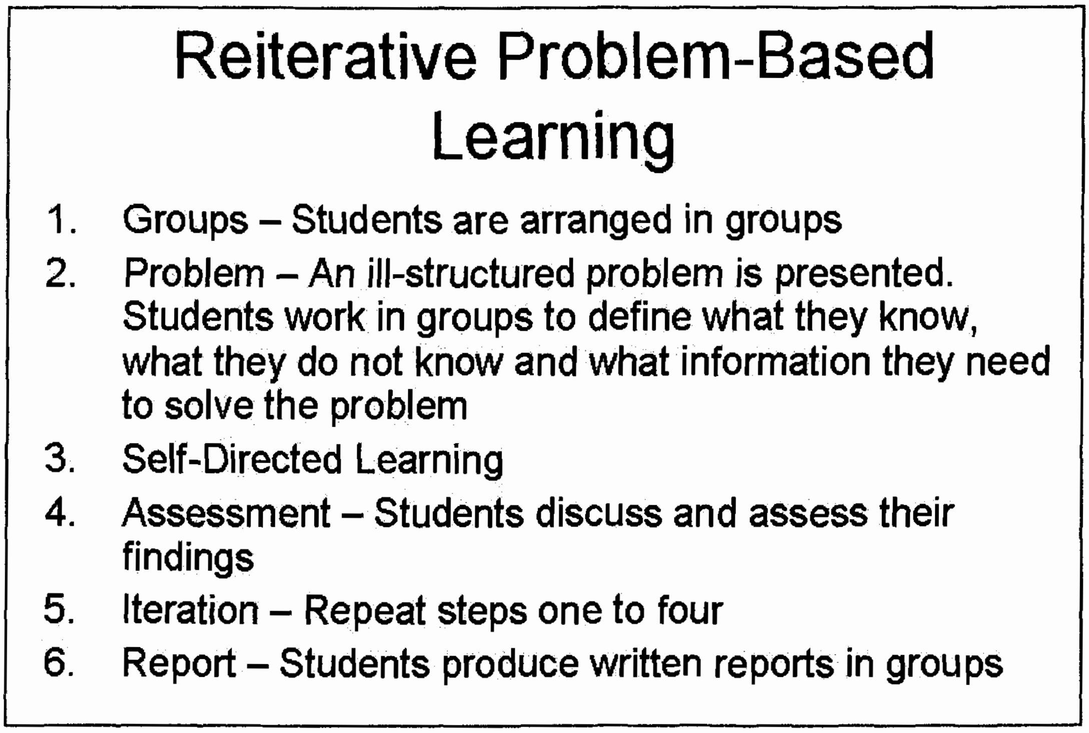
3. Rank the learning issues. Students rank the learning issues generated in the session in order of importance. They decide which questions will be followed up by the whole group, and which issues can be assigned to individuals, who later teach the rest of the group. Students and instructor also discuss what resources will be needed in order to research the learning issues, and where they could be found.

4. Apply the new knowledge. When students reconvene, they present, orally and/or in writing, their findings and recommendations about the previous learning issues. The skills and knowledge acquired through individualized studies are applied to the problem, evaluated, and reinforced through discussion. New learning issues are identified as they progress through the problem. This step contributes to the term reiterative or closed-loop in reiterative PBL (Barrows, 1986).

5. Learn. Finally, the learning that occurred through working on the problem and in individualized studies is summarized and becomes a part of the students' existing knowledge and skills.

An iterative PBL process is shown in Figure 2-1 (Barrows, 1985).

Figure 2-1: Reiterative Problem-Based Learning



Reiterative Problem-Based Learning

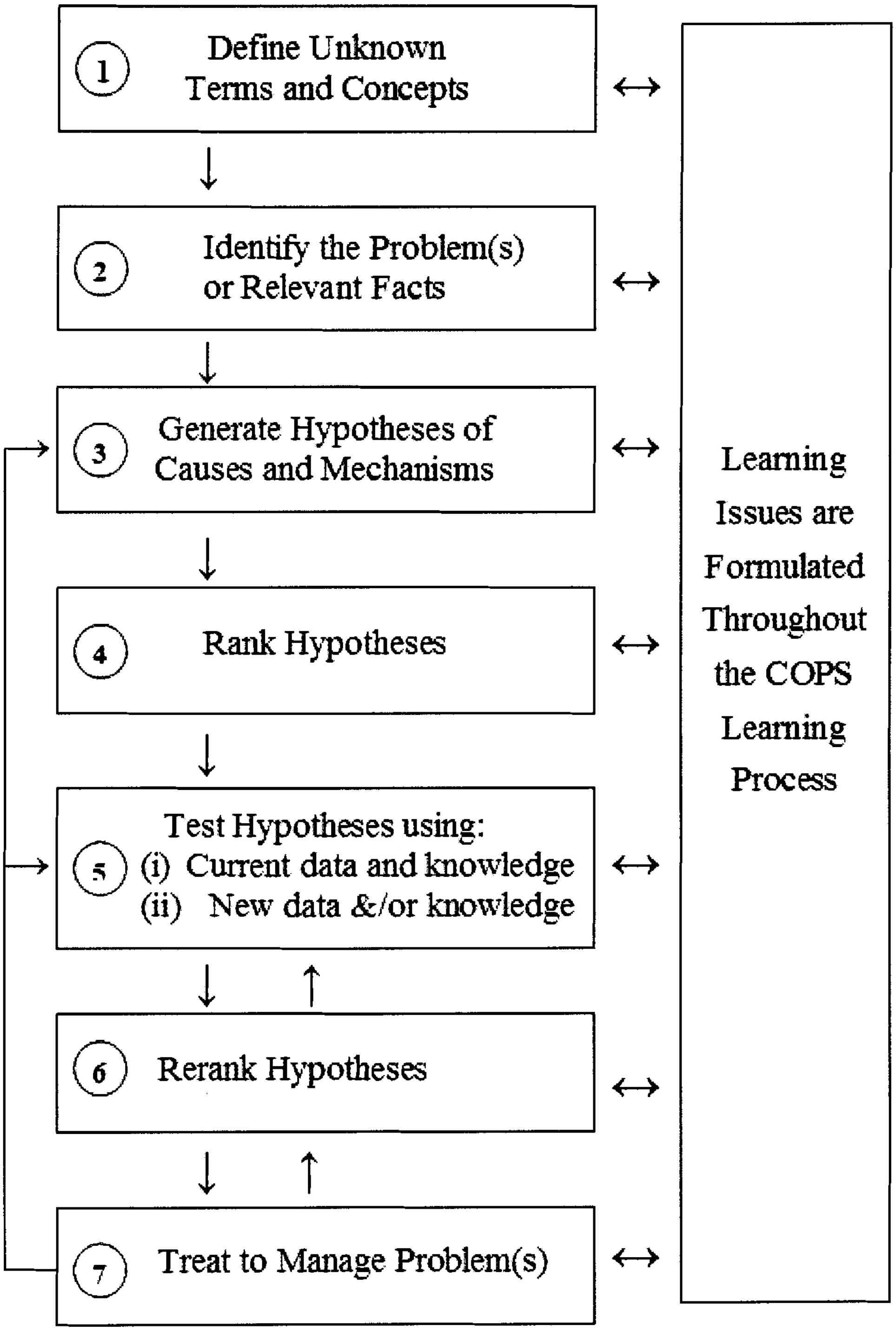
1. Groups – Students are arranged in groups
2. Problem – An ill-structured problem is presented. Students work in groups to define what they know, what they do not know and what information they need to solve the problem
3. Self-Directed Learning
4. Assessment – Students discuss and assess their findings
5. Iteration – Repeat steps one to four
6. Report – Students produce written reports in groups

2.4 Facilitator Roles and Functions

In PBL, the role of a tutor is different from its counterpart in traditional teaching where the instructor's duty is to dispense information to students. In a student-centred PBL setting, a tutor works as a facilitator who must guide, probe and support students' initiatives, not lecture, direct or provide easy solutions. PBL literature has suggested the multiple roles a facilitator must play (Barrows, 1994; Grand'Maison & Des Marchais, 1991; Wilkerson & Feletti, 1991) and identified them as *parent*, *professional*, *consultant*, *confidant*, *learner*, and *mediator* (Wilkerson & Feletti). The tasks of a facilitator can be categorized as follows (Barrows, 1994):

1. Navigate. The facilitator is expected to guide the group through each step of the instructional process. The tutor should ensure that each step in the process is covered thoroughly. A sample process is shown in Figure 2-2 (D. Kaufman, 1995).

Figure 2-2: Outline of the Case-Oriented Problem-Stimulated Learning Process.


1 Define Unknown Terms and Concepts
2 Identify the Problem(s) or Relevant Facts
3 Generate Hypotheses of Causes and Mechanisms
4 Rank Hypotheses
Test Hypotheses using:
5 (i) Current data and knowledge
(ii) New data &/or knowledge
6 Rerank Hypotheses
7 Treat to Manage Problem(s)
Learning Issues are Formulated Throughout the COPS Learning Process

2. Question. The facilitator should question the students at various stages to check their understanding, to clarify points, to lead students to important areas missed by them, to raise controversial questions, and to increase the depth of discussions.
3. Facilitate. The facilitator should perform group management activities such as directing questions from students to facilitator and then back to students, and solicit different points of views from group members, in order to create a positive and constructive group process.
4. Diagnose. The facilitator should monitor the progress of the students and should offer help to those having difficulties before it is too late.

The facilitator is expected to participate more frequently at the start of the learning process. Once the students are familiar with the process, the facilitator will fade out and let the students guide their own learning. In some situations, a PBL session may be run without the help of a facilitator (DeGrave, Volder, Gijselaers, & Damoiseaux, 1990).

2.5 Rationale

In traditional education, students study concepts of general sciences in early years and apply them to clinical settings and diagnosis in latter years. This educational approach has been criticized for a number of reasons (J. E. Des Marchais, Bureau, Dumais, & Pigeon, 1992; Lowry, 1992; H. G. Schmidt, 1983):

1. Basic science and clinical science are unnaturally separated.
2. It can be difficult for students to apply the acquired knowledge.
3. Time is wasted in learning things that are irrelevant or later forgotten.

4. Students are bored and demoralized to study and memorize knowledge that has no apparent relevance.
5. Students resist further learning.

Theoretically, the objectives of PBL address the problems of traditional education. According to Barrows (1983), PBL allows students to better retain the knowledge acquired and apply them in a clinical context. Also, PBL is motivating to the students and helps develop their self-directed learning skills.

The literature has provided reasons in support of PBL. The five categories of rationale described by Kaufman (1995) can be reorganized into four categories: philosophical, educational psychological, professional practice, and political.

2.5.1 Philosophical Rationale in Support of PBL

According to Savery and Duffy (Savery & Duffy, 1995; Shor, 1992), constructivist principles state that understanding is constructed from applying previous knowledge to the environment. When cognitive conflict arises, learning is stimulated. When we negotiate our differences in our understanding with others, we refine and evaluate our knowledge. These are some of the typical activities performed by PBL students.

The student-centred approach of PBL empowers students so that the facilitator becomes a co-learner rather than an authority figure. This is described by the concept of empowering education (Shor, 1992). It not only develops students' responsibilities in self-learning, but also encourages students to help in the learning of other group members.

In PBL, students deal with problems in a multidisciplinary fashion just as they deal with problems in real-world situations. Therefore, the barrier between theoretical and applied knowledge is eliminated.

2.5.2 Educational Psychological Rationale in Support of PBL

Learning theory and cognitive psychology provide ample support for PBL (Chickering & Gamson, 1991; Coles, 1990; Finkle & Torp, 1995; Knowles, 1988; Norman & Schmidt, 1992; Ryan, 1992; H. G. Schmidt, 1983). Schmidt provided support for PBL based on three learning theory principles:

1. Activation of prior knowledge: When students encounter a new problem in PBL, they apply their existing knowledge to try to deal with the problem.
2. Encoding specificity: The more relevant the situation is where learning has occurred (during the PBL sessions) and the more the situation knowledge acquired is applied (in real-world scenarios), the more the opportunity will be for critical thinking and meta-cognitive growth and the more likely that transfer and recall of knowledge will occur. This is known as cognitive dissonance (Finkle & Torp, 1995) or referred to as contextual learning theory (Coles, 1990).
3. Elaboration of knowledge: PBL activities, such as discussion, reporting self-study findings, questioning, and reviewing, allow students to elaborate on the information acquired. This allows students to better understand and retain the knowledge acquired.

PBL provides four motivating factors as described by motivation theory. First, a problem resembling their future practice arouses learners' curiosity and interest. Norman and Schmidt (1992) reported that students rate the PBL environment more stimulating than conventional ones. Second, since the instruction closely resembles students' future practice, "the significance of what has been, or is to be learnt is clearly demonstrated" (Ryan, 1992, para. 8). Third, success is attainable and controllable by the students through appropriate self-learning and group discussion. Finally, the learning activities and rewards lead learners to intrinsic satisfaction.

The group-oriented, student-centred, tutor-facilitated approach in PBL provides an environment that matches the good conditions for learning as described by educational experts (Chickering & Gamson, 1991; Knowles, 1988). Knowles, considered as the father of adult learning theory, listed the conditions for efficient adult learning: an environment that provides physical comfort, trust, respect, helpfulness, freedom of expression, acceptance of differences, where goals of the learning experience are viewed as individual goals, where students feel the responsibility and commit to plan and operate the learning experience, where learners actively engage, and feel the progress to their own goals. Chickering and Gamson proposed the Seven Principles for Good Practice in Undergraduate Education that includes: 1) encouraging student-faculty contact, 2) promoting cooperation among students, 3) promoting active learning, 4) giving prompt feedback, 5) emphasizing time on task, 6) communicating high expectations, and 7) respecting diverse talents and ways of learning.

2.5.3 Professional Practice Rationale in Support of PBL

The amount of information a professional needs to learn for professional practice has been increasing exponentially. It is becoming impossible for future professionals to acquire all the knowledge and skills necessary for their professional lives from any school or teacher (Barrows, 1985). Therefore, it is necessary for future professionals to nurture their lifelong, self-directed learning skills in order to satisfy their professional needs. In certain professions, such as in the medical field, professional practices are changing rapidly. Therefore, the knowledge and skill acquired from a curriculum may become obsolete shortly after the student graduates (Yura & Torres, 1986). Rush, Ouellet, and Wasson (1991) stated that shuffling the substance of the curriculum alone was not sufficient to catch up with societal trends. Routine procedures or pattern recognition are no longer adequate, and problem-solving skill is required to prepare the professionals for the unidentified future requirements of their professional lives (de Tornyay, 1990; Jackling, Lewis, Brandt, & Snell, 1990). Since many real-world problems are multi-faceted, professionals need to work in teams with people from different professional backgrounds and personalities. Therefore, communication skills and familiarity with working in a team environment are skills that professionals need to possess for their career.

PBL can be incorporated to address each of the above requirements by fostering the development of self-directed learning skills, encouraging the development of critical reasoning and analytical skills, and promoting communication skills in a group environment (Barrows, 1985; Engel, 1998).

2.5.4 Political Rationale in Support of PBL

Several reports for reforming higher education have been published by national associations, and the purposes and priorities of university education have been a hot discussion topic among universities. For medical education, the American Association of Medical Colleges has published several influential reports (Muller, 1984; Swanson &

Anderson, 1993). In one such report, Swanson and Anderson suggested reforming strategies and recommended PBL to be a desirable educational methodology. For undergraduate education, government agencies, and others have called for a reform to the conventional content-laden lecture format of science courses (Czujko, 1994; Project Kaleidoscope, 1991; Tobias, 1992, , 1994; Wingspread Conference, 1994). PBL addresses many of the concerns.

Another possible reason for the widespread acceptance of PBL is that in the first few schools that incorporated PBL, the faculty and students considered PBL to be very successful. Because the settings in the schools were so different and diverse, other schools perceived PBL as a universally applicable mode of instruction. The communication between early developers and later adopters might be a contributing factor to the success of PBL in those schools. After the spread of successful stories from several medical schools, PBL became well known and considered less risky than other innovative educational methodologies.

2.6 Effectiveness

Incorporating PBL in the curriculum does not only change the way the faculty teach, but also brings about changes in all areas in the educational process. Each of these changes may affect the outcome of an experimental research on the effectiveness of PBL. For example, Dean, Barratt, Hendry and Lyon (2003) mentioned that students' selection criteria might affect experimental outcome. Moreover, early positive findings from enthusiasts of PBL and the generalization of experimental findings from small-scaled, single-disciplined studies make these findings more misleading. All these make it hard to determine the effectiveness of PBL (Finucane, Johnson, & Prideaux, 1998).

It is easy to find studies with contradicting or inconclusive results on PBL. In a three-year experiment comparing PBL with traditional curriculum, Distlehorst and Robbs (1998) concluded that students in the PBL curriculum had better problem-solving skills in the first three classes. However, Colliver (2000) commented that Distlehorst and Robbs's findings were flawed because they failed to account for the effect of the special selection process for students entering the PBL program, and that the differences between the two groups' problem-solving skills were small. In a survey to measure the preparedness of students for hospital practice performed by the University of Sydney and New South Wales hospitals, comments for students' basic science knowledge were conflicting (Dean, Barratt, Hendry, & Lyon, 2003). In another study, although Schmidt and van der Molen (2001)concluded that medical students in PBL curriculum might be more competent compared to their traditional counterparts, they acknowledged the possible influences of the selection process and students' emotional commitment to their school to the experimental outcome.

2.6.1 Advantages

Many earlier studies have reported positive findings for PBL (A. Kaufman et al., 1989; Moore, 1991; H. G. Schmidt, Dauphinee, & Patel, 1987; Woodward, 1990). However, we have to be cautious about these reports because the studies were either performed in small-scale, single-institute settings or performed by early adopters who were enthusiasts of the new approach.

Because of the large number of studies and literature reviews about the effectiveness of PBL done in recent years, we are able to generalize the advantages of PBL. Psychologically, PBL provides a learning environment that is more nurturing, enjoyable, challenging, stimulating, motivating, and humane (Albanese & Mitchell, 1993; Camp, 1996; Colliver, 2000; Norman & Schmidt, 1992; Vernon & Blake, 1993). In addition, students from PBL curricula were rated higher in a broad spectrum of skills and competencies including but not limited to interpersonal skills, holistic care, team working, and humaneness by themselves and by educational supervisors (Dean, Barratt, Hendry, & Lyon, 2003; A. Jones, McArdle, & O'Neill, 2002). Academically, although PBL students usually rate their clinical knowledge lower than those of the traditional students, there is no difference in the ratings given by educational observers (Dean, Barratt, Hendry, & Lyon; A. Jones, McArdle, & O'Neill; Krynock & Robb, 1996; H. G. Schmidt, 1998). In other words, PBL students are as good as traditional students in terms of their understanding of diseases. Norman and Schmidt wrote an influential paper that reviewed how PBL changes students learning from experimental evidences. They concluded that although PBL students' initial learning might be diminished, their retention of knowledge has been improved. PBL students are also found to be better in giving causal explanations when diagnosing and make less diagnostic errors because they are better in integrating basic science knowledge to clinical problems, transferring concepts to new problems, and that they put more emphasis on in-depth understanding of subject matters (Krynock & Robb; Norman & Schmidt; H. G. Schmidt; Vernon & Blake). Lastly, PBL provides a means for students to self-regulate in their own learning and improves their self-directed learning skills (Norman & Schmidt; Pierce & Gerdes, 2006; H. G. Schmidt; Vernon & Blake).

2.6.2 Challenges

Given all the advantages of PBL, schools and educators face many challenges in running a PBL curriculum. These challenges can be roughly divided into several categories: resources, roles and responsibilities, and assessment.

2.6.2.1 Resources

Running a PBL curriculum requires extensive resources (Colliver, 2000). When designing the curriculum, it takes an enormous amount of time in constructing real-life problems that are at an appropriately challenging level for the students (Simons, Smithies, Daniel, Gulati, & Hazel, n.d.). Timetabling can be aggravating especially in medical schools where students and instructors have numerous time demands during their day (McLoughlin, 2005). Clinical training takes up a huge amount of students' time. The training sites may even be far away from the campus. Coupled with the time for the PBL lessons, group meetings outside the classes, and time for self-studies, it becomes difficult for students to squeeze all these activities into their timetables. Instructors face similar problems because they need more time to contact their students in a PBL curriculum. Because of the small-group nature of PBL, plenty of other resources are required such as a larger number of tutors and classrooms, more librarians, library resources, and simulated patients (Fincham & Shuler, 2001; McLoughlin; Savin-Baden, 2000; Simons, Smithies, Daniel, Gulati, & Hazel). This poses a huge financial burden on institutions incorporating PBL in their curricula. In fact, Albanese and Michell (1993) pointed out that the cost would slow the adoption of PBL in medical schools when the class size reached 100.

2.6.2.2 Roles and Responsibilities

Incorporating PBL into a curriculum can be a challenge because the roles and responsibilities of students and instructors are vastly different from traditional education (Ryan, 1992). Instead of acting as an information reserve that transmits knowledge to students through direct lecturing, a teacher now acts as a facilitator who facilitates group discussions, observes students learn by themselves, and gives limited guidance occasionally (*Disadvantages of Problem Based Learning*, 1996). This has raised concern from some tutors in favour of traditional learning questioning if their students in PBL could acquire the same breadth of knowledge as their traditional counterparts do (Vernon & Blake, 1993). Some students even complain that PBL is a way for tutors to outsource their duties to students because they only need to facilitate (H. Schmidt, Henny, & de Vries, 1992).

Instead of memorizing facts and patterns fed by instructors, PBL students have to take initiative to conduct research, work with group members, and report findings to accomplish their unique learning objectives (*Disadvantages of Problem Based Learning*, 1996). Many students from traditional curricula are not accustomed to self-directed studies (H. Schmidt, Henny, & de Vries, 1992). Students also find the new approach uncomfortable, "risky, scary, and uncertain" (ETE Team, n.d.). According to the ETE Team, it will take at least a year until the students get used to their new roles in PBL.

2.6.2.3 Assessment

Assessing students' performances has been a major challenge faced by PBL educators (Dennen, 2000; , *Disadvantages of Problem Based Learning*, 1996; ETE Team, n.d.; University of California Division of Undergraduate Education, 2000). There are various ways to assess students performance in PBL (*Disadvantages of Problem Based Learning*) but most of them are not as well defined as traditional ones. Therefore, students often have a hard time figuring out the grading criteria of their tutor (ETE Team). On the other hand, it is difficult enough for the instructor to monitor the progress of each group. It is even tougher for the instructor to monitor individual performances in every group. Unless all the group activities, group discussions, and individual work processes are documented thoroughly for every group, assessment will remain as one of the most challenging aspects in PBL (Dennen, 2000).

2.7 Conclusion

PBL is a new learning paradigm that is gaining popularity in recent years, especially in medical schools. Research has suggested that PBL brings about many advantages compared to traditional education. However, many challenges are faced by educators implementing PBL. PBL will be a close-to-ideal methodology to adopt if the challenges can be overcome while keeping its core promises. Researchers attempt to answer these challenges by combining PBL with modern, and multimedia technologies.

3 COMBINING TECHNOLOGY AND PBL

3.1 Introduction

Computing technology can be used in different ways in PBL. It can be used to alleviate the cost and resource demands of PBL (Finucane, Johnson, & Prideaux, 1998). It can also be used to help students in their decision making, data organizing and storage, and help develop students' computer skills (Koschmann, Kelson, Feltovich, & Barrows, 1996; Schank, Berman, & Macpherson, 1999a; Watson, 2002). Using technology to document group process, group discussions and findings can help students keep track of their progress while also helping facilitators enormously on assessment tasks (Schank, Berman, & Macpherson, 1999b). Another important use of technology in PBL is to use it as a communication medium for collaborative learning (Bannon, 1989; Winn, 1992). Technology can even be used to augment PBL and provide means for new activities that were impossible in the past (Bannon). Although technology can be utilized to tackle some of the challenges of or even enhance PBL, it should never be considered as a universal solution to every problem in PBL. Certain problems, such as the change of role and responsibilities of a tutor, cannot be miraculously fixed by adding technology to the process. Rather we should consider technology as the tools we could utilize to support, augment and resolve specific issues of PBL.

3.2 Existing Technologies

This section reviews how technologies have been used by researchers to support PBL.

3.2.1 Asynchronous Collaboration

Educators have been utilizing asynchronous communication technologies such as email, mailing lists, and online Bulletin Board System (BBS) to support PBL in the 80's as documented by Bannon (1989). Among these technologies, BBS is the most widely used and is still one of the major asynchronous communication medium employed to support asynchronous collaboration in PBL. It has been used in the Northwestern School of Medicine (Altman, 1997). Dennen (2000) in San Diego State University has also used BBS, called Web-based conferencing in her study, to test its effectiveness in online PBL.

BBS is a valuable tool especially for medical schools because it allows students to collaborate without meeting in person. Face-to-face meeting is a luxury to medical students because of their busy schedules. Since everything in BBS is documented by the system, it helps tutors assess individual students, also provides "a sense of accountability for all group members" (Dennen, 2000, para. 10). It also helps students track their group process. Its main disadvantage is the delay in response due to its asynchronous nature. Short discussions, like the clarification of an idea, that usually last a minute in face-to-face meetings may last several hours or even days. Difficulty in expressing non-textual clues, like feeling or tone, is also a common problem. Therefore, it is normally used for lengthy argumentative discussions or summary of research findings. Using BBS alone as the communication medium in a PBL class may hinder group process (C. Kamin, O'Sullivan, & Deterding, 2002).

3.2.2 Content Management

Instead of printing in books or papers, course contents can be stored in digital media or computers. For example, Holmes (2004) from Emory University developed

WebQuest that is a collection of webpages containing the content for the PBL lessons stored in a web server. The contents are specially designed for PBL where the students take the roles of notable scientists and defend their findings in order to arouse their interests in learning Atomic Theory. Researchers in Marquette University School of Dentistry have developed "case-based CD-ROM educational modules involving decision tree algorithms that can be used either individually or in small groups" (Whipp, Ferguson, Wells, & Iacopino, 2000). Instead of purely presenting content, students have to make choices on diagnosis and patient treatment. If they make a wrong decision, relevant data will be shown and students can try again.

In addition to providing cheaper and environmental friendlier storage for course content, this kind of system may allow interactivities that are hard to achieve in book or paper format. It also allows access to the content from anywhere with an Internet connection. These systems are best suited for self-studies or for a face-to-face group physically sharing a single computer.

3.2.3 Multimedia and Interactivity

Advancement in digital media technology provides a means of presenting content other than pure text and images. Image manipulation, digital video playback, and sophisticated interactions are achievable.

In terms of digital video usage, the Faculty of Nursing at the University of Newcastle uses a combination of digital video and text-based scenarios stored in compact disc or on web servers to deliver courses offshore (Conway, Little, & McMillan, 2002). The Learning through Interactive Video Education (L.I.V.E) project from University of

Colorado School of Medicine uses digital videos stored in compact disc for case presentation and physical exam demonstration for pediatric PBL (C. S. Kamin, Deterding, Wilson, Armacost, & Breedon, 1999). It also allows asynchronous collaboration in a format very similar to BBS.

On the image manipulation front, Exploring the Environment (ETE), developed by the NASA Classroom of the Future at Wheeling Jesuit University, is a series of webpages on environmental earth science for high school students' PBL studies (ETE Team, n.d.). Students can visualize images from real-world remote-sensing satellites. To help students analyze the data, ETE also contains instructions for manipulating the data through image editing software and even Geographical Information System (GIS) software like ArcView (ESRI, n.d.).

For advanced interactivity, the Southern Illinois University developed a software called Diagnostic Reasoning (Dorsey et al., 1996). It is a text-based patient simulation program allowing students to interact with the patient through a complete diagnostic process including interview, physical exam, lab test, diagnosis and treatment. Limited graphical capabilities are available. For example, the face of the patient is shown while interviewing the patient and the body of the patient is shown when performing physical exam. Utilizing Shockwave technology (Adobe Systems Incorporated, 2006a), Legacy Interactive has developed the online game ER-SIM that is similar to Diagnostic Reasoning but features animations, sound and voice acting for the patient (Legacy Interactive, 2001).

Students find interactive and media rich cases more engaging (Dorsey et al., 1996; C. S. Kamin, Deterding, Wilson, Armacost, & Breedon, 1999). With digital video, students can observe the patient's non-verbal cues such as facial expressions, gestures, and voice loudness. By watching how a practitioner responds and questions the patient, they can enhance their interviewing skills. Moreover, an experiment by the L.I.V.E team suggested that digital video case enhanced critical thinking (C. Kamin, O'Sullivan, & Deterding, 2002). Working with real-life data and working with simulated patients allow students to practice their skills similar to the way they work in their future careers. Like content storage systems, these systems are best suited for individual self-learning or for small group physical sharing of the same computer. Although L.I.V.E's BBS like functionality is meant to be used for online collaboration, almost no evidence of group process was found (C. Kamin, O'Sullivan, & Deterding).

3.2.4 Synchronous Collaboration

Text conferencing has been used to achieve synchronous communication in PBL in the 80s (Bannon, 1989). The recent advancement in web conferencing tools brings online synchronous collaboration very close to face-to-face interaction. There are numerous web conferencing software tools available offering similar capabilities such as Netmeeting (Microsoft, 2004), Microsoft Office Live Meeting (Microsoft, 2006), WebEx (WebEx Communications Inc., n.d.), Elluminate Live! (Elluminate Inc., n.d.), PictureTalk (Pixon Inc., 2006), eAuditorium (NetDIVE Inc., n.d.), and Macromedia Breeze (Adobe Systems Incorporated, 2006b). The list goes on. These software packages offer audio, and for some of them video, conferencing for communication, whiteboarding for collaboration, application or screen sharing for presentation or reporting purposes, and recording for archiving the meeting session.

A web conferencing system is ideal for synchronous online PBL sessions, supporting collaboration to a degree similar to face-to-face meetings. The recording capability archives the session for future references and for a facilitator to assess individual students' performances. It makes class scheduling easier since students and tutor can meet online even when they are at sparsely located clinical sites. However, it provides minimal means for asynchronous communication. Also, since it does not provide the tools to store and organize the course content as a flow, the facilitators have to manually locate and feed content to the shared screen in real time. Therefore, augmenting it with an asynchronous communication tool, like a BBS, and course content management tools, such as WebCT (WebCT Inc., n.d.), can make it work better although this means the students and the facilitator have to use two or three different and non-interoperable software, simultaneously. One of the directions of COMPS research used Elluminate Live! together with WebCT to test the effect of narrative in PBL (Schell, n.d.; Schell & Kaufman, n.d.).

3.2.5 Administration and Others

Other than supporting learning activities, technology can be employed to help in administration purposes. Dublin Dental School & Hospital is looking at the use of computer software for PBL curriculum management (McLoughlin, 2005). The software will be used to manage the content of the blocks, map the curriculum, collecting feedback and grades for students and tutors. Automating these tasks could save a lot of administrative time and trouble, hence reducing costs.

Co-operative Problem-based learning Meta-model (CPM) is not really a tool to support PBL directly (Laforcade, Barbier, Sallaberry, & Nodenot, 2003). Rather, it is a description language built on top of UML to semi-formally describe a learning scenario and document it. It is uncertain whether tutors without extensive computing skills can use it.

3.3 Conclusion

We have reviewed how educators have been using technology to support and enhance PBL experience. Some of them can be answers to the problems of implementing PBL. For example, communication tools, administrative tools, and content repository tools can save the cost of implementing PBL in terms of time and money. Documentation provided by asynchronous communication tools and the recording functionality of web conferencing tools make individual assessment easier. Media-rich and interactive applications increase engagement of students and allow activities not possible without the use of technology. The PBL experience is thus enhanced.

We have also discussed the shortcomings and the absences of certain features of each type of software. One way to solve this problem is to integrate applications from several categories that support various PBL functionalities, together. This allows the applications to complement each other's weaknesses. Consider the following ideal scenario:

The content management system manages all content and materials for use in a PBL class. The multimedia and interactivity capable system presents the content fed by the content management application to the students and tutors. The synchronous collaboration system provides a means for the multimedia rich, interactivity capable PBL class to be held online instead of face-to-face. The asynchronous collaboration system provides a means for the students and tutors to have long discussions when they cannot schedule a time for their meeting. The administration system records the class schedules, the students and the facilitator for each scheduled class, and the material the content management system should bring up for each scheduled class. By integrating the functionalities in an interoperable fashion the advantages of each system are kept while the weaknesses are complemented by other systems.

However, there are two major drawbacks with this approach. First, the students and the instructor have to learn to use several distinct systems. It can take a lot of the time and effort for them to be familiar with several platforms where the resources are expected to be spent on PBL itself. Window management is also a problem when there are several applications running at the same time. Second, there are no well-defined ways for the distinct systems to import and export content to each other. For example, the course content is managed in WebCT (WebCT Inc., n.d.) but there is no simple way to stream and navigate the stored content to WebEx (WebEx Communications Inc., n.d.) to be used in an online PBL session. Another example concerns the student accounts set up in WebEx for use in online PBL lessons. The same user account will not exist in a BBS until the student or the tutor creates it manually for the student. Adding the school registration system's student account to the picture adds to the complexity of the situation. The resources needed for administration will become huge because every single change needs to be reflected in all systems.

To solve all these problems in a clean manner, the SAGE-SIM team decided to develop COMPSOFT as one of the COMPS projects, a software platform specifically designed and customized for use in PBL.

4 COMPSOFT

4.1 Introduction

COMPSOFT is a software platform designed by the SAGE-SIM team and developed by this researcher. It supports and enhances PBL specifically for use in medical schools. It provides functionalities in five aspects: administration, content management, multimedia and interactivity, synchronous collaboration, and asynchronous collaboration. We have seen in the previous section how software systems with these functionalities work together in an ideal scenario. Because COMPSOFT is a single integrated platform, it not only alleviates the difficulty of learning to use several applications from students and facilitators, but also allows seamless data transfer between functionality boundaries. Therefore, the two problems associated with utilizing several systems discussed in the previous section are effectively resolved.

4.2 A Typical PBL Curriculum

Before we delve into the details of the functionalities and user interfaces of COMPSOFT, we will first look at the process and the activities involved in a typical medical PBL curriculum.

Before the lessons begin, students need to register for the courses[1] they want to attend and they will be assigned to groups for group discussions and activities. One or more tutors responsible for a course need to make decisions on the PBL cases and the

[1] The terms class and course will be used interchangeably.

order they are covered. For example, in a paediatric course, a facilitator may decide to cover a PBL case of asthma, then autism, followed by a case of otitis media.

During class time, the facilitator and the students meet at the assigned room. The tutor provides students with case related materials. As the case unfolds, the students will be guided through the materials, have group discussions, and identify learning issues with the guidance of the facilitator. At the end of the lesson, the learning issues identified may be assigned to groups or individual students. Students are required to perform self-directed learning on their assigned learning issues before the start of the next lesson. When the next lesson starts, students report and discuss their findings. Then, they will continue the PBL case studies where they left off. When the tutor decides that a PBL case has been covered thoroughly, the next case will be started.

Typically, the flow of a medical PBL case can be divided into phases and activities. A phase is an aspect of a case that should be covered while activities are the things students perform during a certain phase in the case study. For example, in an asthma PBL case study, the following phases will be covered in order: case presentation, patient history, physical exam, lab test, and diagnosis. During the patient history phase, the following activities will be performed: presenting the patient history in print form, clarifying the terms through question and answer session, identifying the key facts through discussion, discussing possible hypothesis for the patient's problem, ranking the hypotheses based on seriousness, and identifying physical exams needed for the patient. Although the flow is pre-authored, the facilitator may need to jump around the phases and activities, eliminate or inject new discussion activities during the lesson because of the student-directed and dynamic nature of PBL. A sample flow of a PBL case study with five phases and twenty-six activities in total is shown in Figure 4-1.

Figure 4-1: A Sample Flow of a PBL Case Study

Phase One – Case Presentation

1. Present the case in print form (1/2 – 1 typed page).		
2. Clarify the terms and concepts (common understandings)	→	Identify learning issues.
3. Identify the problem. What are the key facts?	→	Identify learning issues.
4. Brainstorm possible explanations (or hypothesis).	→	Identify learning issues.
5. Rank or rate the explanations (e.g., low, medium, high)	→	Identify learning issues.
6. Identify history questions and discuss the reasons for selecting these questions.	→	Identify learning issues.

Phase Two – Patient History

1. Get a history.	→	Identify learning issues.
2. Clarify the terms and concepts (common understandings).	→	Identify learning issues.
3. Clarify the problem. What are the key facts?	→	Identify learning issues.
4. Discuss the possible explanations (or hypothesis).	→	Identify learning issues.
5. Rank or rate and discuss the explanations (e.g., low, medium, high).	→	Identify learning issues.
6. Identify the required physical examination and explain why.	→	Identify learning issues.

Phase Three – Physical Exam

1. Conduct the examination.	⇒	Identify learning issues.
2. Get the results.	⇒	Identify learning issues.
3. Clarify the terms and concepts (common understandings).	⇒	Identify learning issues.
4. Clarify the problem. What are the key facts?	⇒	Identify learning issues.
5. Discuss the possible explanations (or hypothesis).	⇒	Identify learning issues.
6. Rank or rate the explanations (e.g., low, medium, high).	⇒	Identify learning issues.
7. Identify the lab tests and explain why.	⇒	Identify learning issues.

Phase Four – Lab Test

1. Get the lab results.	⇒	Identify learning issues.
3. Clarify the terms and concepts (common understandings).	⇒	Identify learning issues.
4. Clarify the problem. What are the key facts?	⇒	Identify learning issues.
5. Discuss the possible explanations (or hypothesis).	⇒	Identify learning issues.
6. Rank or rate the explanations (e.g., low, medium, high).	⇒	Identify learning issues.

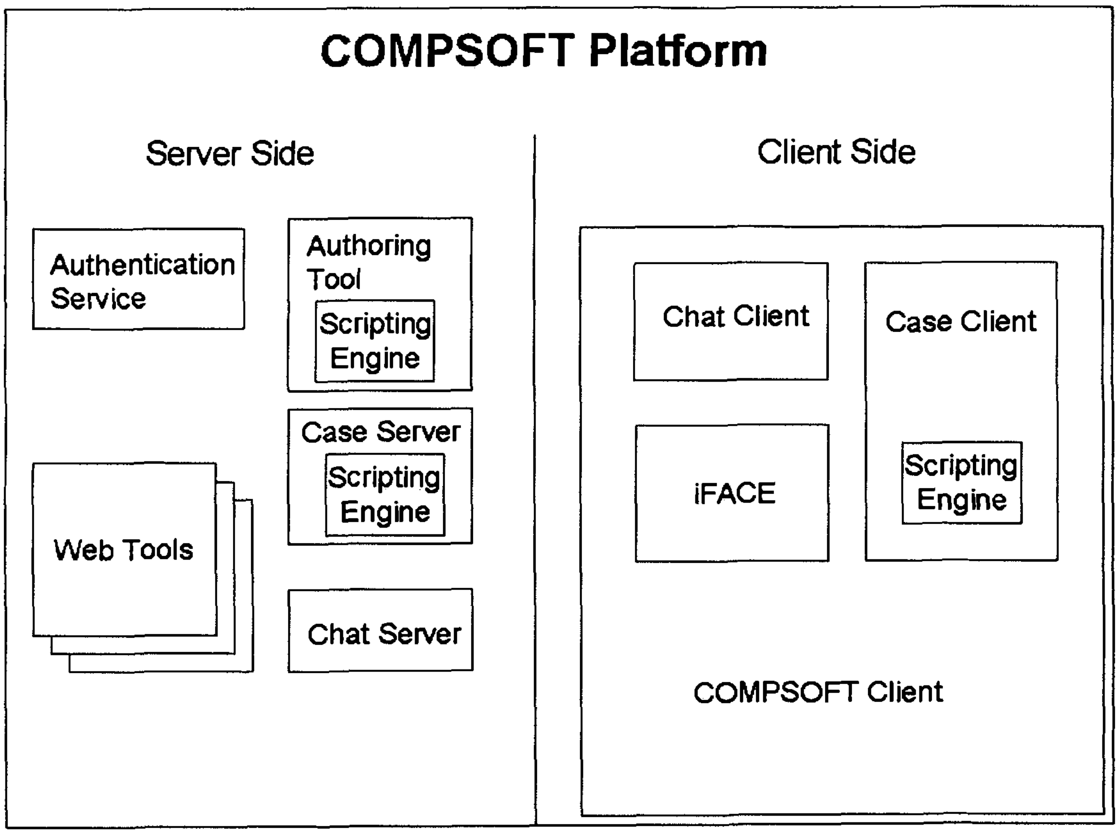
Phase Five - Diagnosis

1. Report your diagnosis.		

4.3 Components

COMPSOFT consists of a number of individual software components that interact with each other to support many aspects of a PBL curriculum. The main components are the authentication service, the scripting engine, the authoring tool, the chat server and the chat client, the case server and the case client, the chat server console and the case server console, the Interactive Face Animation - Comprehensive Environment (iFACE) (DiPaola & Arya, n.d.), the web tools, and the COMPSOFT client. These components are shown in Figure 4-2.

Figure 4-2: COMPSOFT Components



4.3.1 Authentication Service

The authentication service is a vital part of COMPSOFT that handles many of the administrative tasks of PBL. It manages the following registration information:

- The information of facilitators and students such as their name, user ID, and password
- Available classes, the cases covered in each class, the preliminary start date of each case, and whether the case has been finished
- The list of classes a student is registered in and the group the student belongs to in a certain class
- The list of classes a facilitator is responsible for

The above information can be added, removed or edited through the authentication service. The authentication service is implemented in a way that allows it to communicate with existing school registration systems so that many of the tasks can be automated.

During a PBL lesson, it authenticates students and facilitators, directing them to the appropriate chat and case servers when they are successfully authenticated.

4.3.2 Scripting Engine

The scripting engine manipulates case scripts. A case script is an artifact created by the PBL case author that specifies everything about the phases and activities of the case, their ordering, and the locations of the resources and the materials needed for each activity. The ordering of phases can be sequential or can involve branching and cycling. It also records the actual flow of the case study, known as the agenda, in terms of the actual order of the phases and activities performed by the students from the start of the case study so that students and tutors can revisit them, and continue at the point where they left off in the next lesson. The case script is designed to be in plain text Extensible Markup Language (XML) (*Extensible Markup Language (XML)*, n.d.) format to make it easy for editing by hand or by authoring tools. A sample case script can be found in Appendix A[2].

The scripting engine reads and interprets everything recorded in a case script into a form usable by different components of COMPSOFT such as the case server, the case client, and the authoring tool. It also monitors the current PBL activity and adds it to the agenda section of the case script when it is done. In addition, the scripting engine is capable of encrypting and decrypting certain activities of the case script. This feature allows the use of Guided Design or Progressive Disclosure Model, the release of the materials is controlled so that students need to work with limited data to identify their individual learning issues (C. S. Kamin, Deterding, Wilson, Armacost, & Breedon, 1999). This is considered as an attractive feature for "staging a PBL problem in a distance-learning environment" by Watson (2002, p. 3). For example, when discussing what physical exams should be administered to the patient, a student cannot look at the list of physical exams from the materials in the next phase and reveal them to ruin the PBL experience. Furthermore, the scripting engine allows jumping to any phase of the script without confining to a pre-authored flow. It can also inject new discussion activity at any point in the PBL lesson. These two capabilities are ideal for the dynamic nature of a PBL lesson.

[2] Note that PBL phase is named as node in the case script.

Through the scripting engine, activities of a PBL case study can be performed following the predetermined order in the case script or in a free style manner when needed. Resources and materials needed for the current activity can be retrieved. Uncovered materials can be protected. New activity can also be added when the facilitator sees fit. In addition, an agenda recording the flow of the case study is marked in the case script so that students and facilitators can revisit them or continue from where they left off.

4.3.3 Authoring Tool

The authoring tool makes use of the scripting engine and provides an easy to use interface for a case author to create a case script. Although the case script can be authored by typing in, case authors may be alienated by the XML syntax. Therefore, an authoring tool with an easy to navigate graphical user interface is desired. It is also used to manage materials and resources needed for each PBL activity. A case author creates a case script and assigns resources for use in each activity through the authoring tool. The authoring tool will create a directory structure to efficiently store and categorize the content. Therefore, the case author does not need to specify the location of the materials in the script but the authoring tool will automate the task when the material is linked to an activity. Upon completion, the authoring tool will commit the changes and add the case to the class by contacting the authentication service.

The author tool is not yet completed in the current version of COMPSOFT.

4.3.4 Chat Server and Chat Client

These two components work together to provide synchronous text and voice chat capabilities in COMPSOFT. Each chat server corresponds to a single class where every chat client of that same class connects. A student or the facilitator can choose to talk to the whole class, talk to a group only, or talk to a specific student in the class privately. Every conversation and its time, except private communications, will be logged by the chat server for future reference or for grading purposes if server logging is enabled. Other functionalities such as listing the groups and the students in the current class are also available.

Voice chat has not been implemented in the current version of COMPSOFT.

4.3.5 Case Server and Case Client

These two components work together to control the flow of the case during an online PBL lesson. Like the chat server, each case server corresponds to a single class where every case client of that same class connects. Every activity and its timestamps are also logged by the case server if logging is enabled. Both the case server and the client make use of the scripting engine to offer their functionalities.

A student or a facilitator connects to the case server through the case client. Upon successful connection and authentication, the case server provides the case client with information such as the class name, the current case name, the web location of the case script, the group the student belongs to, the list of group members who have logged in, the facilitator information, and the facilitator's IP address[3]. The case client will then be able to download the case script and retrieve the information from it using the scripting

[3] This is used for iFACE as described in the iFACE section.

engine. The agenda and the current activity will then be presented to the student or the facilitator by the case client.

The case server will also inform the case client whether the client will have the privileges of a facilitator at the time of authentication. If so, the case client will have additional privilege to control the flow of the PBL case study. The facilitator can issue commands to the case client to go to the next PBL activity, go to the selected PBL phase in a branch, jump to a specific PBL phase, or inject a new discussion activity at any time. The command will be delivered to the case server where it will be validated and distributed to every case client connected to the case server. Each case client then updates the agenda and presents the new activity to the students.

In summary, the case client presents case related information to the students and the facilitator through the scripting engine. The facilitator uses the case client to control the flow of the PBL case study while the case server synchronizes the change to every case client.

4.3.6 Chat Server Console and Case Server Console

The chat server console and the case server console provide an easy to use graphical interface to manage the lifetime of chat servers and case servers for the PBL classes.

To start a chat/case server, the chat/case server console connects to the authentication service and retrieves a list of classes where there are no corresponding chat/case servers. The administrator can then select a class from the list and start a chat/case server for it. The chat/case server console then register the chat/case server to the authentication service for the class along with the server's connection parameters. This allows the authentication service to direct successful authenticated student or facilitator to connect to the appropriate chat and case servers.

Similarly, when the chat/case server console stops a chat/case server, it informs the authentication service to remove the corresponding entry.

Another functionality of the chat/case server console is to monitor the health of the chat/case servers. It can ping those servers and see if they are running. If any of them are dead or crashed, it will terminate them and remove their entries from the authentication service.

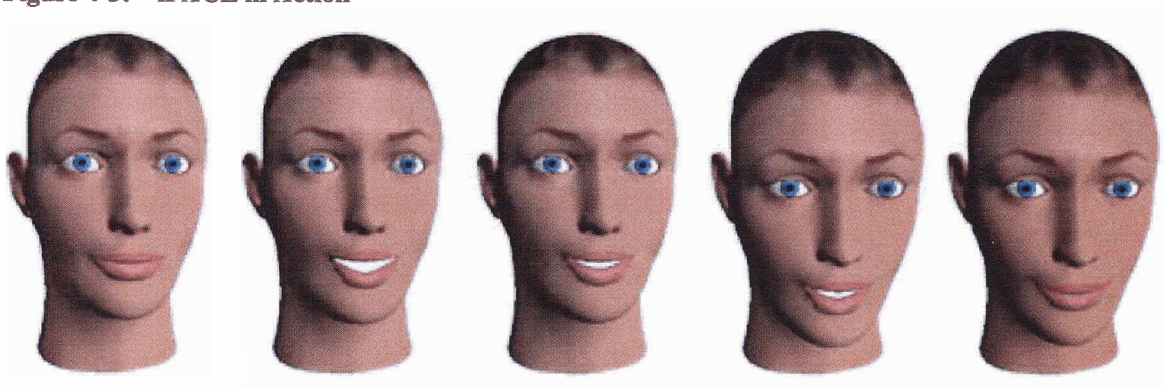
4.3.7 iFACE

iFACE (Arya & DiPaola, 2004; Arya, DiPaola, Jefferies, & Enns, 2006; DiPaola & Arya, 2005, , n.d.) is a powerful interactive 3D face engine. It presents a 3D face that animates and performs lip-sync according to voice data and behavioural parameters. It is easy to modify the 3D face model and the behaviour (or personality) by the provided tools. There are three basic ways for users to interact with the 3D face:

1. Speak directly through a microphone – the 3D face will perform lip-sync and animate according to the preset emotional parameters, and the characteristics of the voice data such as pauses, emphasis, and volume.
2. Play a pre-recorded sound file – the 3D face will animate the same way as if the played voice is from a microphone.
3. Play a voice data over the Internet – the 3D face will animate the same way as if the voice data from the network is from a microphone.

Figure 4-3 shows iFACE in action (Arya, DiPaola, Jefferies, & Enns, 2006).

Figure 4-3: iFACE in Action



Arya (2004) has discussed the importance of a "face multimedia object" that is implemented by iFACE. iFACE provides an engaging, media-rich and interactive medium that can be used in a number of ways in COMPSOFT:

- As a representation of the facilitator during an online PBL session – The facilitator speaks through the 3D face where the students see the 3D face animate with the instructor's voice at the other end of the network.
- As a virtual patient – The virtual patient can be driven by pre-recorded voice data. This is typically used for the patient to present his/her problems. The virtual patient can also be driven by someone, such as the facilitator, speaking through the microphone on the other side of the network. This is good for students practicing their interviewing skills.

We have seen how multimedia enhances PBL. The L.I.V.E. project uses video to promote students' critical thinking (C. Kamin, O'Sullivan, & Deterding, 2002). Like video, iFACE's simulated patient allows students to observe the visual cues from the patient in a more realistic way when a patient is too young to express his/her problems in words (C. S. Kamin, Deterding, Wilson, Armacost, & Breedon, 1999). iFACE is even better than video because it provides a realistic 3D animation that is interactive and more importantly new interactions can be generated by the system without involving human participants. For example, with iFACE, students are not only able to watch the interview of a patient but they can also participate in the interviewing process by asking questions through COMPSOFT's chat client. The facilitator can then react accordingly. This kind of interactivity is not available with video technology alone.

To use iFACE as a virtual patient, the tutor only needs to change the 3D face model or texture with the provided tools, e.g. increase the width of the eyes by ten units, select the emotional parameters, e.g. select the parameter sad, and then record the patient's speech or speak through the 3D face during the class. Compared to hiring an actor patient and some video technicians, preparing the video equipment, putting makeup on the actor, practicing and taking the video for a patient interview, using iFACE can save enormous amount of time and resources. This alleviates a big challenge faced by medical schools in implementing PBL curricula.

iFACE can be used to protect privacy. For example, instead of hiring actors, we can have real patients interviewed by students through the 3D face because the real face of the patient is hidden behind the 3D avatar. Similarly, a tutor can facilitate a class from home wearing pajamas and having a facial mask on his face that is not possible with video/audiographic conferencing system where students can see the face of the facilitator.

Since many people link 3D animations and 3D avatars to video games, having an animated 3D face could make the PBL session more interesting and engaging for the students. The dramatic video portrayal of the 3D face will make students more involved (Foley, Polson, & Vance, 1997).

4.3.8 Web Tools

Three web tools are designed and developed for COMPSOFT: a BBS discussion activity tool, a library tool, and a web whiteboard. The BBS provides a channel for asynchronous collaborations and discussions. Watson (2002) stated that a website was needed to provide supporting materials to students that were not covered during class time. Therefore, we have designed a library web tool that a facilitator of a given class can add and remove materials as well as controlling when to release the materials to the students. As in other synchronous collaborating system, a whiteboard is a very useful tool for online collaboration. These tools are required to be able to communicate with other COMPSOFT components. For example, the BBS needs to communicate with the authentication service so that a student account will be created in the forum associated with a particular class when the student is added to the class. Also, when a user logs in through the COMPSOFT client, she should also be logged into the BBS automatically.

The chat server and chat client, iFACE, the web whiteboard, and the BBS work together to provide an appropriate channel of communication to support different kinds of group collaborations. The case server and case client also play a part in synchronizing group activities. Camp (1996) stated that this can save faculty time for "individual mentoring, observation, and feedback" (p. 5).

The discussion activity tool is a web application that takes a discussion topic and a URL for the reference materials to generate a web page that shows the same. Students can look at the topic for the discussion and the reference material in the same webpage. It also allows the user to launch the reference materials in a separate browser window if it is too big to be embedded into the single page. This tool is used for discussion activities in PBL lessons.

The current version of COMPSOFT does not have completed versions of the library tool and the web whiteboard.

4.3.9 COMPSOFT Client

The COMPSOFT client is the application used by students and facilitators for PBL course interactions. It contains a chat client, a case client, an iFACE instance, a web browser and some additional user interface elements. Students and facilitators interact with these components through the COMPSOFT client application. The web browser in COMPSOFT serves as the presentation medium where content fed by the case client are displayed. The COMPSOFT web tools are also accessed through the web browser. Because of the advancement of web technologies, many different kinds of media can be shown by a browser such as video, sound, pictures, text, interactive media like Shockwave or Flash (Adobe Systems Incorporated, 2006c) games, and various web applications. Therefore, using a browser as the multimedia panel allows a case to be constructed with rich multimedia in mind and allows the use of narrative pedagogy (Diekelmann, 1995; Swenson & Sims, 2000). Imagine a student performing physical exams to a patient interactively where the student can drag and drop tools to the appropriate body parts of the virtual patient where the patient will react based on the exam performed. To enhance reusability, existing web based PBL modules, such as the ETE (ETE Team, n.d.), and existing educational games and simulations (Whipp, Ferguson, Wells, & Iacopino, 2000), can be incorporated directly into COMPSOFT. The possibilities are enormous.

This section discussed the general ideas about the functionalities of COMPSOFT components. The next section will examine the user interfaces of COMPSOFT.

4.4 User Interfaces

COMPSOFT features a number of user interfaces (UI) to interact with different components of the platform. As Holmes (2004) points out, lack of computer skills could hinder a web-based PBL lesson. In consideration of this each interface is designed with simplicity in mind. There are six UIs in total. They are: the authentication service interface, the authentication service web UI, the chat server console UI, the case server console UI, the authoring tool UI, and the COMPSOFT client UI. These interfaces and their features are presented in the following sections.

4.4.1 Authentication Service Interface

Rather than a UI, the authentication service interface features an interface for use by other computer systems and components. It allows any existing school registration system and the other components of COMPSOFT to interact with it, remotely. We will revisit this interface when we discuss the technical details of COMPSOFT.

4.4.2 Authentication Service Web UI

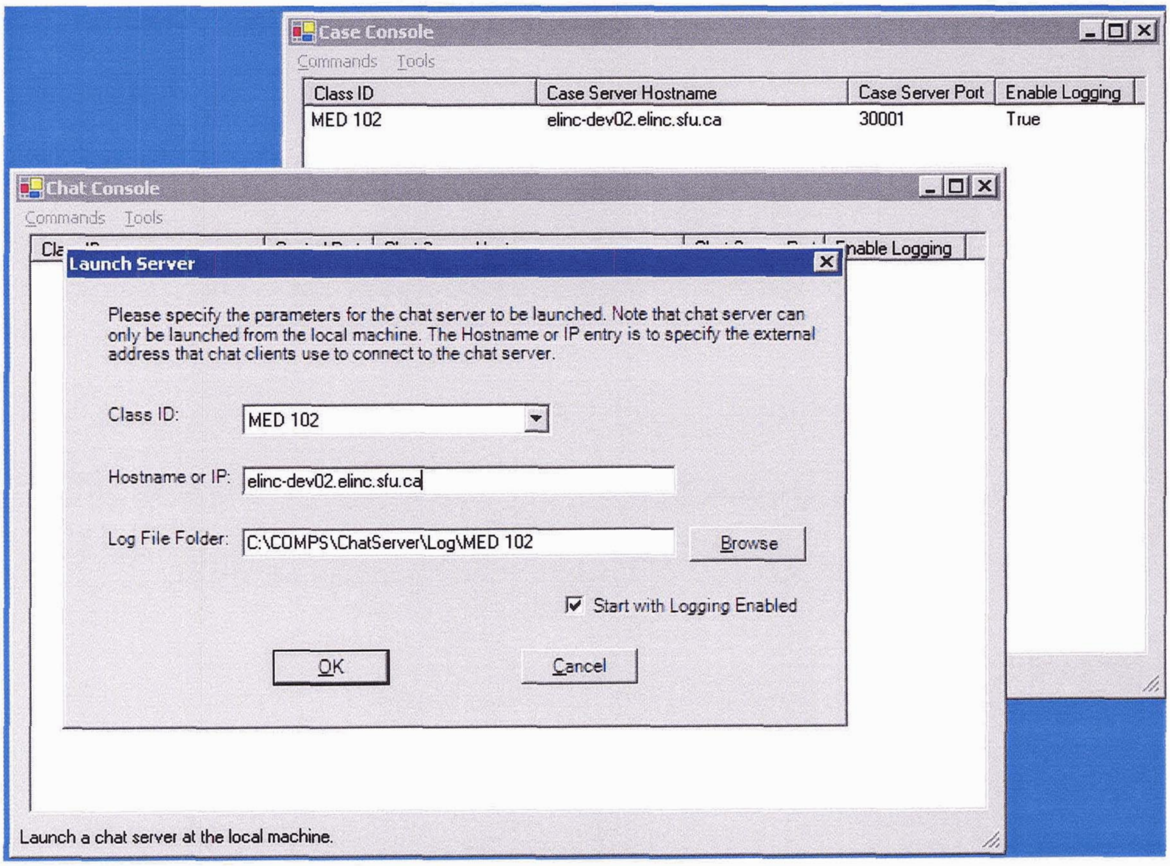
The authentication service web UI provides a graphical interface for faculty to interact with the authentication service manually. It is used when the function to pass registration information to the COMPSOFT authentication service has not been added to the existing registration system. It is also used when facilitators need to add cases to a class before the authoring tool is available or when performing small-scale experiments.

The Authentication Service Web UI is not available for the current version of COMPSOFT.

4.4.3 Chat Server Console UI and Case Server Console UI

The chat server console UI and the case server console UI are depicted in Figure 4-4.

Figure 4-4: Chat Server Console UI and Case Server Console UI



Class ID	Case Server Hostname	Case Server Port	Enable Logging
MED 102	elinc-dev02.elinc.sfu.ca	30001	True

Please specify the parameters for the chat server to be launched. Note that chat server can only be launched from the local machine. The Hostname or IP entry is to specify the external address that chat clients use to connect to the chat server.

Class ID: MED 102

Hostname or IP: elinc-dev02.elinc.sfu.ca

Log File Folder: C:\COMPS\ChatServer\Log\MED 102

☑ Start with Logging Enabled

Launch a chat server at the local machine.

Both UIs look very similar. The only difference between them is the type of servers they manage. From the menu, users can launch a server, ping all servers to make sure they are running, shutdown all the existing servers, and an option to specify the range of port numbers to be used when launching a server. The UI features a list view showing the list of servers launched, the class each server is responsible for, the hostname and port number the server is listening to and whether logging is enabled. Users can click on one of the entries and access functions including pinging that server, enabling or disabling logging, or shutting down that server through the pop up menu. Also shown in the figure is a dialog box for launching a server where the user can choose from a list of classes without a corresponding server, modify the IP address of the server[4], and the directory where the log file is located.

4.4.4 Authoring Tool UI

The Authoring Tool UI is not implemented in the current version of COMPSOFT.

4.4.5 COMPSOFT Client UI

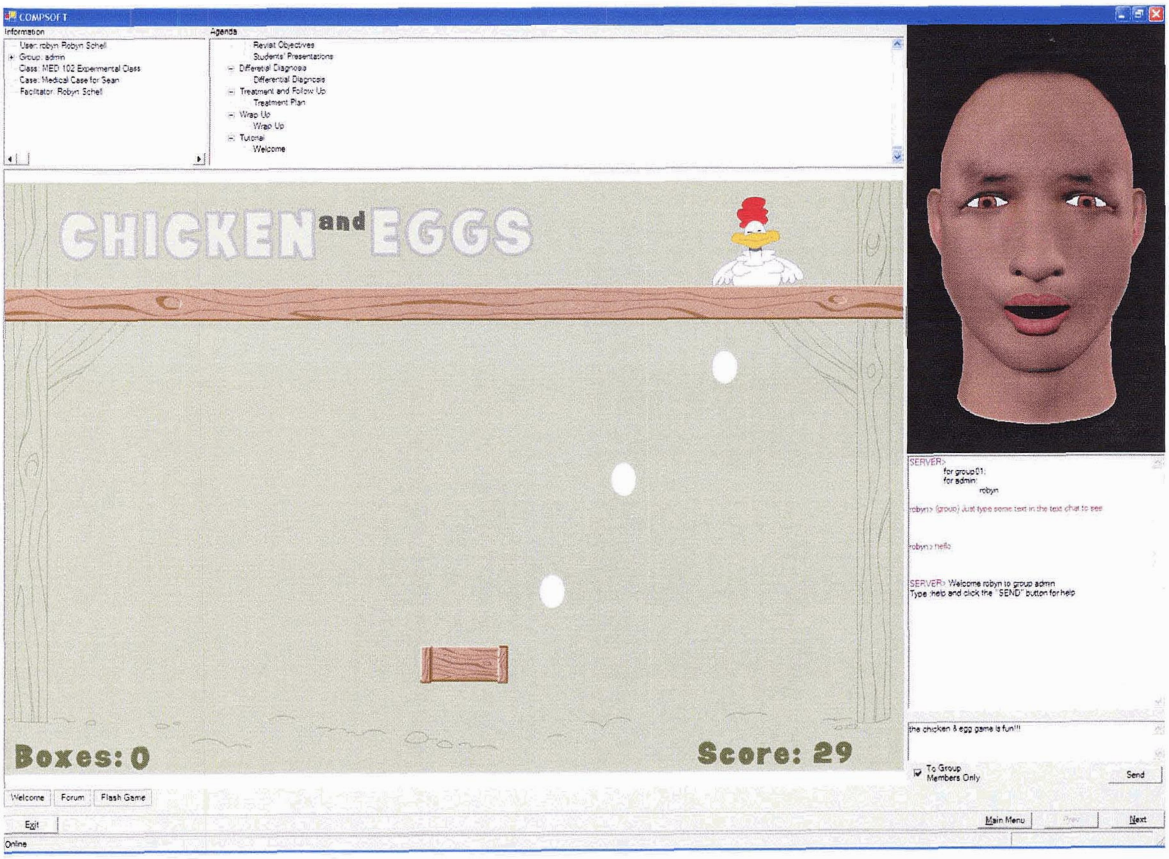
This is the only UI that the students work with and the UI that facilitators use most often. This is also the most complicated UI in the platform because many features are integrated into this interface. Therefore, special care has been taken and human-computer interaction (HCI)[5] (*Human-computer interaction - Wikipedia, the free encyclopedia*, n.d.) principles have been considered in the design of this interface, such as reducing windows-housekeeping actions, color coding texts for speed recognition, minimizing users' input actions, organizing menu items in a meaningful way and ordering them based on ease of use, weight balancing the composition of UI elements, and

[4] This is useful because a machine may have more than one IP addresses.

[5] "Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." (Hewett et al., 1996)

maintaining position consistency of menu items(Shneiderman & Plaisant, 2004; Tidwell, 2005). These will be covered in the following discussions. Although the UI accommodates many features that can possibly be presented across multiple windows, we decided to keep it a single window application because windows-housekeeping actions can hinder the performance of users and induce mistakes (Shneiderman & Plaisant, 2004, p. 503). The following are screenshots of the COMPSOFT client UIs. Figure 4-5 features a Flash game in COMPSOFT client. Figure 4-6 shows the BBS within the client UI. Figure 4-7 shows a virtual patient interview through iFACE in the client. Figure 4-8 depicts video in COMPSOFT client. Figure 4-9 shows a medical report in the form of a PDF document in COMPSOFT client.

Figure 4-5: Flash Game in COMPSOFT Client



User: robyn Robyn Schell
Group: admin
Class: MED 102 Experimental Class
Case: Medical Case for Sean
Facilitator: Robyn Schell

SERVER>
for group01:
for admin:
robyn

robyn> (group) Just type some text in the text chat to see

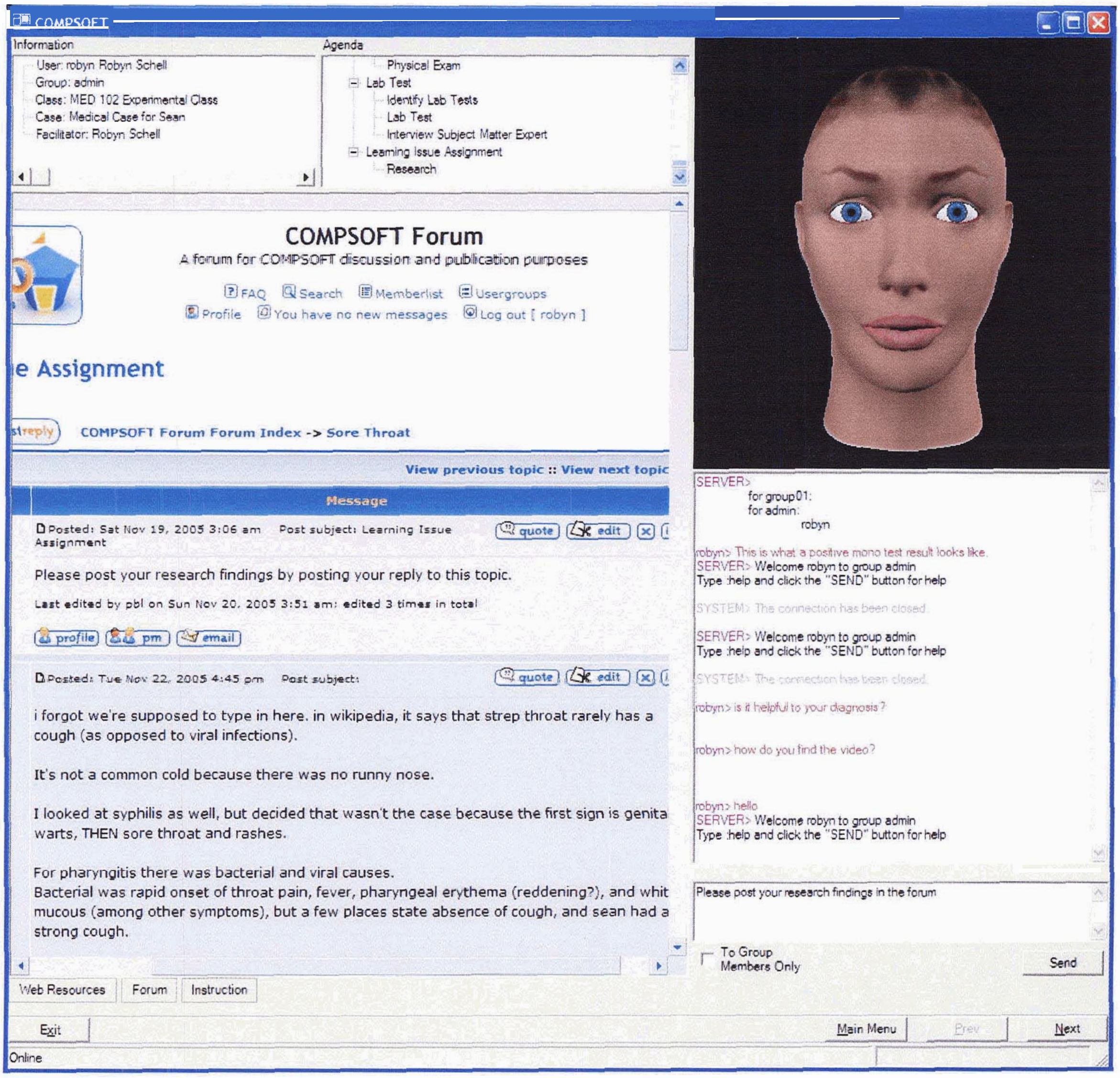
robyn> hello

SERVER> Welcome robyn to group admin
Type :help and click the "SEND" button for help

the chicken & egg game is fun!!!

☑ To Group Members Only

Figure 4-6: BBS in COMPSOFT Client



User: robyn Robyn Schell
Group: admin
Class: MED 102 Experimental Class
Case: Medical Case for Sean
Facilitator: Robyn Schell

Physical Exam
Lab Test
Identify Lab Tests
Lab Test
Interview Subject Matter Expert
Learning Issue Assignment
Research

COMPSOFT Forum

A forum for COMPSOFT discussion and publication purposes

e Assignment

COMPSOFT Forum Forum Index -> Sore Throat

Posted: Sat Nov 19, 2005 3:06 am Post subject: Learning Issue Assignment

Please post your research findings by posting your reply to this topic.

Last edited by pbl on Sun Nov 20, 2005 3:51 am; edited 3 times in total

Posted: Tue Nov 22, 2005 4:45 pm Post subject:

i forgot we're supposed to type in here. in wikipedia, it says that strep throat rarely has a cough (as opposed to viral infections).

It's not a common cold because there was no runny nose.

I looked at syphilis as well, but decided that wasn't the case because the first sign is genita warts, THEN sore throat and rashes.

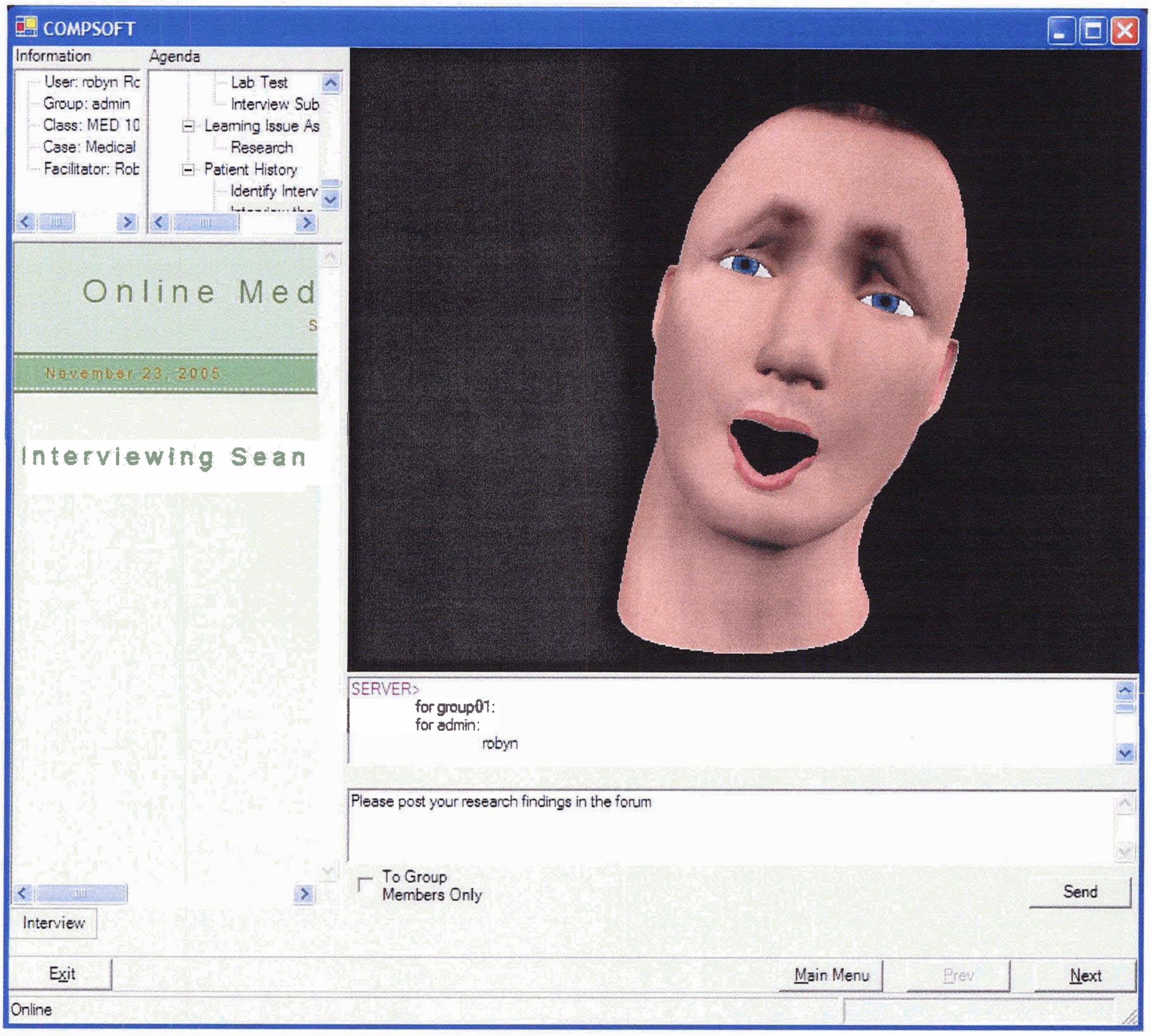
For pharyngitis there was bacterial and viral causes.
Bacterial was rapid onset of throat pain, fever, pharyngeal erythema (reddening?), and whit mucous (among other symptoms), but a few places state absence of cough, and sean had a strong cough.

```
SERVER>
        for group01:
        for admin:
                robyn

robyn> This is what a positive mono test result looks like.
SERVER> Welcome robyn to group admin
Type :help and click the "SEND" button for help

SYSTEM> The connection has been closed.

SERVER> Welcome robyn to group admin
Type :help and click the "SEND" button for help

SYSTEM> The connection has been closed.

robyn> is it helpful to your diagnosis?

robyn> how do you find the video?

robyn> hello
SERVER> Welcome robyn to group admin
Type :help and click the "SEND" button for help
```

Please post your research findings in the forum

□ To Group Members Only

Figure 4-7: Interviewing Virtual Patient through iFACE in COMPSOFT Client

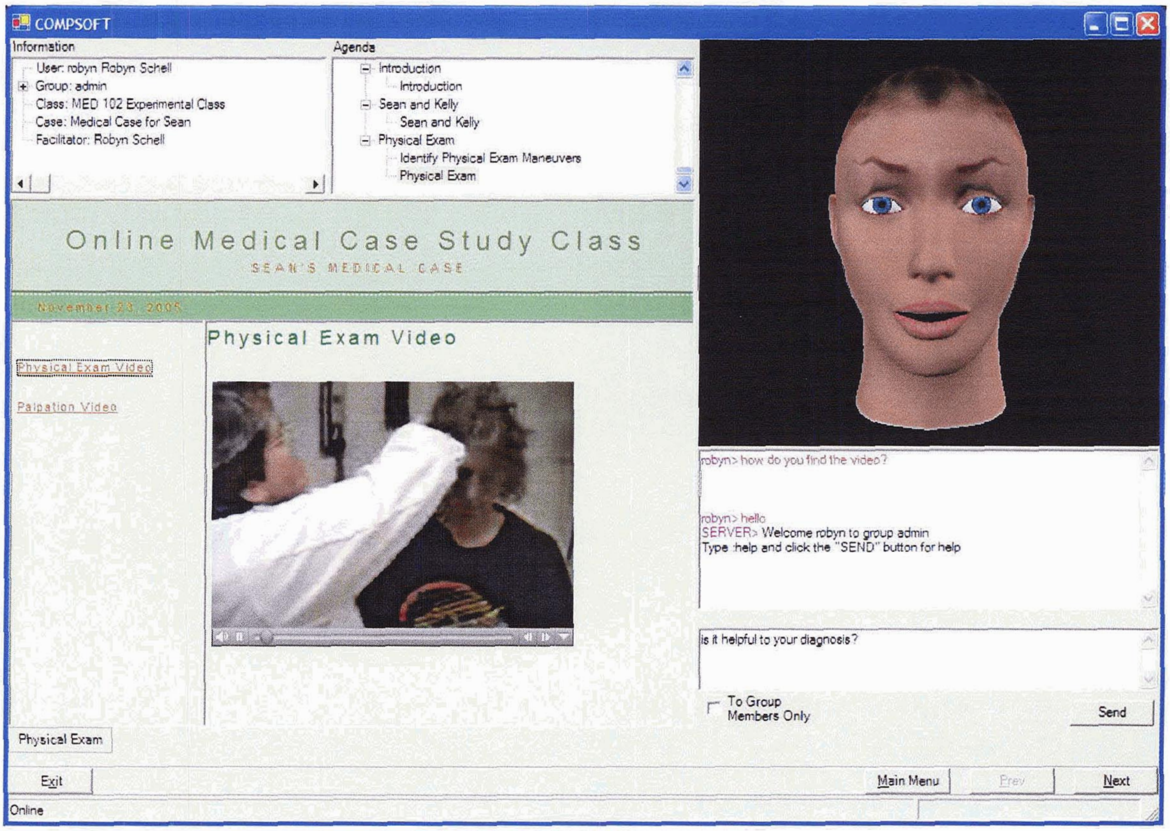


User: robyn Rc
Group: admin
Class: MED 10
Case: Medical
Facilitator: Rob

SERVER>
for group01:
for admin:
robyn

Please post your research findings in the forum

Figure 4-8: Video in COMPSOFT Client



User: robyn Robyn Schell
Group: admin
Class: MED 102 Experimental Class
Case: Medical Case for Sean
Facilitator: Robyn Schell

Online Medical Case Study Class

SEAN'S MEDICAL CASE

November 23, 2005

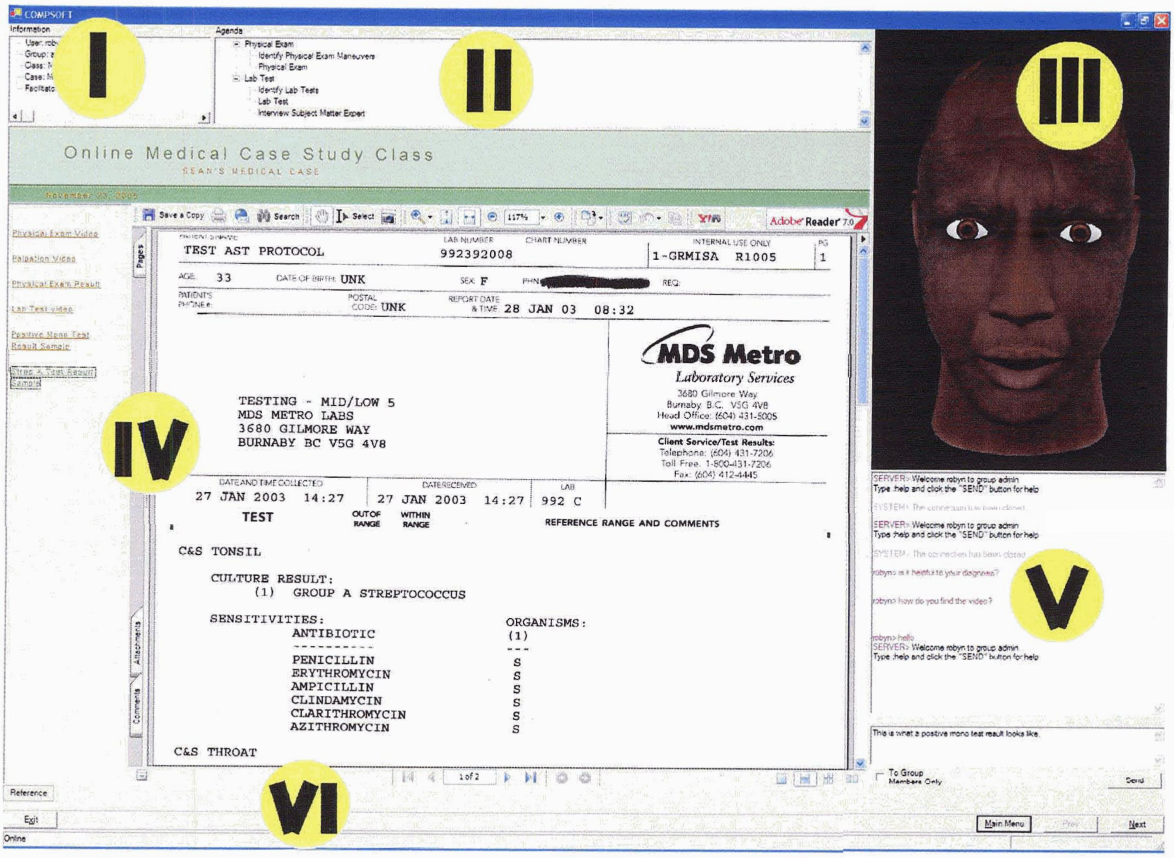
Physical Exam Video

robyn> how do you find the video?

robyn> hello
SERVER> Welcome robyn to group admin
Type :help and click the "SEND" button for help

is it helpful to your diagnosis?

Figure 4-9: PDF in COMPSOFT Client



PATIENT'S NAME: TEST AST PROTOCOL
LAB NUMBER: 992392008
CHART NUMBER:
INTERNAL USE ONLY: 1-GRMISA R1005
PG: 1

AGE: 33
DATE OF BIRTH: UNK
SEX: F
PHN: [redacted]
REQ:

PATIENT'S PHONE #:
POSTAL CODE: UNK
REPORT DATE & TIME: 28 JAN 03 08:32

TESTING - MID/LOW 5
MDS METRO LABS
3680 GILMORE WAY
BURNABY BC V5G 4V8

MDS Metro
Laboratory Services
3680 Gilmore Way
Burnaby B.C. V5G 4V8
Head Office: (604) 431-5005
www.mdsmetro.com

Client Service/Test Results:
Telephone: (604) 431-7206
Toll Free: 1-800-431-7206
Fax: (604) 412-4445

DATE AND TIME COLLECTED	DATE RECEIVED	LAB
27 JAN 2003 14:27	27 JAN 2003 14:27	992 C

TEST | OUT OF RANGE | WITHIN RANGE | REFERENCE RANGE AND COMMENTS

C&S TONSIL

CULTURE RESULT:
(1) GROUP A STREPTOCOCCUS

SENSITIVITIES:

ANTIBIOTIC	ORGANISMS: (1)
PENICILLIN	S
ERYTHROMYCIN	S
AMPICILLIN	S
CLINDAMYCIN	S
CLARITHROMYCIN	S
AZITHROMYCIN	S

C&S THROAT

SERVER> Welcome robyn to group admin
Type :help and click the "SEND" button for help
SYSTEM> The connection has been closed
SERVER> Welcome robyn to group admin
Type :help and click the "SEND" button for help
SYSTEM> The connection has been closed
robyn> is it helpful to your diagnosis?
robyn> how do you find the video?
robyn> hello
SERVER> Welcome robyn to group admin
Type :help and click the "SEND" button for help

This is what a positive mono test result looks like.

Figure 4-9 has labels for the parts of the COMPSOFT client UI. The labels correspond to the following features: I) the information panel, II) the agenda panel, III) the iFACE panel, IV) the browser panel, V) the chat panel, and VI) the menus and status bar.

4.4.5.1 Information panel

The information panel shows general information about the current PBL lesson such as the name of the class, the name of the case, the name of the student using the client, the name of the facilitator, the group the student belongs to, and the list of members in the group. The information presented here will be updated automatically.

4.4.5.2 Agenda panel

The agenda panel shows the phases and activities performed from the start of the case study until the current activity, known as the agenda. The information presented here will be updated automatically as the case study progresses.

4.4.5.3 iFACE panel

This is where the iFACE interface is located. From the screenshots, it is easy to see that the 3D model of the face and its emotional parameters can be changed depending on the PBL activities. A good example would be to compare the emotionally stable physician 3D face in Figure 4-9 and the worried and painful face of the virtual patient 3D face in Figure 4-7. The facial and emotional parameters are specified in the activity section of the case script.

4.4.5.4 Browser panel

This the browser part of the client UI. It is used to show the material associated with the current activities. From the figures, we can see that a variety of media can be presented in the browser panel including plain texts, videos, PDF documents, interactive games, and web tools like a BBS.

4.4.5.5 Chat panel

This is the panel where synchronous chat takes place. The chat panel is divided into three parts. The top window shows the text chat history; the bottom window is where users enter their chat messages or issue other chat commands; the bottom part has a checkbox for a user to specify whether to talk to his group members only or to everyone and a send button to send the message. The chat panel follows the keyboard shortcut of a popular online messenger, MSN Messenger (*MSN Messenger Version 7.5*, 2006), where pressing the Enter key will send a message and pressing Shift-Enter will insert a newline to the message. The text in the chat history window is colour coded to allow speed recognition (Shneiderman & Plaisant, 2004).

4.4.5.6 Menus and the status bar

The status bar is to report the current state of the application to users such as online, offline, and connecting.

There are two rows of menu bars. The bottom one is the main menu that provides functions for exiting the application, logging into a PBL lesson, and logging out. Extra functionalities such as jumping to a PBL phase, injecting a new discussion activity, ending the current PBL case, and starting a new one are available if the user has been authenticated as a facilitator. COMPSOFT supports multiple facilitators for a class. However, only one of them can facilitate the class at a given time because the iFACE can only be owned by a single user. This problem is solved by allowing tutors to sign in as auditors. They can participate in the discussions and all activities but the control of the flow of the case and the iFACE is only owned by one facilitator at a time. A tutor auditing a case can choose to become the facilitator at any time by clicking on a menu item. All the functionalities in the main menu can be reached by a single click or two to minimize the users' input actions (Shneiderman & Plaisant, 2004).

The upper menu bar is an adaptive menu that changes according to the current PBL activity. There may be multiple resources for a single PBL activity. Clicking on the buttons of the adaptive menu brings the user to those materials. For example, at the end of a PBL lesson, the facilitator can guide students to an activity where learning issues are assigned to students for self-directed studies. The material involved in this activity includes the instruction for the assignment, reference materials needed for the students to do their self studies, and the COMPSOFT BBS to post and discuss their findings. Three menu buttons will then be shown and students are allowed to switch back and forth between these materials to accomplish their tasks. These are specified in the case script and shown in the client UI.

The separation of the two menu bars follows the HCI principle of meaningful organization of menu items while the ordering of the main menu bar items are based on ease of use (right to left for the main menu bar buttons and top to bottom for the main menu). These two principles lead to lower error rates, faster thinking times, shorter response times and easer to learn for users (Shneiderman & Plaisant, 2004). Since the items in the main menu bar are shown as buttons, they are placed on the lower right hand corner of the UI instead of the upper left. This contributes to a balanced composition to the UI that makes it weigh more even as well as establishing the ideal diagonal visual flow for users who speak left-to-right languages (Tidwell, 2005). In order to maintain position consistency, a menu item is greyed out instead of removed when the action is not available (Shneiderman & Plaisant).

All the screenshots presented above are taken from a single PBL case study. From the figures, we can see that the client UI allows different kinds of media and web tools, such as the COMPSOFT BBS, to be shown in the browser panel. We can also see that the individual parts of the UI can be resized as needed by the users. This is a feature made for the convenience of the users. The users need not resize the UI parts if they find it confusing to manipulate them.

4.5 Fictitious Scenario

We have examined the functionalities of the components and the UIs of COMPSOFT. A fictitious scenario will demonstrate the way the pieces work together. When a course C and the tutors for the course are decided, the information is added to the school registration system to allow students to register. The school registration system asks the COMPSOFT authentication service to add the class and add the facilitators to the class. When students registered for the class through the school registration system, it informs the authentication service. The authentication service adds the students to the class and randomly divides the students into groups. The facilitators author the cases for the course using the authoring tool and the authoring tool informs the authentication service to add the cases to the course. Before the lesson starts, one of the tutors or some administrator uses the case server console and chat server console to start the case server and chat server for the PBL class. The web locations of the two servers are registered with the authentication service.

When it is time for the lesson, facilitator A launches the COMPSOFT client and logs in. The COMPSOFT client contacts the authentication service, authenticates facilitator A and returns to the client with a list of classes that are accessible by A. A chooses the class C. The client contacts the authentication service again with the new information and the authentication service returns the locations of the case and chat servers along with other information such as the name of the class, and the password to authenticate the case server to the COMPSOFT client. After that, the chat client of the COMPSOFT client connects to the chat server and the case client of the COMPSOFT client authenticates itself with the case server using the given password. When successful, the case server returns to the case client with the current case name, and other related information. The case client makes use of the class ID, the case name and the case server hostname to construct the URL for the case script. It downloads the case script and interprets it with the script engine. The case server then informs every connected client about the online status of facilitator A. Facilitator A's case client receives the online event and passes the event to the COMPSOFT client. The client then retrieves the material for the current activity and displays it. Since A is a facilitator, its IP address will be registered with the case server. A facilitator online event will be distributed by the case server to all connected clients. The students receiving the event will get the facilitator online event along with the IP address of the facilitator. Their iFACE instances in their COMPSOFT client will connect to the facilitator's iFACE instance so that the facilitator's voice will be sent to all connected students for animating the 3D face. The connection sequence of a student is very similar to that of the facilitator. The other facilitators also log in to the system but as auditors.

Now, the lesson is ready to be started. Facilitator A guides students through the case presentation activity. When he decides that it is time to move on, he clicks the "Next" menu item on his COMPSOFT client and the action is sent through the case client to the case server. The case server validates the action, updates the agenda of its own case script and distributes the event to every connected case client. Everyone's COMPSOFT client updates the agenda of its downloaded copy of the case script and proceeds to the next activity upon receiving the next event from the case server through the case client. Now facilitator B decides to host this activity and choose to become a facilitator by clicking on the "Facilitate the Current Case" from the main menu. The action propagates to the case server and distributes to each client. Now, every client notices that the facilitator has been changed from A to B. The process goes on and the lesson ends on the third activity in the second phase of the PBL case study. At this time, students are assigned to do self-directed studies for their learning issues.

During the break time, students perform research and log in to the system through the COMPSOFT client. They are allowed to review the material previously covered and post their findings to the BBS resource provided by the current activity (the third activity of the second phase).

When the next lesson starts, the same login procedures are executed for the tutors and the students. Based on the case script agenda downloaded from the case server, the case study will be continued from the third activity of the second phase.

The above scenario seems to be complicated because it shows the internal workings and interactions among different components of COMPSOFT. When we look at the user's point of view, it is very simple and straightforward.

4.6 Conclusion

We have introduced what COMPSOFT is, its components, interfaces, and how the pieces work together to deliver the functionalities of a typical PBL platform. The next section describes the technical details of COMPSOFT.

5 TECHNICAL DETAILS OF COMPSOFT

This chapter presents the technical design principles, the architecture, and the implementation details of each components of COMPSOFT.

5.1 Design Principle and Implementation Methodology

In the initial phase of development, the SAGE-SIM team was consulted to finalize the system requirements, a preliminary design of the functionalities, and the UI of the COMPSOFT client. The COMPSOFT system has been designed to be extensible, easy to maintain, evolvable, and secure.

Since COMPSOFT involved networking components, a programming platform that provided good network communication and security infrastructure is needed. In addition, it had to be functionality rich, easy to program, stable, and widely supported at present and in the near future. Further, it is expected to have a good integrated development environment (IDE). C# was chosen by this researcher as the programming language along with the Microsoft .NET platform (*Microsoft .NET Homepage*, 2006). There were several reasons for choosing C# with .NET over J2EE (*Java Platform, Enterprise Edition (Java EE)*, n.d.). First, J2EE was a standard that was implemented by many different vendors such as IBM, and many different Apache Software Foundation (ASF) (*The Apache Software Foundation*, n.d.) groups. It would be hard to predict which vendor would be accepted widely and be supported in the future. Second, Visual Studio, the IDE for C# and .NET, was widely accepted to be one of the best IDEs in the market.

Third, because C# was a newly developed language, it is cleaner and free of legacy constructs.

To start with, to prevent making mistakes in important design decisions that may arise later on, and to shorten the time for development, a fast and simple prototype of the COMPSOFT client was implemented at first without the networking functionalities. This exercise not only helped this researcher get familiar with C# and .NET, but also helped tremendously to recognize the pieces that were needed for the COMPSOFT platform to function properly and to understand some of the technical difficulties in implementing COMPSOFT. The prototype was demonstrated to the team to facilitate a revision to the design of COMPSOFT early in the development process.

The researcher then read extensively about the capabilities of the .NET platform to gain some experience on possible ways to separate the concerns of COMPSOFT into components, to identify suitable technologies for each component, and to decide interactions among the components. After that, the researcher designed the platform by breaking down concerns into components, designed the interface for interactions among the components, and then started implementing them one by one.

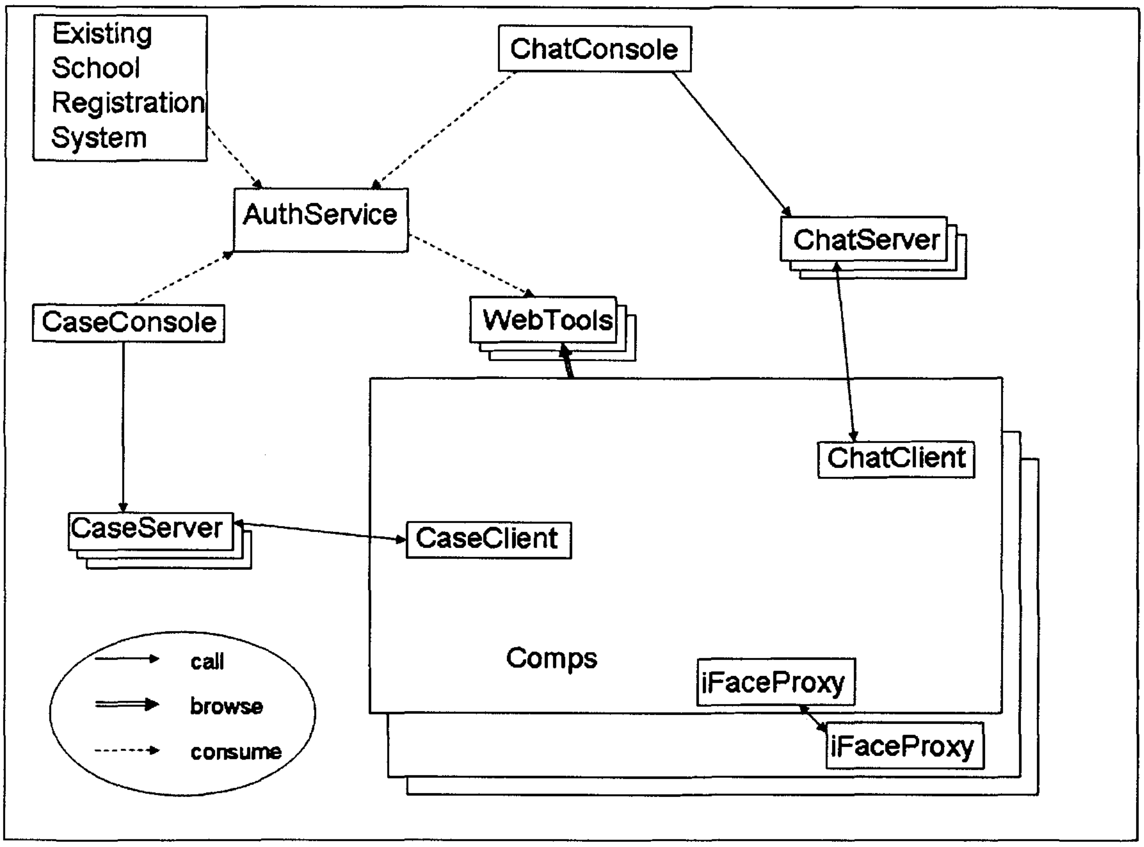
Because of the limited time and resources available, this researcher employed some principles from agile software development methodologies. For instance, instead of spending time to create long documentation before implementation, this researcher kept a developer journal to keep track of the development process and make design decisions. NDoc (*NDoc Online*, n.d.) was used to automatically generate MSDN (*MSDN Home Page*, 2006) style documentation from the source code and comments. The researcher also used a version controlling system called Subversion to manage the source code

(Collins-Sussman, Fitzpatrick, & Pilato, 2004). To ensure that the program was of highest quality and stable, a C# unit testing framework, NUnit, was used to test drive the development (Stott & Newkirk, 2004, April).

5.2 Architecture

For the current version of COMPSOFT, the main technical components are the COMPSOFT common library (`PBLCommon`), the database system (`Database`), the authentication service (`AuthService`), the scripting engine (`ScriptEngine`), the case client (`CaseClient`), the case server (`CaseServer`), the case server console (`CaseConsole`), the chat client (`ChatClient`), the chat server (`ChatServer`), the chat server console (`ChatConsole`), the iFACE proxy (`iFaceProxy`), the COMPSOFT client application (`Comps`), and the web tools (`WebTools`). Components not available in the current version are omitted from the discussion in this section. The architecture of COMPSOFT and the components relationship are shown in Figure 5-1. PBLCommon is not shown in the figure because it is used by all the components in COMPSOFT.

Figure 5-1: COMPSOFT Architecture



Each rectangle is a system/component boundary and each stacked rectangle means one or more instances of a system. In this figure, A "browses" B means B is launched in A's browser. A "calls" B with a unidirectional arrow means A sends network messages to or calls B's procedures remotely. A "calls" B means that the communication is in both directions. A "consumes" B means that B exposes its interface as a Web Service and A calls B's procedure through the exposed interface. Web Service is an open standard and the World Wide Web Consortium (W3C) (*World Wide Web Consortium*, n.d.) defined it as the "programmatic interface" available for "application to application communication" through the "World Wide Web" (*Web Services Activity*, n.d.).

5.3 Implementation Details

This section outlines the technical details of each subsystem in COMPSOFT.

5.3.1 PBLCommon

The most common utility functions used by other components are implemented in this .NET library. These simple classes handle logging, hashing, encryption and decryption, and managing XML configuration files. The `Log` class allows multiple log channels to be added in the form of `StreamWriter`'s. For example, a log can be written simultaneously to the console's standard output as well as a log file by adding a console standard output `StreamWriter` and a `FileStream` `StreamWriter` to the `Log` object. The XML configuration management class allows the handling of XML configuration files other than the .NET preset `app.config` and `web.config` files as well as providing more freedom to the confined rules of the `System.Configuration` workspace. The XML configuration management class is based on Paperless's `AnyConfig` class (Paperless, 2003).

5.3.2 AuthService and Database

In COMPSOFT, the term database refers to the database engine that stores the actual administration data for COMPSOFT. In addition, the term `Database` also refers to the library implemented to access and manage the data in the database engine. From now on, the database engine that stores data will be referred to as the database engine while the library accessing the database engine will be referred to as the `Database`. The researcher has chosen MySQL (*MySQL: The world's most popular open source database*, n.d.) to be the database engine to store administration data in COMPSOFT because it is free, fast, reliable and is one of the most popular open source database engines in the world. The database engine stores all the vital information used by the AuthService such as information for students, tutors, classes, cases, groups in classes, users' passwords, and students registration information. The database engine schema can be found in Appendix B. An administrator user account is inserted into the schema as a super user that the Database uses to access the database engine. Both students and facilitators are treated as users. A user is a facilitator if she belongs to the special group called admin in a class. This makes the system flexible because in this design a graduate student could be a facilitator in an undergraduate course and a student in a graduate course at the same time.

IAuthService is a C# interface that specifies the functionalities of the AuthService such as adding and removing classes, cases, students, facilitators, and registration functions. The complete interface is shown in Appendix C without the comments. Database implements the IAuthService interface. It manages the connection to the database engine and handles the business logic to perform the functionalities in IAuthService. Database accesses the database engine by the MySQL Connector/Net (*Download Connector/Net 1.0*, n.d.). The parameters for accessing the database engine can be configured by modifying the MySqlDatamodel.config XML file at runtime.

AuthService is implemented as a Web Service to allow smooth and simple integration with an existing/legacy school registration system. This allows the registration tasks to be automated. Web Service is built on industrial standards that allow seamless interoperability regardless of what the platform and the language the existing school registration system is implemented in. It implements and exposes the IAuthService interface by utilizing the Database library. This allows one to change the underlying database engine and the implementation without changing `AuthService` itself. Every administration related data is written to the database engine so that it does not need to maintain any state. Such a stateless nature makes it trivial to run multiple instances of `AuthService` and at the same time to provide good scalability and to save networking resources. It is also the first place a COMPSOFT client contacts. Therefore, it should be the most secure part of the entire platform. `AuthService` is managed by the Microsoft Internet Information Services (IIS) (*Internet Information Services*, 2006) which provides mechanisms to ensure the security of the Web Service. Secure Sockets Layer (SSL) is deployed to provide a secure channel for communications with `AuthService` (Humbad, 2005; Seely, 2002, August). IIS also provides lifetime management for `AuthService`. Therefore, there is no need to worry about restarting the Web Service when it is crashed. IIS will handle it automatically.

5.3.3 ScriptEngine

We have earlier discussed the functionalities of `ScriptEngine` and the structure of a case script. The structure of a case script is described by an XML Schema Definition (XSD) (*W3C XML Schema*, n.d.). The `ScriptEngine` validates a case script using the XSD when the case script is loaded, ensuring the integrity of the case script. The XSD for the case script is shown in Appendix D.

`ScriptEngine` can encrypt the activities for a PBL phase but doing that will break the XSD validation. This problem is solved by wrapping the encrypted activities by a `secret` tag and specifying in the XSD that an `activities` tag can be replaced by a `secret` tag.

5.3.4 Case Subsystem

The case subsystem includes the `CaseConsole`, the `CaseServer` and the `CaseClient`. The functionalities of these three components have been discussed earlier. Since the implementation of the `CaseConsole` is rather straightforward, the focus of this discussion will be on the technical details in the `CaseServer` and the `CaseClient`.

`CaseServer` and `CaseClient` communicate through .NET Remoting. .NET Remoting is the infrastructure implemented in the .NET framework to provide Object-Oriented Remote Procedure Call (OO-RPC) capabilities. It allows simple and fast interaction between `CaseClient`'s and `CaseServer`'s. Server can push information to the clients which is hard to achieve by a web application or Web Service that supports the pulling of information by the clients.

After reviewing code samples and articles in Remoting (Kimmel, 2005a, , 2005b; , *.NET Samples - How To: Remoting*, 2006), the researcher decided to implement the server to client communication channel as an event publishing and subscribing mechanism. This decouples the server and the client so that the server is not controlling the client directly but by raising an event that the client can consume. The client then performs the appropriate operations based on the consumed event. In addition, this mechanism promotes asynchronous processing that prevents the scenario where a long remote operation on a single client will hang up the server resulting in long delays for the propagation of the event to the other connected clients.

In addition to the events raised by the `CaseServer`, the `CaseClient` raises its own set of events for the UI to consume. This prevents the hanging of the UI when Remoting calls are made on a thread different from the thread creating the UI. It also decouples the

CaseClient from the UI and allows multiple types of UI such as a text-based UI, a graphical UI (GUI), or even a remote GUI, to run, simultaneously.

The only partially synchronous operations are connect, disconnect and other facilitator commands. For connect and disconnect, the client waits until the operation is finished or a timeout, which is configurable in an XML file, is reached. However, the server serves the request asynchronously when the server has a free thread in its thread pool. Therefore, the operation is only synchronous on the client side but not on the server side. For the facilitator commands such as jumping to a PBL phase, or inserting a new discussion activity, the CaseClient requests the CaseServer to raise an event to all clients and wait for it to finish. However, the event raising is done asynchronously where each event raising operation returns immediately without waiting for the clients to respond. This makes the facilitator commands partially synchronous.

Since the server and the client invoke events in an asynchronous manner by calling BeginInvoke on each attached event handler, exceptions will not be thrown when the connection is broken or when the server or client is crashed or terminated unexpectedly. This creates a huge problem in detecting lost connection. The way to fix this issue is to have the server and the client periodically pinging one another to detect connection lost. This is done by creating a special ping event for the other party to consume so that when the event is raised synchronously[6], an exception will be raised if the other side has been disconnected.

[6] This ping event is raised synchronously instead of asynchronously because exception needs to be raised when connection is lost.

The interface containing the list of events raised by the CaseServer and the CaseClient for its UI is shown in Appendix E. Appendix F shows the interface for the CaseServer, CaseClient and AdminCaseClient. The AdminCaseClient is the interface containing additional functionalities available only to facilitators.

5.3.5 Chat Subsystem

The chat subsystem includes the ChatConsole, the ChatServer and the ChatClient. The functionalities of these three components have been discussed earlier. Similar to the CaseConsole, the implementation of the ChatConsole is rather straightforward. Therefore, the focus of this discussion will be on the ChatServer and the ChatClient.

The implementation of the ChatServer and the ChatClient are based on Yang's picture chat (Yang, 2004). Rather than using Remoting, Socket is used here for network communication. Since most of the network traffic in the text chat[7] subsystem is text, Remoting's binary formatter marshalling will not help in performance. It also eliminates overhead from the Remoting infrastructure. In addition, using Socket makes the detection of lost connections easy. By manipulating the underlying streaming protocol ChatClient commands for different functionalities can be specified.

5.3.6 iFaceProxy

iFACE has been implemented by Dr. Ali Arya (Arya & DiPaola, 2004; Arya, DiPaola, Jefferies, & Enns, 2006; DiPaola & Arya, 2005, , n.d.) as a .NET UserControl, which can by embedded directly into a COMPSOFT client. However, this researcher

[7] Voice chat is not yet available in the current version.

decided to experiment with .NET Reflection to dynamically load the iFACE UserControl. This researcher did not have full access to iFACE and its code, and hence had no control over the bugs generated by iFACE. For instance, problems in iFACE may prevent the COMPSOFT client from running if iFACE is embedded directly into COMPSOFT. Loading iFACE dynamically into COMPSOFT instead of embeding it directly allows the COMPSOFT client to run without instantiating iFACE by replacing the 3D face with a static picture or the face of the facilitator by a digital camera. Users can specify whether they want the COMPSOFT client to load iFACE by checking an option through the COMPSOFT client UI or by modifying an XML configuration file.

To load iFACE dynamically, .NET Reflection is used to load the assembly[8] into memory, query the classes, methods and parameters, fields, and properties from the assembly. After that, an instance of required type is created by calling System.Activator.CreateInstance. To access a method in iFACE UserControl, the Invoke method of the MethodInfo object corresponding to the method has to be called with the following parameters: an instance of the iFACE UserControl type created by System.Activator.CreateInstance, a binding flag for the method invoked, the parameters for the method packaged by an array of the object class, and some other parameters that can be ignored. To simplify these processes and to make invoking methods for a dynamically loaded library transparent, the classes ProxyBase and iFaceProxy are created to handle the plumbing code and to store the MethodInfo's, FieldInfo's and PropertyInfo's into collections for use in calling a method. iFaceProxy provides a way to use the dynamically loaded iFACE UserControl as if it is

[8] ".NET programs are deployed as assemblies. An assembly is one or more EXEs or DLLs with associated metadata information." (Stiefel & Oberg, 2002)

statically linked. The `ProxyBase` class and part of the `iFaceProxy` class are shown in Appendix G.

5.3.7 Comps

COMPSOFT client (`Comps`) makes use of the client side of all the components and provides a UI to interact with them. Most of the source code in `Comps` is UI related. Embedding a `WebBrowser` control in `Comps` is just a matter of selecting it from the list of COM components in Visual Studio. As mentioned in the previous section, `Comps` is a consumer of the events raised by `CaseClient`. The design of `Comps`' UI and its `iFaceProxy` have been discussed in detail in the previous section and in the section about COMPSOFT client UI in the previous chapter.

5.3.8 WebTools

The `WebTools` in COMPSOFT includes the BBS and the discussion tool. The BBS is a modified version of a popular open source BBS, the phpBB (*phpBB.com :: Creating Communities*, n.d.). According to its official web site (*phpBB.com :: Creating Communities*):

> phpBB is a high powered, fully scalable, and highly customizable Open Source bulletin board package. phpBB has a user-friendly interface, simple and straightforward administration panel, and helpful FAQ. Based on the powerful PHP server language and your choice of MySQL, MS-SQL, PostgreSQL or Access/ODBC database servers, phpBB is the ideal free community solution for all web sites.

MySQL is set up as the backend storage for phpBB. The public release of phpBB is modified in roughly two aspects. First, instead of obtaining web form data only by the `POST` method, it has been modified to support the `GET` method too so that the data can be embedded in the web address. This provides a simple way for the browser control in Comps to login to the BBS with the same user ID and credentials as the current user of Comps. Second, phpBB code has been modified to release some of the restrictions of the system. For example, a user is able to edit a post created by another user so that the BBS can be served as a temporary substitute for the whiteboard when a whiteboard application is not available.

The discussion tool is implemented as a simple ASP.NET application. Its functionality has been discussed in an earlier section.

5.4 Deployment

Deployment issues related to the installation of COMPSOFT components, stability and scalability are discussed in this section.

5.4.1 Installation

The server parts of COMPSOFT include a MySQL database system, a Web Service, two consoles for launching an unknown number of servers, a BBS web application, and a discussion tool web application. Also, the case scripts, materials, and resources for case studies need to served by a web server for clients to download. In addition, there should be a way for users to obtain the COMPSOFT client and set it up easily. Furthermore, the XML configuration files for each component needs to be set up properly for the platform to function in a stable fashion. These make the deployment of the platform bothersome.

For this purpose, this researcher has created three set up projects in Visual Studio to automate set up and deployment of COMPSOFT. The ClientsSetup project prepares the directory structure of the COMPSOFT client EXEs, the DLLs, and the preconfigured XML configuration files, and package them into a Windows Installer set up file for easy installation of the COMPSOFT client. The ServersSetup project packages the chat server console, the chat server, the case server console and the case server. Administrator can thus run the set up executable to easily set up those two servers. The WebSetup project packages the authentication service, the discussion tool, the case scripts and resources, and the COMPSOFT client set up files into a structured directory. Administrator can extract them in a Windows Server machine and activate them with the IIS and get them running. Now, students and facilitators can download the client set up file from the web server and set up the clients in their own machines. The BBS, MySQL, and IIS have their own set up routines provided by their individual vendors[9].

5.4.2 Stability and Scalability

COMPSOFT's architecture is designed with stability in mind. To avoid a central point of failure, case servers, chat servers and authentication service all run independent of each other. Let us suppose that the authentication service crashes during a PBL lesson. Since the students and the facilitator have logged on, and they have established their connection with the case server and chat server, they can continue their lesson as usual. If at this time the chat server crashed, the case server and client are still running and the lesson can be continued with disabled chat functionalities. On the other hand, if the chat server and the case server for a class both crash, students and facilitators in other classes will not be affected. Since the central piece of the platform, the authentication service, is managed by IIS, its status will be monitored and the service will be restarted if it is

[9] COMPSOFT BBS uses the same set up routine as the phpBB. IIS is preinstalled in Windows 2003.

terminated unexpectedly. A terminated chat server or case server has to be restarted manually in the current version of COMPSOFT. In the future, monitoring and restarting of servers could be automated.

To make COMPSOFT scalable, each chat server or case server only handles a single class of students. When a machine is not able to handle the number of chat servers and case servers launched because of a large number classes and students, the chat servers and case servers can be run on multiple machines because they are independent of one another. The same applies for the authentication service. As long as they are connecting to the same database engine, remotely or locally, multiple instances of authentication services can be run on different machines because of its stateless nature. However, this is not likely needed because the resource needed to run a stateless Web Service is low and the authentication service is utilized sparingly at authentication times, server launching and terminating times, and registration times. The load balancing capability of the COMPSOFT servers makes the PBL platform highly scalable.

5.5 Conclusion

This section detailed the technical components and features of COMPSOFT. COMPSOFT is designed so that it would be extensible, easy for maintenance, evolvable, and secure Table 5.1 summarizes how these are achieved in COMPSOFT.

Table 5.1 Characteristics of COMPSOFT and the Proofs

Characteristics	Reasoning
Extensible	• COMPSOFT is data driven by the case script • New components can be added to extend COMPSOFT functionalities • New web tools can be implemented to be used by COMPSOFT client
Easy to maintain	• Registration is automated by authentication service • Easy configuration by XML files • Set up projects ease installation and deployment
Evolvable	• Individual component can be changed without affecting the whole system because each component has a clear system boundary and exposes a well-defined interface for interaction
Secure	• Authentication credentials are all hashed • SSL provides a secure communication channel • Every subsequent request to the case server require validation from the server • Case scripts are validated by an XSD at loading time

Last, but not the least, the design and development of COMPSOFT provided an opportunity for this researcher to master technical and conceptual knowledge in C#, .NET, PHP, ASP.NET, and Web Services.

6 USABILITY TESTING

6.1 Introduction

COMPSOFT is in its early phase of development. Although it is functional in most aspects, some of the components are still missing or incomplete. In order to examine the claims made based on the platform, it has to be deployed in a real-world PBL curriculum first. Then, a study has to be designed carefully to measure and analyse the platform's contributions to the PBL curriculum. Such a large-scale study is unrealistic at the current stage of development and is not within the scope set forth for this research. Since the main focus of this research has been identified as the design aspects of the PBL environment, a test of the usability of the system is in order. A diagnostic usability testing can be used to "find out what is working well ... and what is not working well, so that [we] can continue with what is working well and fix what is not working well" (*Usability: Methods for Designing Usable Web Sites*, n.d.).

The usability test involves three students and a facilitator working through an experimental case study. The experimental case study follows a real-world medical diagnosis cycle used in medical education. The case is shown in a series of videos that tell the story of the patient, Sean, starting from an argument with his girlfriend about his medical problem, until the point where diagnosis and treatment are administered. Appendix K shows the draft of Sean's case. Students are required to fill in two brief questionnaires before and after the case study. A focus group is also held after the case study to get additional feedback from the students.

6.2 Methods

This section outlines the methodology of the usability test for COMPSOFT.

6.2.1 Participants

The usability test included three students, a facilitator and a subject matter expert. The facilitator, a Master's Degree Candidate in Education and Technology, is one of the members in the SAGE-SIM team.

The subject matter expert, Dr. Francis Ho, is a practising physician. He is also the author/writer of the experimental case.

According to the pre-case study survey form, the three students were in their third or fourth year of their undergraduate studies. Two of them were from the Faculty of Applied Science and one was from the Faculty of Arts and Social Science. None of them was from the Faculty of Health Science because this usability test was performed during the month of the final examination period. Every student had taken online courses before and had been using the Internet for research frequently. However, none of them had used an audiographic conferencing system before. Two of the students claimed that they did not know their patient assessment skill level while the other student rated his/her assessment skill as above average.

6.2.2 Equipment

COMPSOFT and iFACE were installed in five computers in the Learning and Instructional Development Centre (LIDC) in the Burnaby campus of the Simon Fraser University (SFU). They were all connected to the Internet with a high network bandwidth connection. Since synchronous voice chat was not available in COMPSOFT, Skype

(*Skype - The whole world can talk for free*, 2006), a free Internet phone software, was installed on each machine to provide synchronous voice conferencing capability. The three students and the facilitator used four of the computers[10]. The extra one was prepared as a backup in case one of the computers failed at the time of the usability testing. All five computers were located in different rooms to provide a physically separated environment where no participant would be able to see, talk and interfere with another. A headphone and a microphone were given to each participant for use in the voice conferencing system and iFACE. The COMPSOFT servers, the database engine, the web tools, the case script and other resources were all set up in a computer in the Surrey campus of SFU, about 20 KMs away from the Burnaby campus site of the usability study. A handout showing the basic flow of the PBL case study was prepared and handed over to the facilitator for reference. Consent forms and questionnaires were also prepared.

6.2.3 Procedures

Before the case study began, the students were given a briefing about the usability test and their roles in the test. The students then signed a consent form, followed by filling in a pre-case study survey form. Likert scale (*Likert scale information page*, 1999) was used in the design of the pre-case and post-case study questionnaires. The pre-case study questionnaire and the post-case study questionnaire are shown in Appendix H and Appendix I, respectively.

After the briefing, the students, the facilitator and the subject matter expert went to their assigned computer rooms. Voice conference was already made available through

[10] The subject matter expert shared a single computer with the facilitator.

Skype. This researcher then introduced the student participants to the start up procedures of the COMPSOFT client and to the login procedures of the system through voice conferencing. After that, the researcher muted the voice channel in Skype and started speaking only through the 3D animated avatar face in COMPSOFT client. A warm up session began, and the students were invited to try out different functionalities provided by the COMPSOFT client and the web tools. The students were also given an opportunity to play a simple Flash game through the browser in the client as shown in Figure 4-5.

When the warm up session was over, the facilitator took the seat and started the case study. The case study consisted of eleven PBL phases: the warm up, the introduction, the prelude, the case presentation, the patient history, the physical exam, the lab test, the self-directed study, the differential diagnosis, the treatment and follow up, and the wrap up. In each phase of the case study, the students were involved in different activities such as discussions, viewing a video, and reading materials. The students also were provided with opportunities to practice their interviewing skills at the "interviewing Sean" activity in the patient history phase. During that activity, the subject matter expert *role-played* the patient by speaking through the patient 3D animated avatar face. This is shown in Figure 4-7. In addition, students also explored various techniques of physical exam by watching the videos shown in the patient history and in the physical exam phase as shown in Figure 4-8. After being presented with lab test results of the patient, students were given another opportunity to interview the subject matter expert about the meaning of the tests and other questions related to the diagnosis of the patient. At this time, instead of *role-playing* the patient, the subject matter expert *role-played* a physician named Dr. Pearce, as shown in Figure 4-9. At the self-directed study phase, the facilitator introduced the learning issue assignment to the students. Each student selected a topic for self-directed studies. The facilitator then logged off from the system. The students were given 75 minutes to perform their research and post their findings to the COMPSOFT BBS. When the time was over, the facilitator logged back into the system and the PBL session resumed where students reported their findings as shown in Figure 4-6. The case study continued until the wrap up phase. Appendix A shows the case script used by the usability study. A draft tutorial process outline for use by the facilitator is presented in Appendix J.

After the case study, all participants met at the LIDC seminar room for a focus group meeting where the students filled in a post-case study survey form, and discussed and provided additional feedback on their experience in using the COMPSOFT client for the case study.

6.3 Results and Discussion

This section presents and interprets the feedback given by the participants of the usability test.

6.3.1 Overall Experience

The aim of this usability test is to obtain students' feedback on what is working fine and what needs to be improved in COMPSOFT. Usability test is not a functionality test where each function of the platform is tried in order to find bugs. Therefore, only the basic functionalities are used and many of the functionalities, such as the case branching functionalities, activity injection, private chat functions, multiple facilitators, were not utilized.

Overall, the students agreed that the experience of the online case study was good (two strongly agreed and one agreed) and they all agreed that COMPSOFT was an appropriate technology for online case studies and was easy to use (two agreed and one strongly agreed). These results are very encouraging given the incompleteness of the platform, such as the lack of a whiteboard, and that not all of the useful functions were utilized in the test. Some software bugs persisted in the iFACE component at the time of the test, which will be discussed in the next section.

6.3.2 iFACE

Students' feedback on the iFACE was varied. When asked whether the 3D face had made the case study engaging, their ratings were strongly agreed, agreed and undecided. When asked whether the 3D face was helpful to the case study, their ratings were agreed, undecided and disagreed. Interestingly, the student who rated the engagement level of iFACE found it the least helpful to the case study while the student rated the engagement level of iFACE the lowest among the three found iFACE to be the most useful among their ratings.

Before we interpret students' response, it is essential to know the status of iFACE at the time of the usability test. First, iFACE was a pilot software and the voice module was not optimized. Voice data were sent uncompressed through the network, contributing to a delay of over five seconds for the facilitator's voice to reach the students. Second, the voice module of iFACE had a bug that caused it to leak memory[11]. Therefore, the students and the facilitator had to restart their COMPSOFT clients several times during the test because the computer had run out of memory. Finally, the animator module of

[11] This problem has been fixed by the developer of iFACE at the time of this writing.

iFACE had a bug that caused the 3D face to deform over time, making the face look "creepy" as described by one of the students in the focus group discussion.

Based on the above problems of iFACE at the time of the usability test, students' responses seem reasonable and can even be deemed satisfying because application crashes and long system response times are two of the most frequent frustrating experiences that cause people to consider their time wasted (Shneiderman & Plaisant, 2004). Their experiences in using the phone or other voice conferencing system have made them accustomed to the short delay (almost non-existent) in voice chat. Therefore, iFACE's five-second delay would be viewed as unbearable. Shneiderman and Plaisant (2004) have stated that people's acceptance to response time is affected by "their established expectations based on their past experiences of time required to complete a given task" (p. 463). In the focus group meeting, a student stated that if the bugs were fixed, it would be awesome and the other students concurred.

Below are some of the positive quotes from the students about iFACE that were observed in the focus group:

- "The face can be anyone is cool."
- "Without it, it would be harder to anticipate someone is there. It feels like a teacher is in place. It brings life to the case."
- "I like the 3D face more than the instructor's face from a digital camera."

Below are some of the negative quotes from the students about iFACE that were observed in the focus group:

- "Too many bugs."

- “It is not that useful.”

- “I prefer a digital camera of the instructor’s face.”

Since COMPSOFT client loads iFACE dynamically, it is possible to implement a video conferencing module that shows the face of the facilitator if it is needed. When asked about their experiences in interviewing the patient through iFACE, students generally agreed that the live patient interview is more realistic and can anticipate things not pre-scripted as in video. During the focus group discussion, students found that the expressions of the 3D face were not rich and realistic enough. This could be resolved by having artists or fine art students to spend sometime to adjust the emotional parameters and create richer animations for different expressions. These can then be saved and be reused for future case studies.

In conclusion, iFACE were rated as average in this usability test. However, the bugs and incompleteness of iFACE could have large impacts on its rating. Therefore, more usability tests are needed in the future to examine its true potential when the problems are fixed.

6.3.3 Multimedia

All students strongly agreed that the video case presentation was helpful for the case study and all of them also strongly agreed that the video had made the case study engaging. Furthermore, none of them preferred to have a text-only version of the case. This is a complete consensus on the value of the video playback capability of the browser panel in COMPSOFT. Below are quotes from students in the focus group:

- “Video is fun!”

- "Video? Definitely!"

6.3.4 Communication Medium

The students generally agreed that the synchronous text chat was helpful (one strongly agreed and two agreed). However, the students had given some insightful comments about the text chat in the focus group. The addition of smileys and emoticons as well as the flashing or blinking of the chat window for informing new chat messages are welcomed by the students as features to be added to the text chat. One of the students wanted the feature of auto-saving of chat history at the client computer. Currently, the chat history is only saved at the server. The facilitator can choose to review or distribute it to the students. However, when COMPSOFT client was restarted during the test because of memory leakage from iFACE, the current chat history at the chat history window of the chat client was lost. This had caused much trouble to the students and the experimenter had to print out the server chat history at the time of the test for students' references.

The voice conferencing was considered helpful with two voted on strong agreement. However, one student rated the helpfulness of the voice conferencing as undecided . This student explained that the lack of a way to control who owned the microphone caused people to speak at the same time, ruining the voice conferencing experience. This situation happened often between the facilitator and a student. This was mainly caused by iFACE's delay of propagating the facilitator's voice data. The facilitator spoke when the students stopped talking. However, the facilitator's voice would not reach the students until several seconds later. When the discussion had been silenced for a few seconds, a student decided to say something and the facilitator's voice arrived. This situation will be improved when iFACE optimizes its voice module. Nevertheless, this situation will become unavoidable when the number of students in a class increases. Therefore, a form of microphone passing or coordinating system, such as the hand raising system in Elluminate Live! (Elluminate Inc., n.d.), should be implemented.

Two students agreed that the BBS was helpful while the other was undecided on this issue. During the focus group, one student said that it was confusing to have both a text chat and a BBS. Another student responded that the BBS was more structured and was good for keeping things.

There were two possible explanations for the student's confusion. First, the whole case study, including the self-directed study phase, was conducted in one single day. Every student was connected to the system all the time through the case study. Every student was able to say something in the text chat and every other student could read and respond to it. However, in a real life scenario, when a student is undergoing self-directed study, other students may not be connected to the system. In this case, a discussion with the whole class or with the student's group members can only be done through the BBS. Another possible cause for the confusion stems from the lack of the web whiteboard in the current version of COMPSOFT. In the case study, the BBS was first intended to be used as the whiteboard substitute by having students editing a single post. Later, the facilitator found that it might be more convenient to use the text chat instead. However, the BBS was being used again due to the loss of chat history due to the many restarts of the COMPSOFT client. This also explained why the students were confused. The need for the COMPSOFT whiteboard was further reflected from students' requests for a central place to store the learning issues during the focus group discussion.

6.3.5 Others

There were some other items of interest that were discussed in the focus group related to the usability of COMPSOFT. First, the students found that the agenda panel was not useful and they wanted to be able to review previous materials by pressing the "Prev" button that was currently disabled and greyed out. In fact, during a synchronous PBL session, only the current phase and activity was important. The agenda becomes useful when it is not a scheduled class time when the students are performing self-directed studies or reviewing material previously covered. When a student connects to the system and attends a lesson, the case script is downloaded to the student's computer and encrypted by the scripting engine. As the student goes through the PBL phases during the lesson, the scripting engine decrypts those phases of activities from the case script, making them available for future review. Therefore, the student is free to revisit those activities and materials by choosing to connect to a class not in its scheduled class time. This is known as the offline mode of the COMPSOFT client. The student connected this way will have facilitator like functions such as jumping to a specific phase and activity, going to the next phase or activity, and going back to the previous activity (by pressing the "Prev" button), as long as those phases and activities have been covered in class. This function was implemented in the prototype of COMPSOFT but was disabled during the usability test. Once the COMPSOFT library tool and the offline mode are available, the students concerns would be solved.

Currently the main menu can be accessed by clicking the “Main Menu” button at the bottom of the COMPSOFT client. It will be moved to the top-left corner because students have been accustomed to finding the menu there and had expressed concerns about this issue.

Finally, the students praised COMPSOFT’s ability provided by the scripting engine to show the videos progressively as the PBL case was unfolded.

6.3.6 Conclusion

We have examined COMPSOFT to identify what is working well, what is not working well, and how to improve it. The overall result is satisfactory because the participants rated the platform highly even though there were bugs and missing features in the platform.

7 CONCLUSION

PBL, a student-centric education approach where learning occurs in the process of solving an ill-structured problem in a group setting, has been widely adopted by medical schools for years. It has been adopted recently by other faculties and even by elementary and high schools. Some of the key issues including the rationale and advantages of PBL, the challenges of implementing it, and how technologies are used to alleviate some of the challenges, were discussed. However, each of these technologies only helps in a particular area of a PBL curriculum. Although utilizing multiple technologies at the same time may solve the problem, new issues arise when the users need to master the operations of several systems at the same time. Interoperability becomes an even bigger problem because there are no practical ways for the distinct systems to communicate with one another. Therefore, the SAGE-SIM team decided to develop the COMPSOFT platform to support every aspect of a PBL curriculum.

COMPSOFT is a complete software platform consisting of a number of components. The design and functionalities of each component of COMPSOFT and how they cooperate with one another have been discussed in detail. The rationale of the design, the development process, the architecture, and the implementation details of COMPSOFT were also discussed in detail. COMPSOFT is a much larger platform and it is in an early stage of development.

This project examined the usability of the platform to identify what is working and what needs to be improved. The strengths, the weaknesses, and some of the ways to fix and improve COMPSOFT have been identified as a result of the usability test. The usability test results showed that:

- Participants found the online PBL experience satisfactory.
- Participants felt that COMPSOFT was an appropriate technology for online case studies.
- Even though iFACE was incomplete and was plagued by annoying software bugs, it was still rated as average by the participants.
- The participants commented that iFACE allowed interactive virtual patient interviews that were not possible with pre-scripted videos.
- The video case presentation capability was excellent and the platform's capability of showing the video clips progressively as the case unfolded was highly praised.
- Participants generally agreed that the synchronous and asynchronous collaboration media in COMPSOFT were helpful, although certain important communication tools such as the web whiteboard was missing at the time of the test.

Evidenced by the overall positive ratings of the platform in such an early phase of the development, it can be concluded that COMPSOFT is a successful platform for online PBL learning.

Future directions include completing the functionalities of COMPSOFT, enhancing COMPSOFT by making changes as discussed in the previous chapter, working with iFACE developer to fix the problems in iFACE, and perform more experiments on the platform.

APPENDICES

Appendix A: The Case Script used for the Usability Test

```
 <?xml version="1.0" encoding="utf-8" ?>
<script xmlns="https://sage/pbl"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://sage/pbl caseScript.xsd">
  <global>
    <homepage>https://comp01</homepage>
    <faceResources>
      <faceResource id="0">
        <mainFile local="true">
          <path>HiRes/hrFace-me.x</path>
        </mainFile>
      </faceResource>
      <faceResource id="1">
        <mainFile local="true">
          <path>HiRes/hrFace-fw.x</path>
        </mainFile>
      </faceResource>
      <faceResource id="2">
        <mainFile local="true">
          <path>HiRes/Behaviours/Sean.ptype</path>
        </mainFile>
      </faceResource>
      <faceResource id="3">
        <mainFile local="true">
          <path>HiRes/Behaviours/Default.ptype</path>
        </mainFile>
      </faceResource>
      <faceResource id="4">
        <mainFile local="true">
          <path>HiRes/hrFace-mb.x</path>
        </mainFile>
      </faceResource>
      <faceResource id="5">
        <mainFile local="true">
          <path>HiRes/hrFace-ma.x</path>
        </mainFile>
      </faceResource>
      <defaultResource id="1" />
    </faceResources>
  </global>
  <nodes>
    <node name="Tutorial">
      <activities>
        <activity id="0" name="Welcome">
          <chat enabled="true" />
          <faceId id="5" />
          <faceId id="3" />
          <tabs>
            <tab id="Welcome">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/stg1/act1/welcome.html</url>
            </tab>
            <tab id="Flash Game">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/rsrc/swf/chickenandeggs.swf</url>
            </tab>
            <tab id="Forum">
              <groupUrl groupName="group01" loginRequired="true">
                <url>http://elinc-
dev02.elinc.sfu.ca/forum/login.php?login&amp;autologin&amp;username={0}
&amp;password={1}</url>
              </groupUrl>
            </tab>
          </tabs>
          <activecontrol id="Welcome" />
        </activity>
      </activities>
      <nextCandidates>
        <noderef name="Introduction" />
      </nextCandidates>
    </node>
    <node name="Introduction">
      <activities>
        <activity id="0" name="Introduction">
          <chat enabled="true" />
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Introduction">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/stg1/act1/intro.html</url>
            </tab>
            <tab id="Objectives">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/rsrc/objective.html</url>
            </tab>
          </tabs>
          <activecontrol id="Introduction" />
        </activity>
      </activities>
      <nextCandidates>
        <noderef name="Sean and Kelly" />
      </nextCandidates>
    </node>
    <node name="Sean and Kelly">
      <activities>
        <activity id="0" name="Sean and Kelly">
          <chat enabled="true" />
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Prelude">
```

```
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/stg2/act1/prelude.html</url>
            </tab>
          </tabs>
          <activecontrol id="Prelude" />
        </activity>
        <activity id="1" name="Key Points">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Forum">
              <groupUrl groupName="group01" loginRequired="true">
                <url>http://elinc-
dev02.elinc.sfu.ca/forum/login.php?login&amp;autologin&amp;username={0}
&amp;password={1}&amp;redirect=viewtopic.php?p=5#5</url>
              </groupUrl>
            </tab>
          </tabs>
          <activecontrol id="Forum" />
        </activity>
      </activities>
      <nextCandidates>
        <noderef name="Case Presentation" />
      </nextCandidates>
    </node>
    <node name="Case Presentation">
      <activities>
        <activity id="0" name="Sean Describes his problems">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Presentation">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/stg3/act1/problem.html</url>
            </tab>
          </tabs>
          <activecontrol id="Presentation" />
        </activity>
        <activity id="1" name="Sean's Condition">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Forum">
              <groupUrl groupName="group01" loginRequired="true">
                <url>http://elinc-
dev02.elinc.sfu.ca/forum/login.php?login&amp;autologin&amp;username={0}
&amp;password={1}&amp;redirect=viewtopic.php?p=6#6</url>
              </groupUrl>
            </tab>
          </tabs>
          <activecontrol id="Forum" />
        </activity>
        <activity id="2" name="Risk Factors">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Forum">
```

```
              <groupUrl groupName="group01" loginRequired="true">
                <url>http://elinc-
dev02.elinc.sfu.ca/forum/login.php?login&amp;autologin&amp;username={0}
&amp;password={1}&amp;redirect=viewtopic.php?p=7#7</url>
              </groupUrl>
            </tab>
          </tabs>
          <activecontrol id="Forum" />
        </activity>
        <activity id="3" name="Differential Diagnosis">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Forum">
              <groupUrl groupName="group01" loginRequired="true">
                <url>http://elinc-
dev02.elinc.sfu.ca/forum/login.php?login&amp;autologin&amp;username={0}
&amp;password={1}&amp;redirect=viewtopic.php?p=8#8</url>
              </groupUrl>
            </tab>
          </tabs>
          <activecontrol id="Forum" />
        </activity>
        <activity id="4" name="Identify Learning Issues">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Learning Issues">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/stg3/act4/learningissue.html</url>
            </tab>
          </tabs>
          <activecontrol id="Learning Issues" />
        </activity>
      </activities>
      <nextCandidates>
        <noderef name="Patient History" />
      </nextCandidates>
    </node>
    <node name="Patient History">
      <activities>
        <activity id="0" name="Identify Interviewing Questions">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Discussion">
              <discussion>
                <topic>What questions would you like to ask the patient
about his present illness and past medical history?</topic>
                <information>http://elinc-
dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/stg4/act1/discussion.html</information>
              </discussion>
            </tab>
          </tabs>
          <activecontrol id="Discussion" />
        </activity>
        <activity id="1" name="Interview the Patient">
          <faceId id="0" />
          <faceId id="2" />
          <tabs>
            <tab id="Interview">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/stg4/act2/interview.html</url>
            </tab>
          </tabs>
          <activecontrol id="Interview" />
        </activity>
        <activity id="2" name="Patient Interview Demonstration">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Interview Demonstration">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/rsrc/interview.html</url>
            </tab>
          </tabs>
          <activecontrol id="Interview Demonstration" />
        </activity>
        <activity id="3" name="New Discovery">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Discussion">
              <discussion>
                <topic>What do you know now about Sean's risk factors
and history that might help you with the case?</topic>
                <information>http://elinc-
dev02.elinc.sfu.ca/Comps/Scripts/MED 102/sorethroat/rsrc/pic/Sean and
Kelly 3.jpg</information>
              </discussion>
            </tab>
          </tabs>
          <activecontrol id="Discussion" />
        </activity>
        <activity id="4" name="Differential Diagnosis">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Forum">
              <groupUrl groupName="group01" loginRequired="true">
                <url>http://elinc-
dev02.elinc.sfu.ca/forum/login.php?login&amp;autologin&amp;username={0}
&amp;password={1}&amp;redirect=viewtopic.php?p=8#8</url>
              </groupUrl>
            </tab>
          </tabs>
          <activecontrol id="Forum" />
        </activity>
      </activities>
      <nextCandidates>
        <noderef name="Physical Exam" />
      </nextCandidates>
    </node>
    <node name="Physical Exam">
      <activities>
        <activity id="0" name="Identify Physical Exam Maneuvers">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Discussion">
              <discussion>
                <topic>What physical exam maneuvers will you perform
and why?</topic>
                <information>http://elinc-
dev02.elinc.sfu.ca/Comps/Scripts/MED 102/sorethroat/rsrc/pic/Sean and
Kelly 1.jpg</information>
              </discussion>
            </tab>
          </tabs>
          <activecontrol id="Discussion" />
        </activity>
        <activity id="1" name="Physical Exam">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Physical Exam">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/rsrc/physexam.html</url>
            </tab>
          </tabs>
          <activecontrol id="Physical Exam" />
        </activity>
      </activities>
      <nextCandidates>
        <noderef name="Lab Test" />
      </nextCandidates>
    </node>
    <node name="Lab Test">
      <activities>
        <activity id="0" name="Identify Lab Tests">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Discussion">
              <discussion>
                <topic>What lab tests will you perform and why?</topic>
                <information>http://elinc-
dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/rsrc/physresult.html</information>
              </discussion>
            </tab>
          </tabs>
          <activecontrol id="Discussion" />
        </activity>
        <activity id="1" name="Lab Test">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Lab Test">
```

```
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/rsrc/labtest.html</url>
            </tab>
          </tabs>
          <activecontrol id="Lab Test" />
        </activity>
        <activity id="2" name="Interview Subject Matter Expert">
          <faceId id="4" />
          <faceId id="3" />
          <tabs>
            <tab id="Reference">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/stg6/act4/ref.html</url>
            </tab>
          </tabs>
          <activecontrol id="Reference" />
        </activity>
      </activities>
      <nextCandidates>
        <noderef name="Learning Issue Assignment" />
      </nextCandidates>
    </node>
    <node name="Learning Issue Assignment">
      <activities>
        <activity id="0" name="Research">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Instruction">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/stg7/act1/instruction.html</url>
            </tab>
            <tab id="Web Resources">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/rsrc/library.html</url>
            </tab>
            <tab id="Forum">
              <groupUrl groupName="group01" loginRequired="true">
                <url>http://elinc-
dev02.elinc.sfu.ca/forum/login.php?login&amp;autologin&amp;username={0}
&amp;password={1}&amp;redirect=viewtopic.php?t=2</url>
              </groupUrl>
            </tab>
          </tabs>
          <activecontrol id="Instruction" />
        </activity>
        <activity id="1" name="Revisit Objectives">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Objectives">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/rsrc/objective.html</url>
            </tab>
          </tabs>
          <activecontrol id="Objectives" />
        </activity>
        <activity id="2" name="Students' Presentations">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Forum">
              <groupUrl groupName="group01" loginRequired="true">
                <url>http://elinc-
dev02.elinc.sfu.ca/forum/login.php?login&amp;autologin&amp;username={0}
&amp;password={1}&amp;redirect=viewtopic.php?t=2</url>
              </groupUrl>
            </tab>
          </tabs>
          <activecontrol id="Forum" />
        </activity>
      </activities>
      <nextCandidates>
        <noderef name="Differetial Diagnosis" />
      </nextCandidates>
    </node>
    <node name="Differetial Diagnosis">
      <activities>
        <activity id="0" name="Differential Diagnosis">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Forum">
              <groupUrl groupName="group01" loginRequired="true">
                <url>http://elinc-
dev02.elinc.sfu.ca/forum/login.php?login&amp;autologin&amp;username={0}
&amp;password={1}&amp;redirect=viewtopic.php?t=3</url>
              </groupUrl>
            </tab>
          </tabs>
          <activecontrol id="Forum" />
        </activity>
      </activities>
      <nextCandidates>
        <noderef name="Treatment and Follow Up" />
      </nextCandidates>
    </node>
    <node name="Treatment and Follow Up">
      <activities>
        <activity id="0" name="Treatment Plan">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="Treatment Plan">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/stg9/act1/treatment.html</url>
            </tab>
          </tabs>
          <activecontrol id="Treatment Plan" />
        </activity>
      </activities>
      <nextCandidates>
        <noderef name="Wrap Up" />
      </nextCandidates>
    </node>
    <node name="Wrap Up">
      <activities>
        <activity id="0" name="Wrap Up">
          <faceId id="1" />
          <faceId id="3" />
          <tabs>
            <tab id="End">
              <url>http://elinc-dev02.elinc.sfu.ca/Comps/Scripts/MED
102/sorethroat/stg10/act1/end.html</url>
            </tab>
          </tabs>
          <activecontrol id="End" />
        </activity>
      </activities>
    </node>
    <agenda>
      <agendaItem name="Tutorial" activity="0" />
    </agenda>
  </nodes>
</script>
```

Appendix B: Schema for the Database Engine

```
-- Create the user table
CREATE TABLE usertbl (
  id VARCHAR(50) NOT NULL,
  passwd CHAR(40) NOT NULL,
  fullname VARCHAR(100) NOT NULL,
  email VARCHAR(100),
  PRIMARY KEY(id)
) ENGINE=InnoDB;

-- Create the class table
CREATE TABLE class (
  id VARCHAR(50) NOT NULL,
  name VARCHAR(100) NOT NULL,
  chathost VARCHAR(255),
  chatport INT,
  classhost VARCHAR(255),
  classport INT,
  PRIMARY KEY(id)
) ENGINE=InnoDB;

-- Create the classgroup table
CREATE TABLE classgroup (
  classId VARCHAR(50) NOT NULL,
  groupName VARCHAR(100) NOT NULL,
  PRIMARY KEY(classId, groupName),
  INDEX (classId),
  FOREIGN KEY(classId) REFERENCES class(id)
    ON DELETE CASCADE
) ENGINE=InnoDB;

-- Create the classcase table
CREATE TABLE classcase (
  classId VARCHAR(50) NOT NULL,
  caseName VARCHAR(100) NOT NULL,
  startTime DATETIME NOT NULL,
  ended BIT NOT NULL DEFAULT 0,
  PRIMARY KEY(classId, caseName),
  INDEX (classId),
  FOREIGN KEY(classId) REFERENCES class(id)
    ON DELETE CASCADE
) ENGINE=InnoDB;

-- Create the userclass table
-- Program needs to control the groupName behaviour
-- because it's too complicated to do it by the database
CREATE TABLE userclass (
  userId VARCHAR(50) NOT NULL,
  classId VARCHAR(50) NOT NULL,
  groupName VARCHAR(100) NOT NULL,
  PRIMARY KEY(userId, classId),
  INDEX (userId),
  FOREIGN KEY(userId) REFERENCES usertbl(id)
    ON DELETE CASCADE,
  INDEX (classId),
  FOREIGN KEY(classId) REFERENCES class(id)
    ON DELETE CASCADE,
  INDEX (classId, groupName),
  FOREIGN KEY(classId, groupName) REFERENCES classgroup(classId,
groupName)
      ON DELETE CASCADE
) ENGINE=InnoDB;

-- Create the admin user
INSERT INTO usertbl (id, passwd, fullname)
VALUES ('admin', SHA1('stupid'), 'Admin user');
```

Appendix C: Interface IAuthService

```
public interface IAuthService
{
  bool AddUser(string adminPasswd, string userId, string passwd, string
fullname, string email, out string reason);
  bool RemoveUser(string adminPasswd, string userId, out string
reason);
  bool AddUserToClass(string adminPasswd, string userId, string
classId, string groupName, out string reason);
  bool RemoveUserFromClass(string adminPasswd, string userId, string
classId, out string reason);
  bool AddGroupToClass(string adminPasswd, string classId, string
groupName, out string reason);
  bool RemoveGroupFromClass(string adminPasswd, string classId, string
groupName, out string reason);
  bool AddCase(string adminPasswd, string caseName, string classId,
string datetime, int ended, out string reason);
  bool RemoveCase(string adminPasswd, string classId, string caseName,
out string reason);
  bool AddClass(string adminPasswd, string classId, string className,
out string reason);
```

```
  bool RemoveClass(string adminPasswd, string classId, out string
reason);
  bool Authenticate(string userId, string passwd, out StringCollection
classList, out string reason);
  bool GetAuthenticationDto(string userId, string passwd, string
classId, out AuthenticationDto authDto, out string reason);
  bool ChangePasswd(string userId, string oldPasswd, string newPasswd,
out string reason);
  bool ListAllClasses(string adminPasswd, out StringCollection
classList, out string reason);
  bool RegisterChatServer(string adminPasswd, string classId, string
host, int port, bool force, out string reason);
  bool UnregisterChatServer(string adminPasswd, string classId, out
string reason);
  bool RegisterCaseServer(string adminPasswd, string classId, string
host, int port, bool force, out string reason);
  bool UnregisterCaseServer(string adminPasswd, string classId, out
string reason);
  bool RetrieveMostRecentCase(string adminPasswd, string classId, out
string caseName, out DateTime startTime, out string reason);
  bool RegisterEndCase(string adminPasswd, string classId, string
caseName, out string reason);
}
```

Appendix D: XSD for Case Scripts

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema targetNamespace="https://sage/pbl"
elementFormDefault="qualified" xmlns="https://sage/pbl"
      xmlns:xs="http://www.w3.org/2001/XMLSchema">
      <!-- face resource types -->
      <xs:complexType name="resourcePathType">
            <xs:sequence>
                  <xs:element name="path" type="xs:string" />
            </xs:sequence>
            <xs:attribute name="local" type="xs:boolean" use="required"
/>
      </xs:complexType>
      <xs:complexType name="faceResourceType">
            <xs:sequence>
                  <!-- the main file loaded by iFace -->
                  <xs:element name="mainFile" type="resourcePathType"
/>
                  <!-- the required files for loading the main file
above -->
                  <xs:element name="supportFile"
type="resourcePathType" minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" use="required" />
      </xs:complexType>
      <xs:complexType name="faceIdType">
            <xs:attribute name="id" type="xs:string" use="required" />
      </xs:complexType>
      <xs:complexType name="faceResourcesType">
            <xs:sequence>
            <xs:element name="faceResource"
type="faceResourceType" maxOccurs="unbounded" />
            <xs:element name="defaultResource" type="faceIdType"
/>
        </xs:sequence>
    </xs:complexType>
    <!-- types for global section -->
    <xs:complexType name="globalType">
        <xs:sequence>
            <xs:element name="homepage" type="xs:string" />
            <xs:element name="faceResources"
type="faceResourcesType" />
        </xs:sequence>
    </xs:complexType>
    <!-- types for nodes section -->
    <xs:complexType name="discussionType">
        <xs:sequence>
            <xs:element name="topic" type="xs:string" />
            <xs:element name="information" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="groupResourceType">
        <xs:sequence>
            <xs:element name="url" type="xs:string" />
        </xs:sequence>
        <xs:attribute name="groupName" type="xs:string"
use="required" />
        <xs:attribute name="loginRequired" type="xs:boolean"
use="required" />
    </xs:complexType>
    <xs:complexType name="tabType">
        <xs:sequence>
            <xs:choice>
                <xs:element name="discussion"
type="discussionType" />
                <xs:sequence>
                    <xs:element name="groupUrl"
type="groupResourceType" maxOccurs="unbounded" />
                </xs:sequence>
                <xs:element name="url" type="xs:string" />
            </xs:choice>
        </xs:sequence>
        <xs:attribute name="id" type="xs:string" use="required" />
    </xs:complexType>
    <xs:complexType name="tabsType">
        <xs:sequence>
            <xs:element name="tab" type="tabType"
maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="chatType">
        <xs:attribute name="enabled" type="xs:boolean"
use="required" />
    </xs:complexType>
    <xs:complexType name="activecontrolType">
        <xs:attribute name="id" type="xs:string" use="required" />
    </xs:complexType>
```

```
        <xs:complexType name="activityType">
                <xs:sequence>
                        <xs:element name="chat" type="chatType" minOccurs="0"
/>
                        <xs:element name="faceId" type="faceIdType"
minOccurs="0" maxOccurs="unbounded" />
                        <xs:element name="tabs" type="tabsType" />
                        <xs:element name="activecontrol"
type="activecontrolType" />
                </xs:sequence>
                <xs:attribute name="id" type="xs:integer" use="required" />
                <xs:attribute name="name" type="xs:string" use="required"
/>
        </xs:complexType>
        <xs:complexType name="activitiesType">
                <xs:sequence>
                        <xs:element name="activity" type="activityType"
maxOccurs="unbounded" />
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="noderefType">
                <xs:attribute name="name" type="xs:string" />
        </xs:complexType>
        <xs:complexType name="nextCandidatesType">
                <xs:sequence>
                        <xs:element name="noderef" type="noderefType"
maxOccurs="unbounded" />
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="nodeType">
                <xs:sequence>
                        <xs:choice>
                                <xs:sequence>
                                        <xs:element name="activities"
type="activitiesType" minOccurs="0" />
                                </xs:sequence>
                                <xs:element name="secret" />
                        </xs:choice>
                        <xs:element name="nextCandidates"
type="nextCandidatesType" minOccurs="0" />
                </xs:sequence>
                <xs:attribute name="name" type="xs:string" use="required"
/>
                <xs:attribute name="secret" type="xs:boolean" />
                <xs:attribute name="visited" type="xs:boolean" />
        </xs:complexType>
        <xs:complexType name="agendaItemType">
                <xs:attribute name="name" type="xs:string" />
                <xs:attribute name="activity" type="xs:integer" />
        </xs:complexType>
        <xs:complexType name="agendaType">
                <xs:sequence>
                        <xs:element name="agendaItem" type="agendaItemType"
maxOccurs="unbounded" />
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="nodesType">
```

```
            <xs:sequence>
                  <xs:element name="node" type="nodeType"
maxOccurs="unbounded" />
                  <xs:element name="agenda" type="agendaType" />
            </xs:sequence>
      </xs:complexType>
      <!-- types for script -->
      <xs:complexType name="scriptType">
            <xs:sequence>
                  <xs:element name="global" type="globalType" />
                  <xs:element name="nodes" type="nodesType" />
            </xs:sequence>
      </xs:complexType>
      <!-- definition of schema -->
      <xs:element name="script" type="scriptType" />
</xs:schema>
```

Appendix E: Events Raised by CaseServer and CaseClient

```
public interface ICaseServerEvents
{
  event VisitNodeEventHandler VisitNodeEvent;
  event EventHandler CaseEndedEvent;
  event EventHandler ServerShutdownEvent;
  event StartNewCaseEventHandler StartNewCaseEvent;
  event AddDiscussionEventHandler AddDiscussionEvent;
  event FacilitatorOnlineEventHandler FacilitatorOnlineEvent;
  event EventHandler FacilitatorOfflineEvent;
  event UserOnlineEventHandler UserOnlineEvent;
  event UserOfflineEventHandler UserOfflineEvent;
}

public interface ICaseClientEvents
{
  event VisitNodeEventHandler VisitNodeEvent;
  event EventHandler ServerShutdownEvent;
  event ErrorEventHandler ErrorEvent;
  event EventHandler CaseEndedEvent;
  event StartNewCaseEventHandler StartNewCaseEvent;
  event FacilitatorOnlineEventHandler FacilitatorOnlineEvent;
  event EventHandler FacilitatorOfflineEvent;
  event EventHandler DisconnectedByServerEvent;
  event UserOnlineEventHandler UserOnlineEvent;
  event UserOfflineEventHandler UserOfflineEvent;
}
```

Appendix F: Interfaces for the CaseServer, CaseClient and AdminCaseClient

```
public interface ICaseServer
{
  // public methods to be invoked by admin clients
  CaseResult InvokeVisitNode(string userId, int loginCode, string
nodeName, int activity);
```

```
  CaseResult InvokeEndCase(string userId, int loginCode);
  CaseResult InvokeStartNewCase(string userId, int loginCode);
  CaseResult InvokeAddDiscussion(string userId, int loginCode, string
topic, string refurl);
  CaseResult InvokeFacilitatorOnline(int loginCode, FacilitatorInfo
facilitatorInfo);

  // public methods to be called by clients
  CaseResult Disconnect(string userId);

  // Functions for communication with Case Console
  bool Start(int consoleCode, string classId, string logFolder, bool
logEnabled, string adminPasswd, string authServiceUrl);
  void Shutdown(int consoleCode);
  void ToggleLogState(int consoleCode);
  void Ping();
  bool Ping(string userId, ICaseClient client);
}

public interface ICaseClient
{
  // server event handlers
  void OnVisitNode(object sender, VisitNodeEventArgs e);
  void OnCaseEnded(object sender, EventArgs e);
  void OnStartNewCase(object sender, StartNewCaseEventArgs e);
  void OnServerShutdown(object sender, EventArgs e);
  void OnAddDiscussion(object sender, AddDiscussionEventArgs e);
  void OnFacilitatorOnline(object sender, FacilitatorOnlineEventArgs
e);
  void OnFacilitatorOffline(object sender, EventArgs e);
  void OnUserOnline(object sender, UserOnlineEventArgs e);
  void OnUserOffline(object sender, UserOfflineEventArgs e);

  // public methods
  CaseResult Connect(string clientHost, string host, int port, string
userId, string passwd, bool facilitator, string facehost, int
faceport);
  CaseResult Connect(string clientHost, string host, int port, string
userId, string passwd);
  CaseResult Disconnect(bool wait);
  void Shutdown();
  void Ping();
}

public interface ICaseAdminClient : ICaseClient
{
  // ICaseAdminClient events
  event AddDiscussionEventHandler AddDiscussionEvent;

  // public methods
  CaseResult VisitNode(string nodeName, int activity);
  CaseResult VisitNext();
  CaseResult EndCase();
  CaseResult StartNewCase();
  CaseResult AddDiscussion(string topic, string refurl);
  CaseResult FacilitateCase(FacilitatorInfo facilitatorInfo);
}
```

Appendix G: iFaceProxy and ProxyBase

```
public class ProxyBase
{
  [Flags]
  public enum TypeInfoFlag { None = 0, Methods = 1, Fields = 2,
Properties = 4 }

  protected object _obj;
  protected Type _type;
  protected MethodInfoCollection _methods;
  protected FieldInfoCollection _fields;
  protected PropertyInfoCollection _properties;

  public ProxyBase(string assemblyPath, string className, TypeInfoFlag
typeInfoAttr)
    : this(Assembly.LoadFrom(assemblyPath), className, typeInfoAttr)
  {
  }
  public ProxyBase(Assembly assembly, string className, TypeInfoFlag
typeInfoAttr)
  {
    // initialize the class's type, function, field and properties
references from the assembly
    InitTypeInfo(assembly.GetType(className), typeInfoAttr);

    // create the face instance
    _obj = Activator.CreateInstance(_type);
  }
  public ProxyBase(object obj, TypeInfoFlag typeInfoAttr)
  {
    _obj = obj;
    InitTypeInfo(obj.GetType(), typeInfoAttr);
  }
  private void InitTypeInfo(Type type, TypeInfoFlag typeInfoAttr)
  {
    // type
    _type = type;
    // method types
    _methods = new MethodInfoCollection();
    if ((typeInfoAttr & TypeInfoFlag.Methods) != TypeInfoFlag.None)
      _methods.AddRange(_type.GetMethods(BindingFlags.Public |
BindingFlags.Instance | BindingFlags.DeclaredOnly |
BindingFlags.InvokeMethod));
    // special treatment for clean up method
    MethodInfo mi = _type.GetMethod("Dispose", BindingFlags.Public |
BindingFlags.Instance | BindingFlags.InvokeMethod);
    if (mi != null)
      _methods.Add("Dispose0", mi);
    // face field types
    _fields = new FieldInfoCollection();
    if ((typeInfoAttr & TypeInfoFlag.Fields) != TypeInfoFlag.None)
      _fields.AddRange(_type.GetFields());
    // face property types
    _properties = new PropertyInfoCollection();
    if ((typeInfoAttr & TypeInfoFlag.Properties) != TypeInfoFlag.None)
```

```
      _properties.AddRange(_type.GetProperties());
  }
  public object Instance
  {
    get { return _obj; }
  }
  public object Invoke(string methodName, object[] args)
  {
    if (args != null)
      methodName += args.Length.ToString();
    else
      methodName += 0;
    return _methods[methodName].Invoke(_obj,
BindingFlags.Instance|BindingFlags.InvokeMethod|BindingFlags.Public,
null, args, null);
  }
  public object GetFieldValue(string fieldName)
  {
    return _fields[fieldName].GetValue(_obj);
  }
  public void SetFieldValue(string fieldName, object val)
  {
    _fields[fieldName].SetValue(_obj, val);
  }
  public object GetPropertyValue(string propertyName)
  {
    return _properties[propertyName].GetValue(_obj, null);
  }
  public void SetPropertyValue(string propertyName, object val)
  {
    _properties[propertyName].SetValue(_obj, val, null);
  }
}

public class iFaceProxy : ProxyBase
{
  const string FacePath = @"C:\iFace\Wrappers.dll";
  const string FaceClass = "iFace.Wrappers.Face3DControl";
  const string GeometryPath = @"C:\iFace\Geometry.dll";

  private AnimatorProxy _animator;
  private Type _animationTypeType;

  public iFaceProxy()
    : base(FacePath, FaceClass, TypeInfoFlag.Methods |
TypeInfoFlag.Fields | TypeInfoFlag.Properties)
  {
    // Since we'll be using some functions from the animator field,
    // we'd simply create another proxybase object for it.
    _animator = null;
    // The following is the argument type for the OpenAnimation method.
    _animationTypeType = null;
  }

  public bool Init(string chrFile, int frametime, Form form)
  {
    return (bool)Invoke("Init", new object[] { chrFile, frametime, form
});
  }

  public void Record()
  {
    Invoke("Record", null);
  }

  public void Play()
  {
    Invoke("Play", null);
  }

  // This method is a bit tricky because it involves a chain namespace
and enum values.
  // since the caller can't reference to the argument, we will change
it to a string and let
  // our proxy handle that.
  public void OpenAnimation(string animationType, string file)
  {
    if (_animationTypeType == null)
    {
      MethodInfo mi = _methods["OpenAnimation2"];
      ParameterInfo[] pi = mi.GetParameters();
      _animationTypeType = pi[0].ParameterType;
    }
    Invoke("OpenAnimation", new object[] {
Enum.Parse(_animationTypeType, animationType, true), file });
  }

  public void OpenGeometry(string fileX)
  {
    Invoke("OpenGeometry", new object[] { fileX } );
  }

  public float aspectRatio
  {
    get { return (float)GetFieldValue("aspectRatio"); }
    set { SetFieldValue("aspectRatio", value); }
  }

  public DockStyle Dock
  {
    set { SetPropertyValue("Dock", value); }
  }
      .
      .
      .
      .
}

public class MethodInfoCollection : Hashtable
{
  public MethodInfo this[string key]
  {
    get { return (MethodInfo)base[key]; }
  }
  public void AddRange(MethodInfo[] methodInfos)
  {
    foreach (MethodInfo mi in methodInfos)
    {
      ParameterInfo[] pi = mi.GetParameters();
      Add(mi.Name + pi.Length, mi);
    }
  }
}

public class FieldInfoCollection : Hashtable
{
  public FieldInfo this[string key]
  {
    get { return (FieldInfo)base[key];  }
  }
  public void AddRange(FieldInfo[] fieldInfos)
  {
    foreach (FieldInfo fi in fieldInfos)
    {
      Add(fi.Name, fi);
    }
  }
}


public class PropertyInfoCollection : Hashtable
{
  public PropertyInfo this[string key]
  {
    get { return (PropertyInfo)base[key]; }
  }
  public void AddRange(PropertyInfo[] propertyInfos)
  {
    foreach (PropertyInfo pi in propertyInfos)
    {
      Add(pi.Name, pi);
    }
  }
}
```

Appendix H: Pre Case Study Survey Form

COMPSOFT Pre Case Study Survey Form

I'm interested in your opinion on your previous online experience and your educational background. Your answers will be anonymous.

1) **Gender** □female □male

2) **SFU Faculty**
□Science
□ Applied Science
□Health Science
□Business Administration
□Arts & Social Science

3) **Year of Study**

4) **Computer Skill Self-Rating**
□ Excellent □ Very good □ Good □ Fair □Poor

5) **How many online courses have you taken before?**

6) **You use the Internet for research frequently.**
□ Strongly Disagree □ Disagree □ Undecided □Agree □ Strongly Agree

7) **Have you used Microsoft NetMeeting or some other audiographic conferencing system before?**
□ Yes □ No

8) **You are knowledgeable in patient assessment.**
□ Strongly Disagree □Disagree □Undecided □ Agree □ Strongly Agree

9) **You are experienced in patient assessment.**
□ Strongly Disagree □ Disagree □ Undecided □ Agree □ Strongly Agree

Appendix I: Post Case Study Survey Form

This survey will be distributed to students in person after the second session. In general we want to your opinion about the effectiveness of the various aspects of the online PBL tools.

Please rate how strongly you agree or disagree with each of the following statements by circling the appropriate number.

SD = Strongly Disagree, D = Disagree,
U = Undecided, A = Agree, SA = Strongly Agree

Item	SD	D	U	A	SA
1) The overall experience of this online case study was good.	1	2	3	4	5
2) COMPSOFT is easy to use.	1	2	3	4	5
3) COMPSOFT is an appropriate technology for online case studies.	1	2	3	4	5
4) The interactive 3D face was helpful for this case study.	1	2	3	4	5
5) The interactive 3D face made this case study engaging.	1	2	3	4	5
6) The video case presentation was helpful for this case study.	1	2	3	4	5
7) The video case presentation made this case study engaging.	1	2	3	4	5
8) The voice chat was helpful for this case study.	1	2	3	4	5
9) The forum was helpful for this case study.	1	2	3	4	5
10)The text chat was helpful for this case study.	1	2	3	4	5
11)This case study helped me improve my patient assessment skills.	1	2	3	4	5
12)This case study helped me improve my research skills.	1	2	3	4	5
13)I would prefer a text-only version of the case study.	1	2	3	4	5

Appendix J: Outline of the Tutorial Process for the Facilitator

Session 1 Tutorial Script

Facilitator	Students	Resource
Run "ice breaker" activity that gets everyone using various features such as text chat (Shift-Enter starts a newline instead of sending), posting, editing posts one-by-one and refreshing to view changes in the forum. Also explain to them unless directed by the facilitator, they don't need to click anything to get the case going. Get students to talk about themselves.	Resizing controls. Use text chat. Click on the forum button. Post reply to thread the practice thread in forum. Edit posts by others. Navigate the forum.	Try It!
Click next. Describe PBL (a medical case study that revealed to you in stages. At each stage we discuss what you know so far and what you need to know to solve the case. In between sessions you research the issues you identified in the session.)	Students comment or ask questions.	Diagnosis Cartoon
Ask students to click on Objectives button.	Students click on Objectives tab.	Objectives
Ask one student to start keeping a list of learning issues in Text Chat in CAPS.		
Click next. Show Scene 1 Sean and Kelly		Scene 1

Click next. Ask, do we know about Sean from this video that might help us with the case later on? Scribe please.	Scribe edits the post titled Key Points.	Post: Key Points
Click next. Show scene 2a.	View video.	Scene 2a Sean Presents His Problems.
Click next. Ask, "What is the key information you learned from this video about Sean's condition?" Ask scribe to edit the Sean's Condition post and the others to note on paper.	Students write notes at home that summarize these facts.	Post: Sean's Condition.
Click next. Describe "Risk Factors". Give an example. Ask, "What are the relevant of certain risk factors in Sean's case? (age, sex, ethnicity, lifestyle etc.)"	Each student comes up with an idea and asks the scribe to edit the post about risk factors.	Post: Risk Factors
Click next. Ask for new scribe. Ask: From what you know so far what kinds of hypotheses and rationales do you think account for Sean's problem particularly in relation to throat problems and rashes??	Scribe edits the post with various ideas.	Post: Differential Diagnosis

Click next. Bring up LEARNING ISSUES.	Students should be start identifying learning issues. Scribe types these in text chat in CAPS.	Screen Title: LEARNING ISSUES
Click next. Ask, "Are there any other questions you would ask the patient about his present illness and past medical history?" After 3 mins, tell the students to take turn to ask the patient questions until they all pass.	Students are given 3 - 5 mins to discuss possible questions.	Discussion screen with: discussion instructions Graphic Exam2.jpg
Click next. Dr Francis role plays the patient and answers students' questions.	Each student takes a turn to ask one question until they all pass.	Screen Title: Interviewing Sean
Click next. Introduce next video. Ask students to click on History of Present Illness to view Scene 2b Ask students to click on Past Medical History to view Scene 3	Students click on Scene 2. Then click on Scene 3 to view it.	Scene 2b History of Present Illness. Scene 3 Past Medical History
Click next. Ask, "What do you know now about Sean's risk factors and history now that might help you with the case?" Summarize: Let's sum up what we know so far. Ask	Students discuss.	Discussion screen with: Sean and Kelly 3 jpg

student to summarize.		
Click next. Here's our hypothesis. Add anything?		Post: Differential Diagnosis
Ask for New Scribe.	Student scribes.	
Click next. Say, "The next step is Physical Exam." Ask, what would you do in the Physical Exam and why? *Click next.*	Students say their ideas.	Discussion screen with: Sean and Kelly 1 jpg
Optional: **Ask the students to click and view palpation video, saying, "Here's a technique often used in a physical exam called palpating the cervical nodes. The doctor describes what she's looking for."**	**Students click on the Palpation video link.**	**Show second video of Palpating Cervical Nodes.**
Optional: **Ask, "What did she look for and why?"**	**Students briefly mention a few key points.**	
Ask the students to click and view Physical Exam Video.	Students click on the Physical Exam video link when the facilitator asks them to.	Physical Exam.
Click next. What lab tests would you do and why? (Blood tests, throat swab, Epstien Barr, mono, white cell count.)	Students suggest additional tests.	Discussion topic with Physical Exam results for reference

Click next. Pretend I'm Sean and explain to me in lay terms your current understanding of his problems and your plans for further investigation.	Students role-play this scenario.	Exam 2 jpg.
Click next. Ask students to click on Lab Test Video link to view the video Ask students to click and see a sample positive mono test result and a sample strep A test result.	Students click on Lab Test Video link to view Students click on mono test result link followed by Strep A test result link.	Video lab tests Positive mono test result Strep A test result
Click next. **Introduce the subject matter expert.** Ask the students to interview the Physician to gather more information. (10 – 15 minutes)	Students interview Dr Francis	Lab Test
Click next. Review learning issues. (**NOTE:** *Facilitator needs to ask students to click on the corresponding buttons at the bottom of the application to switch among Instruction, Web Resources and Forum respectively.*) Describe learning issue assignment. Identify your sources and be	Students follow the facilitator by clicking on the appropriate button at the bottom.	Instruction Screen Web Resources Screen Forum Topic: Learning Issue Assignment

able to explain why they considered this resource to be credible and what you found out. Post the url or any text to the forum and you can describe the points below.		
Evaluate the session. Write points.	Students list their points in TEXT CHAT.	
Thank students and remind them of next session.		Session 2 When?
END OF FIRST SESSION		

Session 2 Tutorial Process

Students present their research on the learning issues identified in the last session. Students identify their sources and explain why they consider this resource to be credible and tell us what they found out.

Facilitator	Student	Resource
Click next. Describe Objectives of this session (Learning issues, differential diagnosis and treatment plan.) Ask students for questions and comments.	Student comment or ask questions.	Screen: Objectives.
Click next. Refer to each student's post and click on links when needed.	Students present.	Forum Topic: Learning Issue Assignment
Click next.		Text Chat

What is your leading hypothesis in Sean's case and explain why. **Say**, "There are number of diseases that can account for sore throat and some of the symptoms Sean described. **Ask**, "How can you differentiate them?		
Click next. Show treatment and follow up information. Ask, "What issues would a health professional consider in designing this treatment plan?"	Students discuss.	Treatment Plan
Wrap up and thank students. Ask them to meet you for post test and/or focus group.		ER. jpg
END OF SECOND SESSION		

Appendix K: Sean's Case Draft

Scene One
(The Night Before)

Fade in.

As the scene opens we see a young man of about 18 years of age seated on a couch with his partner, a young woman of the same age. Their names are Sean and Kelly. They talk in what we will learn is Sean's parent's family room. Sean is visibly ill. He speaks in a voice barely above a whisper and appears pale. Their conversation is punctuated by Sean's dry cough and his near constant touching of his throat. While Kelly appears calm and concerned, Sean becomes agitated as he speaks his lines. Through their conversation we find out that Sean has had a sore throat, fever and neck pain for a week. He complains in a vehement manner and also mentions that he has developed an itchy rash on his ankle. Kelly, in an attempt to be consoling, mentions she had similar symptoms, but they only lasted a couple of days and tells Sean to consult a doctor. Sean, far from being comforted accuses Kelly of giving him an "illness", alluding to a possible S.T.D. As we fade to black we hear the sound of a face being slapped and Sean asking, "What'd ya do that for?"

Fade to black.

Scene Two
(Present History)

Fade in.

We see Sean being lead into a doctor's examination room by a nurse. He is told to have a seat and that the doctor will be with him shortly. The nurse leaves. Sean still appears to be quite ill and coughs a bit as he sits. There is a knock on the door and Dr. Pearce enters. He greets Sean and asks how he is feeling. As they converse Sean describes his illness. Sean looks directly into the camera as if the camera is Dr. Pearce. He mentions: he has been sick for a week, has a rash on his ankle, severe throat pain particularly when swallowing and that there has been some drooling. As Sean describes his throat pain there is quick cut to an animation from the camera's P.O.V. entering Sean's mouth and moving back till we see his inflamed throat. The doctor asks if Sean is a smoker. Sean appears slightly uncomfortable and looks away as he answers "no". There is a silence as the doctor looks at him in a questioning way. Sean confesses that he had been smoking marijuana before he became ill, then looks sheepish. The doctor replies flatly saying: "Oh yes I'm sure that's it."

Fade to black.

Scene Three
(Past History)

Fade in.

We are back in Dr. Pearce's examination room. He tells Sean that he needs to know a little more about his medical history. Sean says, he is generally in good health and very seldom gets sick, which is why he concerned about his present state. Dr. Pearce then runs through a list of chronic illnesses asking if Sean has ever suffered from any of them; asthma, bronchitis, pneumonia or diabetes. Sean answers in the negative. Then the doctor asks about other illnesses such as, allergies or H.I.V. Again Sean responds in the negative.

As Dr. Pearce goes through his list of chronic and other illnesses links to more information about the various maladies appear on the screen in a series of dissolve transitions. The viewers will be able to stop the presentation by clicking on one of links and then be taken to the appropriate online resource. After studying the resource, the viewer will be able to return to the programme where they left off.

Fade to black.

Scene Four
(Symptoms)

Fade in.

Once again we are in Dr. Pearce's examination room. As we fade up from black Dr. Pearce is explaining to Sean that he now wants to review the symptoms as presented. We cut to an angle close on Sean's face. He looks ill. The shot freezes on a close up. We dissolve to a half tone of Sean. The screen squeezes from right to left until Sean occupies the left half of the screen only. Dr. Pearce then lists the symptoms in voice over off screen. As he proceeds with this, the various symptoms come up as bullet points on the right half of the screen. We move relatively quickly through the points: "he is running a fever of 40 degrees C, he looks tired and is sleepy all the time, there is a dry cough and his speech is only slightly above a whisper, there is swelling about the neck and it is tender, there is loss of appetite and while there is mild shortness of breath there is no chest pain". In the final bullet point the doctor mentions that Sean's spit (phlegm) is a white watery mucous. We then dissolve to a series of still images displaying various types, colours and consistencies of sputum. The doctor describes each image. As he finishes we dissolve back to the list of bullet points.

Fade to black.

Scene Five
(The Physical Exam)

Fade in.

We open close on Sean's face. He looks unwell. The shot freezes and we dissolve to a two shot of Sean and the doctor sitting across from one another in the doctor's examination room. As the shot freezes and the dissolve begins, we hear the voice of Dr. Pearce off screen describing the various attributes of the patient found during the physical examination. Dr. Pearce then steps into the frame from screen left overlaying the two shot of he and Sean. The doctor speaks directly to the camera. He mentions Sean's general malaise, his 40 degree Celsius temperature and that his skin was warm to the touch and slick with perspiration. He moves on to talk about Sean's throat exam detailing that there were marked white exudates on the pharynx bilaterally. As Dr. Pearce is describing this we cut to a still photograph of the pharynx with exudates. An arrow moves across the photograph pointing out the exudates. We return to the shot of Dr. Pearce. As the scene continues he talks about the tender and swollen glands on both sides of Sean's neck. He says that this involves both the anterior and posterior cervical lymph nodes. As the doctor talks about the cervical nodes we cut to a brief segue showing and describing how to palpate the cervical nodes. (Includes position and methodology) Again we return to the shot of Dr. Pearce. He now documents Sean's blood pressure, pulse and respiratory rate. He mentions that Sean's chest is clear and his heart sounds normal. Following this Dr. Pearce starts to talk about Sean's abdominal exam. We cut to the abdominal in progress. As the exam proceeds we hear Pearce's voice off screen in sync with the on screen action. He says there is no abdomen distension and the liver is normal with no enlargement. He demonstrates palpation while mentioning that he can feel the tip of the spleen. Once again we return to the shot of Dr. Pearce as he concludes by saying that Sean has tenderness below the ribs and that his bowel sound was normal.

Fade out

Scene Six
(Tests)

Fade in.

We are in Dr. Pearce's office. Dr. Pearce is about to perform some tests on Sean. This will involve doing a rapid slide test for heterophile antibody and taking a throat swab. As Dr. Pearce prepares to and then performs these two tests he explains what he is doing to Sean and in turn to the audience. Should an additional more detailed explanation be required, the action will freeze as in scene five with the doctor stepping into the scene from off screen frame left to form a superimposition. At that point more detail can be added. The doctor would then step out of the frame and the action would resume.

After Dr. Pearce completes his explanation and testing he tells Sean that he must now go to the laboratory for blood work. We dissolve to a shot of Sean entering the lab. The action then freezes and as in scene four, the screen will squeeze Sean's image to the left. A list of tests (including: CBC, lytes, BUN, creatine, random blood sugar, EBV, mono screening, urinalysis, R&M and C&S) then appears on the right hand side of the screen. In voice over we hear Dr. Pearce saying the next step is to send Sean for the following tests. The doctor then lists the tests as they appear on the screen. As the list forms, each of the tests will have a link associated with it for more in depth information.

Fade to black.

Scene 7
(Investigation)

Fade in.

As this section will be more graphically based there will be heavy reliance on both synchronous and asynchronous collaboration. The word "investigation slowly forms on the screen. While the word is forming the voice of Dr. Pearce is heard off screen introducing this section. He says that based on the links mentioned in the last scene a general discussion of various forms of testing will follow.

The first test will be a throat culture. The phrase "throat culture" forms on the screen. Here we will see how a throat culture is processed in the lab. We will follow a lab. tech. through the various steps involved with a pause at the end for discussion. After that we will follow a similar sequence for the processing of a sputum culture.

Next the phrase "rapid test for Strep A" forms on the screen. From there we dissolve to a short video sequence of the same lab. tech. mentioned earlier going through the process of a Strep A test.

We then dissolve to a title reading "The Range". This will then dissolve to a list of titles appearing as individuals. (The individual titles being; electrolytes, BUN, creatinine and blood sugar.) Beneath each of the individual titles will be a link to a separate page where the students will find a listing of the normal ranges for the results of the individual tests. These individual titles are meant to be in essence a list of discussion points for use by the instructor.

Once this section is completed we dissolve to another title, EBV, which appears with a link below it to more information on the topic. The instructor may then lead a discussion on what EBV is.

Following this we dissolve to a title reading, "Mono Screening (heterophile antibody)". As with the previous title there will be a link to more information. The students may then discuss what mono screening (heterophile antibody) is.
We then dissolve to what will be the last title of this section, "The Chest X-Ray". This title will then dissolve to a normal chest x-ray. The students can then discuss what they see and the various features they should be looking for.

Fade out.

REFERENCE LIST

Adobe Systems Incorporated. (2006a). *Macromedia Shockwave Player.* Retrieved March 24, 2006, from http://www.macromedia.com/software/shockwaveplayer/

Adobe Systems Incorporated. (2006b). *Macromedia - Breeze.* Retrieved March 24, 2006, from http://www.macromedia.com/software/breeze/

Adobe Systems Incorporated. (2006c). *Macromedia - Flash Professional 8.* Retrieved April 8, 2006, from http://www.macromedia.com/software/flash/flashpro/

Albanese, M. A., & Mitchell, S. (1993). Problem-based learning: a review of literature on its outcomes and implementation issues. *Acad Med, 68*(1), 52-81.

Altman, M. (1997, April). *Electronic collaboration and PBL.* Paper presented at the Computers in Healthcare Education Symposium, Philadelphia, PA.

The Apache Software Foundation. (n.d.). Retrieved April 8, 2006, from The Apache Software Foundation: http://apache.org/

Armstrong, E. G. (1998). A hybrid model of problem-based learning. In D. Boud & G. Feletti (Eds.), *The Challenge of Problem-Based Learning* (2 ed., pp. 137-149). London, UK: Kogan Page.

Arya, A., & DiPaola, S. (2004). *Face as a Multimedia Object.* Paper presented at the 5th International Workshop on Image Analysis for Multimedia Interactive Services, Portugal.

Arya, A., DiPaola, S., Jefferies, L., & Enns, J. T. (2006). *Socially Communicative Characters for Interactive Applications.* Paper presented at the WSCG, University of West Bohemia, Campus Bory, Plzen, Czech Republic.

Bannon, L. J. (1989). Issues in Computer-Supported Collaborative Learning [Electronic Version]. *NATO Advanced Workshop on Computer-Supported Collaborative Learning* Retrieved March 22, 2006 from http://www.ul.ie/~idc/library/papersreports/LiamBannon/12/LBMarat.html.

Barrows, H. S. (1983). Problem-based, self-directed learning. *The Journal of the American Medical Association, 250*(22), 3077-3080.

Barrows, H. S. (1985). *How to Design a Problem-Based Curriculum for the Preclinical Years.* New York, NY: Springer.

Barrows, H. S. (1986). A Taxonomy of Problem-Based Learning Methods. *Medical Education, 20*(6), 481-486.

Barrows, H. S. (1994). *The Tutorial Process.* Springfield, IL: Southern Illinois University School of Medicine.

Barrows, H. S., & Tamblyn, R. M. (1980). *Problem-Based Learning: An Approach to Medical Education*. New York: Springer.

Boud, D. (Ed.). (1985). *Problem-Based Learning in Education for the Professions*. Sydney, Australia: Higher Education Research and Development Society of Australasia.

Boud, D., & Feletti, G. I. (Eds.). (1998). *The Challenge of Problem-Based Learning* (2 ed.). London, UK: Kogan Page.

Camp, G., PhD. (1996). Problem-Based Learning: A Paradigm Shift or a Passing Fad? *Medical Education Online, 1*(2). Retrieved March 10, 2006, from http://www.msu.edu/~dsolomon/f0000003.pdf.

Chickering, A. W., & Gamson, Z. F. (1991). *Applying the Seven Principles for Good Practice in Undergraduate Education*. San Francisco, CA: Jossey-Bass.

Coles, C. R. (1990). Evaluating the effects curricula have on student learning: toward a more competent theory for medical education. In Z. M. Nooman, H. G. Schmidt & E. S. Ezzat (Eds.), *Innovation in Medical Education: An Evaluation of Its Present Status*. New York, NY: Springer.

Collins-Sussman, B., Fitzpatrick, B. W., & Pilato, C. M. (2004). *Version Control with Subversion*: O'Reilly.

Colliver, J. A., PhD. (2000). Effectiveness of Problem-based Learning Curricula: Research and Theory. *Academic Medicine, 75*(3), 259-266.

Conway, J., Little, P., & McMillan, M. (2002). Congruence or conflict? Challenges in implementing problem-based learning across nursing cultures. *International Journal of Nursing Practice, 8*(5), 235-239.

Czujko, R. (1994). Physics Job Market: A statistical overview. *AAPT Announcer, 24*, 62.

de Tornyay, R. (1990). The Curriculum Revolution. *Journal of Nursing Education, 29*(7), 292-294.

Dean, S. J., Barratt, A. L., Hendry, G. D., & Lyon, P. M. A. (2003). Preparedness for hospital practice among graduates of a problem-based, graduate-entry medical program. *The Medical Journal of Australia, 178*(4), 163-166.

DeGrave, W. S., Volder, M. L., Gijselaers, W. H., & Damoiseaux, V. (1990). Peer teaching and problem-based learning: Tutor Characteristics, Tutor Functioning, Group Functioning, and Student Achievement. In Z. M. Nooman, H. G. Schmidt & E. S. Ezzat (Eds.), *Innovation in Medical Education: An Evaluation of Its Present Status* (pp. 123-134). New York, NY: Springer.

Dennen, V. P. (2000). Task Structuring for On-line Problem Based Learning: A Case Study. *Educational Technology & Society, 3*(3). Retrieved March 22, 2006, from http://ifets.ieee.org/periodical/vol_3_2000/d08.html.

Des Marchais, J. (1990). The involvement of teachers as problem-based learning tutors in the new Sherbrooke programme. *Annals of Community-Oriented Education, 3*, 35-54.

Des Marchais, J. E., Bureau, M. A., Dumais, B., & Pigeon, G. (1992). From traditional to problem-based learning: a case report of complete curriculum reform. *Medical Education, 26*(3), 190-199.

Dewey, J. (1909). *How we think.* Lexington, Massachusetts: D.C. Heath.

Diekelmann, N. (1995). Reawakening thinking: Is traditional pedagogy nearing completion? *Journal of Nursing Education, 34*(5), 195-196.

DiPaola, S., & Arya, A. (2005). *Socially Expressive Communication Agents: A Face-centric Approach.* Paper presented at the European Conference on Electronic Imaging and the Visual Arts, Florence.

DiPaola, S., & Arya, A. (n.d.). *iFACE: Interactive Face Animation - Comprehensive Environment.* Retrieved April 1, 2006, from http://ivizlab.sfu.ca/research/iface/

DiPaola, S., Chan, J., & Arya, A. (2005). *Simulating Face to Face Collaboration for Interactive Learning Systems.* Paper presented at the E-Learn Conference, Vancouver.

Disadvantages of Problem Based Learning. (1996). Retrieved March 15, 2006, from San Diego State Universiy DCDPBL Institute: http://edweb.sdsu.edu/clrit/learningresource/PBL/DisPBL.html

Distlehorst, L. H., & Robbs, R. S. (1998). A Comparison of Problem-Based Learning and Standard Curriculum Students: Three Years of Retrospective Data. *Teaching and Learning in Medicine, 10*(3), 131-137.

Dorsey, J. K., Gocey, J., Murrell, K., Rinderer-Rand, H., Hall, C., & Myers, J. H. (1996). Medical student response to an interactive patient simulation program used to supplement child abuse education. *Child Abuse & Neglect, 20*(10), 973-977.

Download Connector/Net 1.0. (n.d.). Retrieved April 8, 2006, from MySQL AB: http://dev.mysql.com/downloads/connector/net/1.0.html

Duch, B. J., Groh, S. E., & Allen, D. E. (Eds.). (2001). *The Power of Problem-Based Learning: A Practical "How To" for Teaching Undergraduate Courses in Any Discipline.* Sterling, VA: Stylus Publishing.

Edwards, J. C. (1990). The problem-based curriculum at Bowman Gray School of Medicine. *Academic Medicine, 65*(6), 363-364.

Elluminate Inc. (n.d.). *Live eLearning Solution & Collaboration Solution Software.* Retrieved March 26, 2004, from http://www.elluminate.com/

Engel, C. E. (1998). Not Just a Method But a Way of Learning. In D. Boud & G. Feletti (Eds.), *The Challenge of Problem-Based Learning* (2 ed.). London, UK: Kogan Page.

ESRI. (n.d.). *ArcView - The Geographic Information System for Everyone.* Retrieved March 24, 2006, from http://www.esri.com/software/arcview/

ETE Team. (n.d.). *ETE Teacher Pages.* Retrieved March 6, 2006, from Wheeling Jesuit University Center for Educational Technologies: http://www.cotf.edu/ete/teacher/introduction.html

Extensible Markup Language (XML). (n.d.). Retrieved April 8, 2006, from W3C: http://www.w3.org/XML/

Fincham, A. G., & Shuler, C. F. (2001). The changing face of dental education: the impact of PBL. *Journal of Dental Education, 65*(5), 406-421.

Finkle, S., & Torp, L. (1995). *Introductory Documents.* Aurora, Illinois: Illinois Math and Science Academy.

Finkle, S., & Torp, L. (1996). *Problems as Possibilities: Transforming Garbage to Gold.* Retrieved March 10, 2006, from Illinois Mathematics and Science Academy, Center for Problem-Based Learning: http://www.imsa.edu/programs/pbln/problems/garb/index.html

Finucane, P. M., Johnson, S. M., & Prideaux, D. J. (1998). Problem-based learning: its rationale and efficacy. *The Medical Journal of Australia, 168*, 445-448.

Foley, R. P., Polson, A. L., & Vance, J. M. (1997). Review of the literature on PBL in the clinical setting. *Teaching and Learning in Medicine, 9*, 4-9.

Grand'Maison, P., & Des Marchais, J. E. (1991). Preparing faculty to teach in a problem-based learning curriculum: the Sherbrooke experience. *Canadian Medical Association Journal, 144*(5), 557-562.

Hewett, Baecker, Card, Carey, Gasen, Mantei, et al. (1996). *ACM SIGCHI Curricula for Human-Computer Interaction: 2. Definition and Overview of Human-Computer Interaction.* Retrieved March 15, 2006, from ACM SIGCHI: http://www.sigchi.org/cdg/cdg2.html

Holmes, M. L. (2004, June). *Difficulties Implementing a Web Based PBL: Insight into Atomic Theory taught via a WebQuest.* Paper presented at the PBL 2004 International Conference, Cancun, Mexico.

Human-computer interaction - Wikipedia, the free encyclopedia. (n.d.). Retrieved March 23, 2006, from http://en.wikipedia.org/wiki/Human-computer_interaction

Humbad, S. N. (2005). Enabling SSL in IIS on Windows XP Professional [Electronic Version]. Retrieved April 8, 2006 from http://www.somacon.com/p41.php.

Internet Information Services. (2006). Retrieved April 8, 2006, from Microsoft Corporation: http://www.microsoft.com/WindowsServer2003/iis/default.mspx

Jackling, N., Lewis, J., Brandt, D., & Snell, R. (1990). Problem Solving in the Professions. *Higher Education Research and Development, 9*(2), 133-149.

Java Platform, Enterprise Edition (Java EE). (n.d.). Retrieved April 8, 2006, from Sun Microsystems: http://java.sun.com/javaee/

Jones, A., McArdle, P. J., & O'Neill, P. A. (2002). Perceptions of how well graduates are prepared for the role of pre-registration house officer: A comparison of outcomes from a traditional and integrated PBL curriculum. *Medical Education, 36*, 16-25.

Jones, D. (1996). What Is PBL? [Electronic Version]. Retrieved March 9, 2006 from http://edweb.sdsu.edu/clrit/learningresource/PBL/WhatisPBL.html.

Kamin, C., O'Sullivan, P., & Deterding, R. (2002, April). *Does Project L.I.V.E. Case Modality Impact Critical Thinking in PBL Groups?* Paper presented at the Annual Meeting of the American Educational Research Association, New Orleans, LA.

Kamin, C. S., Deterding, R. D., Wilson, B., Armacost, M., & Breedon, T. (1999). The development of a collaborative distance learning program to facilitate pediatric problem-based learning. *Medical Education Online, 4*(2). Retrieved March 24, 2006, from http://www.med-ed-online.org/t0000008.htm.

Kantrowicz, M., Kaufman, A., Mennin, S., Fulop, T., & Guilbert, J. J. (1987). *Innovative tracks at established institutions for the education of health personnel: An experimental approach to change relevant to health needs.* Geneva, Switzerland: World Health Organization.

Kaufman, A. (Ed.). (1985). *Implementing problem-based medical education: Lessons from successful innovations.* New York: Springer.

Kaufman, A., Mennin, S., Waterman, R., Duben, S., Hansburger, C., Silverblatt, H., et al. (1989). The New Mexico experiment: Educational innovation and institutional change. *Academic Medicine, 64*(6), 285-294.

Kaufman, D. (1995). Preparing Faculty as Tutors in Problem-Based Learning. In W. A. Wright & Associates (Eds.), *Teaching Improvement Practices: Successful Strategies for Higher Education* (pp. 101-125). Bolton, MA: Anker Publishing Company.

Kimmel, P. (2005a). .NET Remoting and Event Handling in VB .NET, Part 2 [Electronic Version]. *Developer.com.* Retrieved April 8, 2006 from http://www.developer.com/net/vb/article.php/10926_3487111_1.

Kimmel, P. (2005b). .NET Remoting and Event Handling in VB .NET, Part 3 [Electronic Version]. *Developer.com.* Retrieved April 8, 2006 from http://www.developer.com/net/vb/article.php/10926_3487776_1.

Knowles, M. S. (1988). *The Modern Practice of Adult Education: From Pedagogy to Andragogy.* Englewood Cliffs, NJ: Prentice Hall Regents.

Koschmann, T., Kelson, A. C., Feltovich, P. J., & Barrows, H. S. (1996). Computer-Supported Problem-Based Learning: A principled approach to the use of computers in collaborative learning. In T. Koschmann (Ed.), *CSCL: Theory and practice of an emerging paradigm* (pp. 83-124). Hillsdale, NJ: Lawrence Erlbaum Associates.

Krynock, K. B., & Robb, L. (1996). Is Problem-Based Learning a Problem for your Curriculum? *Illinois School Research and Development Journal, 33*(1), 21-24.

Laforcade, P., Barbier, F., Sallaberry, C., & Nodenot, T. (2003). *Profiling Co-operative Problem-Based Learning Situations.* Paper presented at the IEEE International Conference on Cognitive Informatics, London, UK.

Legacy Interactive. (2001). *ER-SIM: The Online Interactive Shockwave Medical Simulation Game.* Retrieved March 20, 2006, from http://www.ersim.com/main.html

Likert scale information page. (1999). Retrieved April 10, 2006, from Learning Technology Dissemination Initiative: http://www.icbl.hw.ac.uk/ltdi/cookbook/info_likert_scale/index.html

Lowry, S. (1992). What's wrong with medical education in Britain? *British Medical Journal, 305*(6864), 1277-1280.

Mann, K. V., & Kaufman, D. (1995). A response to the ACME-TRI Report: The Dalhousie problem-based learning curriculum. *Medical Education, 29*, 13-21.

McLoughlin, J. (2005). Co-ordinating and Managing PBL Programmes: Challenges and Strategies. In T. Barrett, I. Mac Labhrainn & H. Fallon (Eds.), *Handbook of Enquiry and Problem-based Learning Irish Case Studies and International Perspectives*. Dublin: Centre for Excellence in Learning and Teaching, NUI Galway and AISHE.

Microsoft. (2004). *NetMeeting Home*. Retrieved March 24, 2006, from http://www.microsoft.com/windows/netmeeting/

Microsoft. (2006). *Live Meeting Product Information*. Retrieved March 24, 2006, from http://www.microsoft.com/office/livemeeting/prodinfo/default.mspx

Microsoft .NET Homepage. (2006). Retrieved April 8, 2006, from Microsoft Corporation: http://www.microsoft.com/net/default.mspx

Moore, G. T. (1991). The effect of compulsory participation of medical students in problem-based learning. *Medical Education, 25*(2), 140-143.

MSDN Home Page. (2006). Retrieved April 8, 2006, from Microsoft Corporation: http://msdn1.microsoft.com/en-us/default.aspx

MSN Messenger Version 7.5. (2006). Retrieved April 8, 2006, from Microsoft Corporation: http://messenger.msn.com/Xp/Default.aspx

Muller, S., Chairman. (1984). Physicians for the Twenty-First Century: Report of the Project Panel on the General Professional Education of the Physician and College Preparation for Medicine. *Journal of Medical Education, 59*, Part 2.

MySQL: The world's most popular open source database. (n.d.). Retrieved April 8, 2006, from MySQL AB: http://www.mysql.com/

NDoc Online. (n.d.). Retrieved April 8, 2006, from http://ndoc.sourceforge.net/

.NET Samples - How To: Remoting. (2006). Retrieved April 8, 2006, from Microsoft Corporation: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpqstart/html/cpsmpnetsamples-howtoremoting.asp

NetDIVE Inc. (n.d.). *Web conferencing, Web meeting, Voice Conferencing, Online meeting, Application Sharing, Distant Learning*. Retrieved March 24, 2006, from http://www.netdive.com/indexea.htm

Neufeld, V. R., Woodward, C. A., & MacLeod, S. M. (1989). The McMaster M.D. program: A case study of renewal in medical education. *Academic Medicine, 64*, 423-432.

Norman, G. R., & Schmidt, H. G. (1992). The psychological basis of problem-based learning: a review of the evidence. *Academic Medicine, 67*(9), 557-565.

Paperless. (2003). *Handle multiple configuration files.* Retrieved April 8, 2006, from The Code Project: http://www.codeproject.com/csharp/anyconfig.asp

phpBB.com :: Creating Communities. (n.d.). Retrieved April 9, 2006, from phpBB Group: http://www.phpbb.com/

Pierce, J. W., & Gerdes, D. (2006). Self-Directed Learning of Children Engaged in Problem-Based Learning. *the Problem Log, 8,* 4-6.

Pixon Inc. (2006). *PictureTalk: Web conferencing, Web meeting, Online collaboration, Contact.* Retrieved March 24, 2006, from http://www.pixion.com/

Project Kaleidoscope. (1991). *What Works: Building Natural Science Communities.* Retrieved March 10, 2006, from http://www.pkal.org/documents/VolumeI.cfm.

Rhem, J. (1998). Problem-Based Learning: An Introduction. *The National Teaching & Learning Forum, 8*(1). Retrieved March 9, 2006, from http://www.ntlf.com/html/pi/9812/pbl_1.htm.

Rush, K. L., Ouellet, L. L., & Wasson, D. (1991). Faculty development: the essence of curriculum development. *Nurse Education Today, 11*(2), 121-126.

Ryan, K. (1992). Educational approach and curriculum design: a comparison of behavioural objectives and problem based learning in post graduate nursing education [Electronic Version]. *St Vincent's Hospital Nursing Monograph.* Retrieved March 10, 2006 from http://www.ciap.health.nsw.gov.au/hospolic/stvincents/1992/a07.html.

SAGE: Simulation and Advanced Gaming Environments for Learning. (n.d.). Retrieved April 10, 2006, from SAGE: http://www.sageforlearning.ca/

Savery, J. R., & Duffy, T. M. (1995). Problem Based Learning: An instructional model and its constructivist framework. *Educational Technology, 35*(5), 31-38.

Savin-Baden, M. (2000). *Problem-based Learning in Higher Education: Untold Stories.* Buckingham: Open University Press.

Schank, R. C., Berman, T. R., & Macpherson, K. A. (1999a). Learning by doing. In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory* (Vol. 2, pp. 161-181). Mahwah, NJ: Lawrence Erlbaum Associates.

Schank, R. C., Berman, T. R., & Macpherson, K. A. (1999b). Toward the development of flexibly adaptive instructional designs. In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory* (Vol. 2, pp. 183-213). Mahwah, NJ: Lawrence Erlbaum Associates.

Schell, R. (n.d.). Supporting Student Collaboration Online: VOIP - a Communications Option.

Schell, R., & Kaufman, D. (n.d.). The Case for Stories: Towards Developing Narrative-based Case Studies for Medical Education.

Schmidt, H., Henny, P., & de Vries, M. (1992). Comparing Problem-Based with Conventional Education. *Annals of Community Orientated Education, 5*, 193-198.

Schmidt, H. G. (1983). Problem-based learning: rationale and description. *Medical Education, 17*(1), 11-16.

Schmidt, H. G. (1998). Problem-based learning: does it prepare medical students to become better doctors? *The Medical Journal of Australia, 168*, 429-430.

Schmidt, H. G., Dauphinee, W. D., & Patel, V. L. (1987). Comparing the effects of problem-based and conventional curriculum in an international sample. *Medical Education, 62*, 305-315.

Schmidt, H. G., & van der Molen, H. T. (2001). Self-reported competency ratings of graduates of a problem-based medical curriculum. *Academic Medicine, 76*, 466-468.

Seely, S. (2002, August). HTTP Security and ASP.NET Web Services [Electronic Version]. *MSDN Library*. Retrieved April 8, 2006 from http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebsrv/html/httpsecurity.asp.

Shneiderman, B., & Plaisant, C. (2004). *Designing the user interface: strategies for effective human-computer interaction* (4 ed.): Addison Wesley.

Shor, I. (1992). *Empowering Education : Critical Teaching for Social Change*. Chicago, IL: University Of Chicago Press.

Simons, R., Smithies, A., Daniel, B., Gulati, S., & Hazel, P. (n.d.). *Problem Based Learning*. Retrieved March 15, 2006, from City University: http://www.paulhazel.com/docs/pbl.htm

Simulated Gaming for Medical Learning [Electronic (2005, June). Version]. *BCNETwork News*. Retrieved January 10, 2006 from http://www.bc.net/news_events_publications/newsletters/june-2005/simulated_gaming_for_medical.htm.

Skype - The whole world can talk for free. (2006). Retrieved April 10, 2006, from Skype Limited: http://www.skype.com/

Stiefel, M., & Oberg, R. J. (2002). *Application development using C# and .NET*. Upper Saddle River, NJ: Prentice Hall PTR.

Stott, W., & Newkirk, J. (2004, April). Test-Driven C#: Improve the Design and Flexibility of Your Project with Extreme Programming Techniques. *MSDN Magazine, 19*(4). Retrieved April 8, 2006, from http://msdn.microsoft.com/msdnmag/issues/04/04/ExtremeProgramming/default.aspx.

Swanson, A. G., & Anderson, M. B. (1993). Educating Medical Students: Assessing Change in Medical Education - the Road to Implementation (ACME-TRI Report). *Academic Medicine, 68*(6), Supplement.

Swenson, M. M., & Sims, S. L. (2000). Toward a narrative-centered curriculum for nurse practitioners. *Journal of Nursing Education, 39*(3), 109-115.

Tidwell, J. (2005). *Designing Interfaces: Patterns for Effective Interaction Design* (1 ed.): O'Reilly Media.

Tobias, S. (1992). *Revitalizing Undergraduate Science: Why Some Things Work and Most Don't.* Tucson, AZ: Research Corporation.

Tobias, S. (1994). *They're Not Dumb, They're Different: Stalking the Second Tier.* Tucson, AZ: Research Corporation.

University of California Division of Undergraduate Education. (2000). A Conversation with Rudi Berkelhamer. *SMET Newsletter, 2*(2). Retrieved March 20, 2006, from http://www.due.uci.edu/smet/newsletter/newsletter2b.html.

Usability: Methods for Designing Usable Web Sites. (n.d.). Retrieved April 10, 2006, from U.S. Department of Health and Human Services: http://www.usability.gov/methods/type_of_test.html

Vernon, D. T. A., & Blake, R. L. (1993). Does problem-based learning work? A meta-analysis of evaluative research. *Academic Medicine, 68*(7), 550-563.

W3C XML Schema. (n.d.). Retrieved April 8, 2006, from W3C: http://www.w3.org/XML/Schema

Watson, G. (2002). Using Technology to Promote Success in PBL Courses. *The Technology Source.* Retrieved March 22, 2006, from http://www.udel.edu/pbl/articles/Using-Technology-to-Promote-Success-in-PBL.pdf.

Web Services Activity. (n.d.). Retrieved April 8, 2006, from W3C: http://www.w3.org/2002/ws/

WebCT Inc. (n.d.). *WebCT.com.* Retrieved March 24, 2006, from http://www.webct.com/

WebEx Communications Inc. (n.d.). *WebEx: Web Conferencing, Online Meetings, and Video Conferencing.* Retrieved March 24, 2006, from http://www.webex.com/

Whipp, J. L., Ferguson, D. J., Wells, L. M., & Iacopino, A. M. (2000). Rethinking Knowledge and Pedagogy in Dental Education. *Journal of Dental Education, 64*(12), 860-866.

Wilkerson, L., & Feletti, G. (1991). Problem-based learning: one approach to increasing student participation. In A. F. Lucas (Ed.), *New Directions for Teaching and Learning: No.37. The Department Chairperson's Role in Enhancing College Teaching* (pp. 51-66). San Francisco, CA: Jossey-Bass.

Wingspread Conference. (1994). *Quality Assurance in Undergraduate Education: What the Public Expects.* Denver, CO: Education Commission of the States.

Winn, W. (1992). The assumptions of constructivism and instructional design. In T. M. Duffy & D. H. Jonassen (Eds.), *Constructivism and the technology of instruction: A conversation* (pp. 177-182). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Woodward, C. A. (1990). Monitoring an innovation in medical education: the McMaster experience. In Z. M. Nooman, H. G. Schmidt & E. S. Ezzat (Eds.), *Innovation in medical education : an evaluation of its present status* (pp. 27-39). New York, NY: Springer.

World Wide Web Consortium. (n.d.). Retrieved April 8, 2006, from W3C: http://www.w3.org/

Yang, K. W. (2004). *Picture Chat using C#.* Retrieved April 9, 2006, from The Code Project: http://www.thecodeproject.com/csharp/ykw1.asp

Yura, H., & Torres, G. (1986). Today's Conceptual Frameworks within Baccalaureate Nursing Programs. In *Faculty-Curriculum Development: Curriculum Design by Nursing Faculty.* New York, NY: National League for Nursing.