Journal of Universal Computer Science, vol. 2, no. 10 (1996), 679-693
submitted: 15/10/96, accepted: 15/10/96, appeared: 28/10/96 *© Springer Pub. Co.*

Computer-Supported Human Cooperation in Electronic Classrooms

Tomasz Müldner
(Jodrey School of Computer Science
Acadia University
Wolfville, N.S., Canada B0P 1X0
tomasz.muldner@acadiau.ca)

Robin A. Nicholl
(Jodrey School of Computer Science
Acadia University
Wolfville, N.S., Canada B0P 1X0
robin.nicholl@acadiau.ca)

Abstract: Recently, there has been growing excitement over the use of electronic classrooms. This paper provides a carefully balanced view of the advantages of using various technological facilities to improve teaching and learning, while recognizing the continuing need for a human teacher who coordinates and controls the learning process. In particular, our view is that technology should not be intrusive and should serve as a „means" towards specific educational goals, not as an „end" in itself. To illustrate this view we offer as a „case study" our vision of a course that uses electronic classrooms to assist teachers and students.
Category: K.3.2 [Computers and Education]: Computer and information science education; K.4.2 [Computing Milieux]: Computers and Society - Social issues; K.4.3 [Computers and Society]: Organizational Impacts
Key Words: Computer science education, case study, computer supported cooperative work, education, electronic classrooms, groupware

1 Introduction

With the increased availability of powerful personal computers and fast networks there is a growing trend towards learning processes that emphasize interaction and team work. *Groupware* or *computer-supported cooperative work* (CSCW) refers to computer-based systems that provide an interface to a shared environment in order to support groups of people working towards a common goal. [Grudin 94] presents some of the challenges faced by developers of these systems. [Norman 96] introduces some of the extensive research on using computing technology to help teachers be more effective.

Here we use the term „electronic classroom“ to mean a setting in which every student and the teacher has access to a personal computer (either desktop [Shneiderman 95], or notebook [Acadia 96], [Holmes 96]), and all computers are networked, allowing for various types of computer-supported interaction. We limit the scope of our discussion to „physical“ classrooms in which the teacher and students share the same room (i.e. we consider synchronous, face-to-face systems), rather than distributed, asynchronous, virtual classrooms as used for example in distance education. So far as areas of teaching are concerned, we do not limit ourselves to our own field (computer science). On the contrary, our arguments apply to learning or teaching any material that has a significant skill component; for example music, computing, engineering, mathematics, medicine, law, education, management, natural languages, sport, science (labs), counseling and the like.

It has been claimed that CSCW in electronic classrooms provides an „ideal“ teaching environment. However the authors of this paper feel that much too often a human factor, the role of the teacher, is not sufficiently appreciated. Therefore our paper will address not only technological novelties that support new types of learning and teaching, but also the educational reasons for using the electronic classroom. This is also the reason we have coined a new name for our work - namely computer-supported human cooperation (CSHC).

We are particularly interested in the following two issues: how the technology (computers) can help in education, and how teachers should respond to the new challenges of teaching in electronic classrooms. In this paper we focus on the first of these questions.

Yelon [Yelon 96] identified four common attributes of excellent teachers who received teaching awards and high evaluations from peers or students:
1) concern about subject matter
2) concern about students
3) dedication in teaching, and
4) putting into practice ten instructional principles (see Appendix).

Our paper presents a hypothetical course in which we set up an electronic classroom and decide on instructional strategies used to teach in this classroom. We start by introducing in [Section 2] some basic terminology. Since our goal is to improve the learning and teaching environment, we next present in [Section 3] various objectives for achieving this, using a hypothetical course as a „case study“ to make this more concrete. We describe methods to be used in our experiment, various observations resulting from our discussions with students and colleagues, and an evaluation of the results of the experiment. Finally, in [Section 4] we present our conclusions.

2 Basic Terminology and Assumptions

We consider only courses where students acquire mastery of one or more skills in addition to understanding the facts and theories of the subject matter.

Although much learning takes place outside classes (typically two-thirds or more of the time spent on a course), we tend not to consider activities that take place outside the (physical) electronic classroom. Also, we do not consider student preparations outside of classroom, such as reading, studying, researching or calculating. In this context we distinguish various types of interaction.

A *class* is a regularly scheduled meeting of students and teacher(s), arranged so that students can learn something from the presence of the teacher. (Students often learn from each other as well.) There are several types of class, such as lectures, tutorials and seminars.

In a *lecture* the teacher must present students with a coherent picture of some part of the subject matter. Communication flows primarily from teacher to student, complemented by student responses to indicate agreement or disagreement, belief or disbelief, understanding or misunderstanding. A skilled lecturer can communicate effectively with a large number of students at once. The lecturer is the focus of attention and must be able to control what occurs.

In a *tutorial* the teacher and students together increase their understanding of the subject matter by exercising their skills to apply what has been learned. Many different forms of interaction occur in different stages of a tutorial; they may even occur simultaneously. Students may collaborate to perform tasks with or without teacher collaboration, in response to comments from the teacher, or prior to the teacher's comments on their work. Tutorials are most effective when students can all participate and contribute, so it is important that the number of students be small.

A *seminar* is a class somewhere between a lecture and a tutorial. As in a lecture, there is some concrete subject matter, but as in a tutorial, the teacher and students work together. Here, a studio class, see [Holmes 96], will be considered a seminar rather than a lecture.

A *classroom* is used as a generic term, to mean a room to be used for a lecture of for a tutorial.

We recognize that many teachers find it valuable to offer lectures and tutorials during the same class, when class size and other circumstances permit, but we distinguish between lectures and tutorials because they require different types of computer support due to the different patterns of communication/interaction.

In terms of an operational scope, we consider a fairly conventional course for full-time students working for a university degree on-campus; students are assumed to be familiar with basic computing technology (if not, a preliminary tutorial is probably offered; see [Acadia 96]).

The proposed approach is meant to improve the following three skills: (1) learning; (2) retaining information; and (3) being able to retrieve information. We explain our approach using the example of a hypothetical course in programming, but our arguments equally apply to other disciplines. Although we do not describe any

concrete implementation, our approach is certainly „implementable“; and in [Section 3.5] we comment on several existing systems.

3 Case Study: Course

3.1 Objectives

In general terms, the main objective of any course is to produce educated individuals who are versatile, thoughtful, and able to cope with a changing environment. In this case study, our objective is to investigate and describe how computer technology can contribute to these broad objectives, specifically by creating a class environment that is conducive to helping students to learn faster and better.

We use Yelon's principles (see [6 Appendix]) to explain what we mean by „teaching better“. For each class, whether lecture or tutorial, it is essential that students arrive knowing what to expect in this class (*why* they will study a new idea and *what* this idea is about); they learn during the class; and they leave with an understanding of what new ideas they have learned, and how to organize these ideas so that they can be retrieved in the future. Therefore, in a way, each class consists of a *prolog* which reviews prerequisites for the class, the „body“ of the class, and the *epilog* which summarizes the class and helps to organize its contents and move concepts from a short-term memory to a long-term memory. In addition, the epilog briefly describes what will happen in the next class.

Regarding objectives for the body of the class, the first and most important is that students must be motivated: they must have clear reasons as to why they should study the topics in question. Also, students must be challenged by the goal of the class. This would help them to stay alert and remain interested in the outcome of the class. At the same time, teachers should avoid causing frustration over new and extraneous issues. Second, students must be stimulated by using varied instructional tools and techniques, so that they are not bored and tired by a monotonous class presentation. Next, the organization of the class must be clearly focused on the goal, and must not be distracted by discussing peripheral issues (for example topics related to the use of technological devices). In our opinion, it is absolutely essential that the teacher maintains close contact with the students. This contact can take the form of „eye contact“, questions, and other forms of interaction as well as active collaboration with students. This contact with the students must be open and free of any kind of intimidation, although the teacher must remain in control of the class. Finally, the entire class session should not usually be spent on „doing“ rather than listening; listening is one of the most important skills to be learned and practiced.

3.2 Nature of the Course

In general terms, the objectives for our hypothetical course are to teach students while addressing the issues set out in [Table 1].

Issue	*Reason for learning*	*Methods of learning*
General description	to motivate („why bother“)	description of why the idea is useful
Fundamental facts, including terminology, notation, etc.	to simplify learning and understanding of material and to allow effective communication with other people	listening, reading, and answering definition questions in quizzes, etc.
Standard theory or theories	to help gain a general point of view, in order to be able always to come up with a description applicable to large class of cases	generalizing facts and finding general patterns; reading, research, pedagogical examples and problems
Critical thinking and analysis	to verify students' understanding of the material and to evaluate it intelligently and expand it	group discussions, case studies, realistic problems and examples, quizzes, tests
Skill or skills	to give students an opportunity to use what has been learned, and to satisfy professional requirements	practice, observation, experiment, review and/or analysis, often under the direction and/or observation of teachers and/or other students.

Table 1: Issues to Address when Teaching a Course that Involves Skills

Specifically, our hypothetical course, as a representative skill, is a course on computer programming - an area where appropriate theories remain elusive. We can classify the topics to be taught as in the table above:
general description - the need for computer programming languages, and their specific constructs; the significance of the particular programming language used in the course
facts - the syntax, semantics and pragmatics of the programming language
theories - e.g. how to specify, design and test a computer program
critical thinking - e.g. how to inspect and/or explain a computer program
skills - e.g. how to design, code, specify, inspect, explain, test, modify, check a program, program component or interface; this includes individual skills, group participation and the use of tools (e.g. compiler, linker, editor, operating system shell).

3.3 Methods

Our methods described in this section include both traditional instructional strategies and novel technological tools. Clearly computer technology can be used to provide and support facts, data and information, and examples. It can also support feedback and evaluation, exercises, problems, demonstrations, help, challenge, and competition. Therefore, we start with a description of the physical environment of our

electronic classroom. Below, we will use the term „participant“ when we do not wish to distinguish between a teacher and a student.

3.3.1 General Description of the Electronic Classroom to be Used

Evidently, to be most effective, lectures require teachers who are stimulating speakers and knowledgeable in the subject matter as well as students who are good listeners, are motivated to learn, and already know something about the subject.

The physical setting of the electronic classroom must not be neglected because it can contribute to students' motivation, stimulation, and encouragement for active participation. First of all, the class size should never be larger than 40-50. Tutorial rooms should allow comfortable interactions within a group of no more than 10-20 students, allowing the focus to change easily from person to person, but also allowing the teacher to reclaim control when appropriate. These tutorials allow teachers to establish some personal contact with students and to reinforce and monitor what has been learned. Teachers must monitor student progress when they work independently (individually or in small groups); of course, this should not be done only electronically. For a studio classroom, which specifically supports groups of students, a classroom may have a „café-like“ setting, with say four students sitting around each table. Classrooms used for tutorials will generally require additional technology related to the specific subject and skill being taught (e.g. language laboratory, keyboard room, science or engineering lab). For example, „studio physics“ may require analog devices; and „studio music“ may require real, synthesized, or virtual instruments; computer science may require circuit boards.

Room facilities should include various multimedia devices, such as a sound system, microphones, audio and video recorders, and television sets. Electronic classrooms also need several large boards (including white-boards), screens and other display devices. Dynamic display devices - including computer displays, video players and televisions - are particularly useful to give substance to more abstract subject matter. Interactive display devices are important as they allow participants to mark up the images to highlight important characteristics.

The room layout should focus on the teacher rather than the technology, and therefore we are opposed to dark rooms with glaring screens. The teacher will need a console to control various devices, but must not be hidden behind it; on the contrary, the teacher's place should be a focal point of the console, see for example [Acadia 96].

Every participant has an individual computer (either a desktop or a notebook) with all the basic facilities for interaction, such as a screen and input devices, and access to facilities for permanent storage and for printing. The computers in the classroom must be interconnected, forming a local network, and this network may additionally be connected to Internet. Here, we don't concern ourselves with the details of physical implementation (for example whether or not each computer has a separate hard disk, or if there is a global storage device connected through a physical

network or through infrared links). It is desirable for teachers to be able to *restrict* interactions in the classroom. For example during lectures students' network access may have to be disabled so that the teacher can display the same image on all students screens at the same time.

While it is not relevant to us whether or not all students have the same type of a computer, we do assume that they use compatible software; for example, that all interactions occur as a result of the same procedure, executed on identical, or almost identical interface.

Effective classes require the active involvement of every participant. Besides the human participants, we consider another „participant": the electronic environment consisting of organizers, data repositories, etc. (for details see below). Therefore, we assume that the electronic classroom supports the following types of electronic communications and interactions:

- teacher accessing students (a single student, or broadcast to all students)
- access within groups of students
- student accessing teacher
- student or teacher accessing the environment (or vice versa)
- teacher taking over input devices of the selected student.

3.3.2 Shared Workspaces

Every participant's computer screen will have a designated area, called a „Shared Workspace" (SW), which works like a mail box, but is specifically designed to be used for classroom interactions. When a teacher submits data to a student, this student will be notified by the SW organizer. Access within a group of students is organized in a similar way. When a teacher wants to take control of a student's machine, the SW organizer will submit a request, and the student would have to approve it. The SW organizer will have various options available (for example, to automatically save any submissions).

The electronic environment is not exactly a part of an electronic classroom equipment, but it appears to be useful in saving, organizing, and retrieving information, and therefore we describe it briefly here. The SW organizer is an integrated system for dealing with a repository of information such as:

- specific Internet sites (for example information on mammals)
- local courseware
- any information that was broadcast
- correspondence (electronic mail) with other participants
- articles posted to News groups
- other useful information, including addresses (ftp, URLs, etc.), dates of tests, and so on.

The organizer allows the participant to maintain information, for example to update information automatically (e.g. by checking Internet sites for new information

or by checking new on-line notes from the teacher). Also, the organizer can be used to create (explicitly or implicitly) links between various types of information; for example links between a specific part of the courseware, information available on Internet, and notes. Finally, the organizer would provide a revision system, so that, for example, a participant could compare two versions of courseware notes or programming examples (code) to see what changes have been made.

It should be noted that the repository should have a user modifiable structure, which helps to organize and retrieve information. This structure may be just a tree-like structure so that users can identify topics, that may consist of sub-topics and pieces of information. As an alternative, this structure may be organized as a hypermedia structure, using a system such as HM-card, see [Mayrhofer 96]. In addition, the repository consists of nodes that are either fixed (e.g. a part of courseware), or modifiable (created by a student), and links that again are of the same two kinds.

Intelligent agents could help in organizing and retrieving relevant information. For example, a student working on Java may ask these agents to seek for any new information on Internet that is related to Java, assign a high priority to electronic mail on this topic, etc.

To support taking on-line notes, and capturing data and on-line notes, tutorial rooms should have several display devices that allow presenters (whether teacher or student) to show, for example previously stored examples or incomplete work being done during the tutorial, and to provide some form of shared work space in which the presenter can collaborate with other participants to make changes.

3.3.3 Specific Methods

This section describes methods used to achieve our objectives for a specific, one-term course on computer programming. For the sake of discussion, we assume that the Java language is studied. First, we describe specific methods for lectures and then methods for tutorials.

Students attend regular lectures, interleaved with tutorials. Both lectures and tutorials are given in electronic classrooms, as described above.

Before the beginning of a term, a complete courseware on Java is made available for students to install on their own computers, Students also have access to the SW organizer, including the repository which allows them to create personalized information. To help students acquire the necessary knowledge to use courseware and SW, a tutorial on these applications is given at the beginning of the course. In general, to avoid cognitive overhead all courseware used has the same interface and uses the same conventions.

Our basic assumption is that an electronic presentation (courseware) used in the class is readily available to students, who can review it at their own time and pace. We also assume that this presentation consists of two basic parts: one which can’t be

modified and another which can be customized. This assumption is based on our belief that there is no single type of learner; students learn in a variety of different ways and so should be provided with the means to customize class presentations to their learning needs and preferences. Therefore even the part that cannot be modified can still be annotated, and hyper-linked with other parts; perhaps from the repository described above.

A prolog of every class is created by the teacher and made available to students before the class. This prolog briefly recalls the most recently learned topics, and provides some short examples. Familiar examples from previous courses, from the text, and from lectures (especially those that students have previously discussed or studied) are used to relate new material to the students' current knowledge.

Similarly, an epilog of each class is made available before the class, possibly with additional annotations created by the teacher at the end of a class. This epilog lists topics covered in this class, provides references to these topics (examples, bibliography, etc.), and announces topics to be covered in the next class.

The body of the lecture consists of a presentation of courseware by the teacher. This presentation is shown on a large screen, but students can also see its copy on their personal computers. The lecture which merely concentrates on a presentation that is available off-line, would be boring and would not stimulate students. Therefore, it is expected that during the lecture the teacher makes verbal comments on the presented topics, and students may choose to record these comments in a form of margin notes or other types of annotations in their copies of the presentation (depending on facilities available, students will have to type these comments, or record parts of them through the classroom's audio system). Later, students can use the SW organizer to incorporate new information into their repositories. In addition, the teacher presents additional examples, some of which are created interactively with the help of students. Specifically, for our case study, the teacher would prepare examples of Java code. The teacher may choose to write the code which is complete and correct, or not quite complete and ask students to comment or to provide the missing part of the code. Another possibility is that the teacher poses a problem (for example, a specification of a programming problem) and asks students to design a solution, or even write the code which implements a solution to the given problem. Of course, students wouldn't have to manually copy anything that is provided by the instructor on-line; instead they can use the SW organizer to electronically copy such material.

It should be noted that no meta-issues related to technological aspects of the lecture should ever be mentioned during this lecture; otherwise, students would be distracted and wouldn't focus on the actual subject in question. As we mentioned above, students are supposed to learn how to use courseware and its environment before the beginning of the term; and if there is a need for any update, this need should be addressed by special workshops.

For each user, SW maintains a complete history of each lecture. Then, a specific user may wish to incorporate parts of this history in her or his repository. For

example, the teacher may decide to include some of the examples discussed in the recent lecture in a global repository available to all students, or store them in the courseware itself for use in future editions of this course.

In order to stimulate students, varied techniques and instructional tools will be used. First, students will be encouraged to actively participate in the lecture. There are many ways in which this objective can be accomplished. For example, the teacher will reinforce lecture material with immediate practice of skills; in our example, programming or design skills. The teacher may frequently interact with students by asking questions. To avoid intimidation, sometimes it may be best to use anonymous comments or questions. Most of this activity is rather informal and takes place outside the classroom - for example, during private study, in discussions with students and teachers, in libraries, in residences or on the Internet. This is made easier when class material is also available outside the classroom (hence some „data capture“ facility and some networking).

The tutorials are designed so that every student can also participate actively in a more formal setting, gain confidence and skill in expressing ideas, in asking questions, in listening to and understanding others, and thereby become a more active listener and a more active participant in lectures.

In our opinion, it is essential that the lecturer regularly teaches (at least) one of the tutorial classes; this has important benefits for the lecturer and for the students. (Even if circumstances dictate that some tutorials must be taught by other professors or by teaching assistants.) The lecturer needs to know what has actually been covered in all tutorials, so it is convenient if this is recorded „on-line“. Similarly „tutors“ need to know what was covered in lectures!

Tutorials provide the teachers with an opportunity to monitor closely how students are learning and hence allow remedial steps to be taken when necessary. Attendance at tutorials is required and suitable interaction is encouraged in tutorials (and in lectures). In particular, during tutorials students critique solutions that have been offered previously, perhaps by the teacher, or by other students. It is important to show, comment on and/or improve on work that has been done by students outside class.

3.4 Observations

These are the observations we made while „teaching“ our hypothetical course. Some of these might be considered to be „teething problems“ caused by our lack of prior experience in using electronic classrooms, while others show the difficulty in predicting the consequences for learning and for teaching of using electronic classrooms. We assess the significance of these observations in section 3.5.

Students varied widely in their previous experience with computing technology; this posed problems for the teacher in deciding what could be assumed and what must be taught.

In several classes students had to perform some tasks mechanically ("parrot-like") without understanding what they were doing; this was caused by lack of familiarity with the specific hardware and software. This was particularly true for classes at the start of term and at the start of a few individual classes later in the term.

Occasional failures of the (external) computer network caused serious disruptions of several classes. In this case the "backup" system is the (human) teacher, but without access to the prepared teaching materials.

Classrooms had to be set up at the start of each class; sometimes this was problematic. These problems contributed to tension in teachers, who were concerned that they would be perceived as incompetent teachers if they could not easily reset or reconnect some pieces of computer equipment. At times the technology distracted teachers from their teaching and added to their discomfort. True equipment failures compounded this problem, since teachers had a tendency to feel guilt if they could not get the equipment to work correctly.

We were surprised to discover that students actually found it *monotonous* when all notes were made available electronically; they also missed the opportunity to highlight things that they considered to be important but the professor (providing the notes) did not. On the other hand, students clearly felt freer to participate in class questions and discussion when they realized that records were being kept automatically and were available to them afterwards.

Attempts to create highly interactive lectures were generally unsuccessful. The number of students was simply too large for effective participatory work; there was too much interaction! For the same reason it was not usually practical to deal with anonymous questions during lectures; and dealing with them conscientiously after lectures also placed a great burden on teachers. It did seem that some students chose to ask questions that they really should have answered for themselves (with a little thought). However anonymous questions were a useful indicator to the teacher when several students expressed difficulty with the same topics.

Professors developed various techniques to improve the situation without cutting off interaction in lectures, but tended to place great importance on interaction in tutorials. Students learned to use tutorials to evaluate, refine and extend (i.e. analyse critically) solutions that had been presented in lectures, and to explain and defend their own solutions. This increased their confidence in dealing with the material and allowed them to exercise skills that had not been exercised previously.

Predictably, students responded in various ways to the changed teaching environment. Those students who felt comfortable in the new environment did become more active participants in class and this increased participation in the course extended to their out-of-class learning as well. Other students continued to participate reluctantly and/or infrequently and attempts to *force* them to participate led to considerable student discomfort. Finally there were a few students, perhaps the more

independent learners, who found the classroom interaction distracting and preferred to study the material away from the electronic classroom.

Some professors were not prepared for the amount of additional time that was needed to use the technology to prepare notes and other materials before the course began and for the additional interactions that took place between classes. However professors were able to demonstrate some phenomena in class more effectively than ever before, most notably the use of computer simulation to demonstrate the dynamics of a process in action.

Some students had similar difficulty coping with the additional workload of independent study and interaction that professors demanded.

Although university administrators were quite willing to spend money on physical equipment such as computer hardware, they proved much more reluctant to spend money on the software required to make best use of this equipment.

The university's technology support ("help") system was not adequate for assisting instructors and/or students with the technology, both inside and outside the classroom.

The projection system required the room lights to be dimmed during class; this made it harder for participants to interact. The projection system also made it difficult to display effectively the output of some software, particularly due to small fonts.

3.5 Evaluation and Suggestions for Improvement

Extensive introductory tutorials are required before electronic classes start so that all students have the same basic knowledge and experience in using the computers and related software (such as SW) that will be used in the classroom. Also, there is a need for workshops for faculty, especially for those faculty members who have never taught in electronic classrooms. These workshops should be conducted by faculty members who were successful in teaching in electronic classrooms (possibly measured by student questionnaires). Introductory tutorials should continue for the first two or three weeks of classes to help those who get behind the others.

There is a clear need for equipment which is reliable and whose use does not become an issue in teaching. For example, an unreliable network that tends to crash will distract students' attention to the point where it may be better to switch back from an electronic classroom to a traditional one. It may be worthwhile to install a local server in each classroom, so that problems in a university-wide LAN need not disrupt the classroom presentation (provided, of course, that no access to Internet is required).

Electronic classrooms which provide a network of computers, with a separate computer for every student are not sufficient. What is really essential is to create the possibility of interactions with students. It is extremely useful to ask students to

perform a certain task, and then to take over control of one student's computer, project its screen onto the large screen and annotate the student's solution. However, our initial claim that these interactions should, among other things, support copying on-line notes so that students do not have to retype instructor's notes, turned out not to be true. First of all, note taking is an important skill which students need to exercise. Secondly, students who type their own notes rather than merely copy them, have a chance to customize these notes; that is type more or fewer details depending on their needs. Finally, some note taking gives students a chance to actively participate in the class and avoid boredom.

Electronic classes with say 50 students do not give enough time for many interactions with students. Therefore, it is very important that these classes are interleaved with tutorials for smaller groups of students. During tutorials students must have a chance to do exercises related to recent classes to make sure that they understand the material and master various topics. We realize that this requirement may be difficult to implement because now not only an instructor is needed but also a tutor, nevertheless we strongly believe that this component is essential.

Well structured help is an absolute necessity. Both, instructors and students must know where they can go if help is needed. Sometimes there is an urgent need for a solution and this may require a phone or access to a person who can help. In other cases, help through electronic mail is quite satisfactory. It is unlikely that one person would be able to answer all questions, thus we recommend various levels of help, and additional a „mediator" who can direct a person who is not sure what help is required.

A very good projection system which doesn't force all lights to be turned off is a necessity. We confirmed our initial claim that the instructor must be a focal point of the classroom, and that the human interactions between the instructors and the students can not be replaced by electronic interactions between computers. A good teacher will likely be even a better teacher in an electronic classroom, and a poor teacher will not become an excellent teacher just because of the use of technology.

4 Conclusion

We have reported „teaching methods", „observations" and „evaluation" of a course that has not really been offered. How seriously should these be taken? In fact our observations are firmly based on the actual experiences of ourselves and our colleagues, and these have informed our opinions on how to improve the situation. For example, we are currently investigating how to change our current first year programming course to resemble the model described in this paper; we would like to have fairly large electronic classes (up to 100 students) taught for 2 hours a week with additional tutorials for small groups of 10-15 students. Currently we have experience in teaching this course using electronic classes of up to 60 students for 3 hours a week with no additional tutorials or lab classes.

There is no doubt that our hypothetical course could be implemented. All of the necessary hardware facilities exist. The only missing component is software like

the SW described in [Section 3.3.2]. While some interactions can be implemented through hardware links and switches (for example, see [Shneiderman 95]), we believe that a software solution is cheaper and more flexible. We tried to find such software and failed. Microsoft's *NetMeeting* came close. Unfortunately this system seems to be quite unsuitable for teaching purposes because any student can take control of the classroom interaction simply by moving the mouse. *Lotus Notes* has several useful features, such as the pre-loading of Internet pages, but it does not support all of the interactions that we consider essential.

We believe that students' motivation, ability and knowledge on entry to the course play a vital role in how they perform. Clearly, students who don't know much about basic computer operations and aren't able to learn enough about them in the initial tutorials will suffer many disadvantages. For example, they will be easily lost when using internet browsers, SW, or other software tools. Regarding students' motivation, we are of the opinion that it significantly influenced by the teacher: students who initially are not motivated may become enthusiastic whereas students who initially are motivated may become discouraged. Finally, the students' ability must be taken into account. Some students initially choose to study a discipline that really doesn't quite suit them. For example, as many as 10% to 20% of students who enter Computer Science may switch to a different discipline during their first year at university: some find it too difficult, others simply find it boring. The percentage of students who fail (or drop) a first year course in an electronic classroom does not, by itself, imply a failure of electronic classrooms.

It is our strong belief that electronic classes are the future for teaching. However, we can see no *intermediate step* that leads towards this style of teaching. Instead the teacher must fully commit all available resources and time to try to provide the ideal environment. Even small glitches, which can be disregarded in a traditional environment, may become extremely frustrating in an electronic class. In the case where there is no adequate software to support human cooperation, it may be better to reduce dependence on computer technology and limit computer use to tutorial or laboratory classes, rather than integrating computers into „electronic" classes. Computers are intended to support and extend this cooperation; they will be an obstacle rather than an aid if they cannot be used efficiently. Finally, the role of the human teacher has not disappeared. Teachers continue to be the most important part of any class, determining much more than the technology used how much their students actually learn.

References

[Acadia 96] Acadia University: „Acadia Advantage", http://www.acadiau.ca/

[Grudin 94] Grudin, J.: „Groupware and Social Dynamics: Eight Challenges for Software Developers"; Communications of the ACM, 37, 1 (1994), 93-105.

[Holmes 96] Holmes, M., Porter, D.: „Student Notebook Computers in Studio Courses"; ED-MEDIA'96 Conference, AACE Proceedings, Boston (June 1996).

[Mayrhofer 96] Mayrhofer, H., Scherbakov, N., and Andrews K.: „HM-Card: A New Approach to Courseware Production"; ED-MEDIA'96 Conference, AACE Proceedings, Boston (June 1996).

[Mühlhäuser 96] Mühlhäuser, M., Borchers, J., Falkowski, C., Manske, K.: „The Conference/Classroom of the Future: an Interdisciplinary Approach"; ED-MEDIA'96 Conference, AACE Proceedings, Boston (June 1996).

[Norman 96] Norman, D.A., Spohrer, J.C.: „Learner-Centered Education"; Communications of the ACM, 39, 4 (1996), 24-27.

[Rubin 89] Rubin, Neal L.: „A System for Example-Based Programming"; Proc. CHI, ACM Press (1989), 63-67.

[Shneiderman 95] Shneiderman, B., Alavi, M., Norman, K, and Borkowski, E.: „Windows of Opportunity in Electronic Classrooms"; Communications of the ACM, 38, 11 (1995).

[Yelon 96] Yelon: „Powerful Principles of Instruction"; Longman (1996).

Appendix

10 Principles of Instruction [Yelon 96]

1). *Meaningfulness*. Motivate students by helping them connect the topic to be learned to their past, present, and future.
2). *Prerequisites*. Assess students' level of knowledge and skill and adjust instruction carefully, so students are ready to learn the material at the next level.
3). *Open Communication*. Be sure students find out what they need to know so they can focus on what to learn.
4). *Organized Essential Ideas*. Help students focus on and structure the most important ideas, to be able learn and recall those ideas.
5). *Learning Aids*. Help students use devices to learn quickly and easily.
6). *Novelty*. Vary the instructional stimuli to keep students attention.
7). *Modeling*. Show students how to recall, think, act, and solve problems so that they are ready to practice.
8). *Active Appropriate Practice*. Provide practice to recalling, thinking, performing, and solving problems so that students apply and perfect their learning.
9). *Pleasant Conditions and Consequences*. Make learning pleasing, so that students associate comfort with what is learned; and make learning satisfying, so that students keep learning and using what is learned.
10). *Consistency*. Make objectives, tests, practice, content, and explanation consistent, so that students will learn what they need and will use what they have learned outside of the instructional setting.