Experience with the Learning Web

Mildred L. G. Shaw and Brian R. Gaines
University of Calgary
Calgary, Alberta, Canada T2N 1N4
mildred@cpsc.ucalgary.ca, gaines@cpsc.ucalgary.ca,

Abstract: The *learning web* approach can be introduced into existing undergraduate courses and doing so provides the basis for its implementation on a larger scale transcending institutional boundaries. However, an essential prerequisite is the restructuring of existing pedagogical approaches, de-emphasizing received wisdom and authority of the instructor, and emphasizing collaborative learning and meta-reflection on all aspects of the learning process itself. The instructor becomes a facilitator of students learning to learn, and eventually that process of learning to learn must itself become an overt topic for discussion by students fully participating in its management. In the same way that in the current educational system the home prepares for the school and the school prepares for the university, the university must come to see itself as preparing students for a role in a learning society in which they have come to understand and manage their own processes of lifelong learning. This article describes experience in using the learning web approach and technologies in some senior undergraduate courses in Computer Science.

1 Introduction

The learning web approach (Norrie and Gaines, 1995) offers the opportunity to restructure higher education to function through distributed communities that transcend conventional institutional, regional and national boundaries. However, the methodologies and technologies involved may be applied on a smaller scale to existing courses in order to facilitate collaborative and self-directed learning and break out of the traditional lecture and classroom format. This evolution based on existing courses is important because it provides the resources and experience to develop the learning web on a larger scale. Transforming an existing course to operate as a learning web not only facilitates coping with larger and more heterogeneous classes locally, but also makes the learning materials widely available for use elsewhere. Initially, institutions may share learning web course material as they ‘share’ textbooks, each using it independently within their local courses. However, the ethos of the learning web approach is such that it will be very natural to merge courses across institutions until eventually the notion of a course at an institution itself becomes meaningless.

Our initial aspirations for learning web applications are modest in comparison with the wider vision of what is possible in the long term. However, the seeds of large scale implementation are present in the early applications. Already materials that have been developed for local use have been accessed across the web by teachers and students in more than 25 countries, and learning web tools are being used by others world-wide to facilitate learning in a range of applications wider than we have imagined. The Internet provides an environment where the seeds propagate rapidly, and ‘collaboration’ occurs between groups who know very little of one another.

2 The Learning Web Approach to Computer Science Teaching

This article describes how the learning web approach and tools are being used in two senior undergraduate courses in Computer Science. The students have access to excellent computing facilities and are familiar with the use of computer-based tools, so these experiments do not address issues of access or literacy. The students should have no problems with the technology and, if they do so, it is the technology which is problematic.

However, computer science teaching generally follows the traditional view of ‘privileged knowledge’ in which it is the business of the professor to impart the knowledge embedded in texts. Students are fed ‘knowledge’ in measured portions, expected to digest it, and give evidence in the form of assignments and examinations that they have done so. The ‘received wisdom’ approach emphasizes individual learning and provides little experience in collaborative activities. This is singularly inappropriate to the post-modern employment environment of computer scientists. They need to become reflective practitioners in Schön’s (1983) terminology, because they will need to continually adapt throughout their careers to social change in which their discipline plays a major role. Thus, our experiments have been directed towards using the learning web approach to facilitate collaborative learning in a community of students for whom it is not natural through past experience, and yet where employers expect graduates to function effectively within teams and organizations.

The received wisdom approach is inappropriate to the post-modern world which is characterized by rapid change rather than the application of well-established knowledge. A more operational view of knowledge takes a constructivist approach to learning which is exemplified by the work of Piaget (1972), Papert (1980) and others. This theory of knowledge implies that students learn through active involvement in the social processes of the construction of meaning. Understanding is based on active participation in the subject matter and reflection on conversations with others on the topics of a course. This is the Schön's "reflection-in-action" in which research and learning is a joint enterprise, and leads to a less authoritarian model of professionalism.

As Sculley (1991) has emphasized, the key strength of 21st century organizations will be their ability to unleash and coordinate the creative contributions of many individuals; over-specialization and a limited perspective can be a dead-end trap; individuals will need to have the flexibility to move around; diverse educational experience will be the critical foundation for success; what we will need is not just mastery of subject matter but mastery of learning; we must have access to the unbounded world of knowledge; we must create a learning environment in which research and instruction are integrated. He specifies the requirements for lifelong learning:

- It should require rigorous mastery of subject matter.
- It should hone the conceptual skills that extract meaning from data.
- It should promote a healthy skepticism that tests reality against multiple points of view.
- It should nourish individual creativity and encourage exploration.
- It should support collaboration.
- It should reward clear communication.
- It should provoke a journey of discovery.
- It should be energized by the opportunity to contribute to the total of what we know and what we can do.

These have been the goals of restructuring some Computer Science courses using the learning web approach.

3 Software Engineering

Software engineering is an excellent topic within which to introduce social perspectives because it has long been widely accepted that the problems of harnessing the talents of individual programmers into collaborative teams is a major one for the computer industry (Brooks, 1975; Mayrhauser, 1990). In recent years the gap between customer and user requirements and computing system specification has also become of major concern—*requirements engineering* is now a major sub-discipline in its own right (RE, 1993; Shaw and Gaines, 1994). Thus, it is natural to introduce psychological and social issues in a software engineering course, and to design project work that gives students personal experience of social phenomena in overt and discussible form.

The curriculum developed for CPSC 451, a required software engineering course for all Computer Science majors at this University, is centered on projects that involve the students playing roles in teams representing customer and supplier organizations. There is a lecture component of 3 hours a week for 13 weeks, and a laboratory (practical) component of 3 hours a week for 13 weeks. The lectures are of standard format covering classical software engineering topics and methodologies, including the SEI levels of maturity and continuous improvement in an organization (Humphrey, 1989).

Each student is assigned to two different groups of 10 to 12 students. In one group she is one of the supplier team, and in the other one of the customer team. The students are assigned based on a number of factors such as: having taken the human-computer interface course, having taken a theory course, and length of time in the program, to try to make each group as varied as possible but at the same time as similar as possible to the other groups. The total class size in the past has been around 50 to 60 students, but in recent years has increased to 120 to 150 due to financial constraints.

The process starts in the last 5 minutes of the very first class, when each customer group is allocated a project at random. The course web site gives a very short, informal and vague description of the problem. For example:

> *Write a specification for a system linking supermarkets to a grocery supplier to process the ordering from one end and the invoicing from the other. Both sides should have strategies e.g. reorder when the stock drops to a certain level; do not invoice until the stock has been received. This requires the minimum data entry, but complete security.*

Each group gets a different problem, but they are all of a similar level of complexity. Each customer group has two days to prepare an informal requirements document for the project and post it on the web. They are subsequently responsible for its evaluation and criticism as it progresses; that is, they are the customer for the system. They are expected to be present at all presentations to ask questions, and comment on all the write-ups and documentation. Each grade, given by the instructor not by the students, depends on how thoroughly the evaluation is carried out, the extent to which it is fair and reasonable and the extent with which it agrees with a well-founded methodology. Groups are advised to show all drafts to the teaching assistant (TA), and discuss any problems or disagreements. It is not very long after the start of the project that the customer and supplier groups reach the conclusion that the interchange cannot be done entirely by web-based documents, and that they need to meet and negotiate problems, expectations, and what will be included in each version of the software.

The supplier group works for the customer group, receiving informal requirements for a system, annotating the rqeuirements on the web with queries and suggestions, producing a formal specification, a management plan, the analysis and design in the form of an overall and a detailed design document including test plans, a user manual, coding a prototype, evaluating and refining it, and presenting a final (prototype) product according to the details given. Public (to the whole class) oral presentations and discussions are required at various points within the project, and are evaluated and assessed by the customers. It is certainly not required, but often the students will arrive in their best business clothes for the presentations, and fully enter into the roles they have been given. Every 3 to 5 days another part of the project becomes due for submission to the customers, and students are quickly made aware of the social pressures to conform to due dates. This may be the first time that any of them have considered that due dates are not altogether arbitrary, and that other people's deadlines depend on them. In turn, each student may be inconvenienced by other people's last minute rush to complete work, not only in the other group, but also in their own where, for example, an editor may require input from several people before a final document can be prepared.

No marks are given for coding. This is where much of the effort goes, and this practice seems to the students to be unfair. However, in this course all the students have a thorough grounding in programming, and what they are learning is the application of what they already know or is being covered in specific lectures, the management of their time, how their own working style fits into the range of working styles among their peers, and how their own time and work management affects others. Each student keeps a log of activities in the form of a diary with dates, what was done, and time spent on each item. At the end of each month, every student prepares a set of reports assessing each member of their supplier group. This requires a paragraph per person, including themselves, outlining who did what during the month, and how their work can be assessed, using some sort of grading scheme on one or more criteria. This means that they must get to know the people in their group as soon as possible, and decide how each has contributed to the group work during the month. This is not optional, but is a required part of the assessment to pass the course. In general the reports are thoughtful and responsible. The students do not always give themselves top marks, but say things like: "x did not contribute much to the group discussions — maybe I spoke too much of the time and didn't give x a chance"; or "I thought y was really stupid at first because his section of the documentation was such a mess — but I soon found out that he was a very good programmer and just had difficulty expressing himself in English".

In order to prepare students for the project, information is given about personality variables, the range of possible working styles, and how a person's strong points should be built on rather than concentrating on their weaknesses. Each student is profiled using a short form of the Myers-Briggs Type Indicator to find out their own preferences, and discuss how each type can benefit from input from other types. The students invariably get the message, without it being made explicit, that their job is to encourage the smooth functioning of the group and to involve every member as equally as possible without requiring everyone to take part in every job.

It is not necessary to make explicit the psychological and sociological perspectives of the academic curriculum in CPSC 451. In any event, the science curriculum to which most of the students have been exposed encourages linear thinking and objectivist values, and is a poor foundation from which to understand the life world. The students *experience* the significance of roles, conceptual systems and inter-personal interactions. The alternation of their own customer and supplier roles brings them to terms with the nature of conceptual systems, both their subjective artificiality and their ethical implications in terms of role consistencies, responsibilities and accountabilities. Being responsible for conceiving and articulating *requirements*, in particular, is a new experience for most students, and leads them to be more thoughtful about how those requirements arise.

It has been natural to introduce the learning web tools into the operation of CPSC451 as part of the customer-supplier interaction in the group projects which form a major part of the course. All the course materials are already made available to the students through the web, and it is natural for them to place the documents relating to customer requirements, supplier comments and proposed solutions, the ensuing negotiations, and descriptions and screen dumps of the developed applications on the web. The use of a list server for the discussion and negotiations means that the process of constructing a specification is automatically documented and the archives are available on the web. Repertory grid tools are already in use in requirements engineering for the elicitation of the critical dimensions of an application from customer, user and implementer perspectives, and concept maps are in use as diagramming tools for computer-aided software engineering.

Thus, none of the current learning web technologies is seen as being a radical departure from normal practice once the students have adopted their roles within the customer-supplier teams. In CPSC451 the tools are used to encourage reflection directed to the processes of software engineering, and not to the processes of learning about software engineering. This is done in a senior course for which CPSC451 is a prerequisite which is usually taken one year later as described in the following section.

4 Advanced Information Systems

Advanced information systems provide a topic where social perspectives are readily seen to be essential to redress technological bias. In developing the curriculum for CPSC 547, an optional course on advanced information systems for Computer Science majors, it was known that the final year students already had theoretical foundations for technologies such as object-oriented programming and databases. In addition, many of the students who were attracted to this course also had substantial industrial experience. For example, they understood object-oriented technology in terms of type theory, modularity, and so on, and they understood that it was having a major impact on industry, but they had few sources available on how to bridge the gap between theory and practice: for example, to be able to see object-oriented databases as providing a more effective enterprise modeling technology than relational databases; from there, to go on to the questions of the interplay between organizational needs and technological capabilities; from there, to go on to the question of the influence of the technology on organizational design; and so on.

However, the agenda for CPSC 547 goes far beyond these simple techno-social considerations. The students in this course are preparing for a new industrial infrastructure which is itself radically different from that of a few years ago. It is ‘post-modern’ in the sense of Ekins and Max-Neef’s (1992) *real life economics* recognizing the plurality of economic sectors including environmental and domestic capital, of Warnecke’s (1993) *fractal company* designed to encourage the growth of complete and robust sub-organizations self-similar in their functionality to the whole, of Wheatley’s (1993) *emergent organizations* recognizing the adaptability of the creative chaos of the life world. In Calgary, the recession has seen the end of the large-scale information systems divisions of the oil majors that has dominated computer science employment opportunities in Alberta. Hundreds of information systems professionals have already had to come to terms with a new industrial environment that emphasizes small, adaptable, entrepreneurial organizations. Our graduating students need skills that go beyond mere technical proficiency to cope with the new challenges and opportunities.

The design of an environment for reflective learning has been influenced by the recommendations and beliefs of Carl Rogers (1961) for generating a positive atmosphere in which students exhibit mature everyday behavior, are less defensive, more adaptive, and more able to meet situations creatively. This involves treating each student as an individual, being available to discuss problems individually and help with students’ decision-making, creating a supportive and empathic class atmosphere in which each student is given positive encouragement to discuss issues of concern, and making the instructor’s thoughts and views genuinely available for discussion. According to Rogers this allows each student to experience and understand aspects of her/himself which may not have been previously available, to become more integrated and more able to function effectively, to be more self-directing and self-confident, to become more self-expressive, to be more understanding and accepting of others, to be able to cope with new problems more adequately and more comfortably.

However, it is not simple to switch to this type of classroom interaction for those with years of experience with a traditional approach to learning. For one thing, it threatens the view of the “authority” of the professional who is the ultimate source of all knowledge, and hence requires a high degree of competence and understanding of the subject matter and its ramifications. It also involves a personal commitment to knowing every student in a class of 50 by name by the end of the second week of the course, and being willing to support requirements for resources and equipment that cannot be planned. The students do a great deal of reading the literature, thinking and discussing issues. Emphasis is on cooperation, mutual acceptance and support for differing points of view. After a few lectures, the students in CPSC 547 take over the course and run it through their own group research, presentations and demonstrations addressing major issues in advanced information systems. The students work extremely hard, are incredibly motivated and enthusiastic, achieve a very high standard of work, and think deeply not only about the technology but also about the social and ethical implications of its applications.

In CPSC 451 they learn experientially from playing the relatively well-defined roles of customers and suppliers. In CPSC 547 they learn both experientially and intellectually from playing the open-ended roles of being researchers and educators in their own right. Each presentation tends to set a new standard of excellence which those in the later groups are determined to transcend, and find they must cooperate strongly to do so. Whereas CPSC 451 is a compulsory course, CPSC 547 is optional, and the fact that it has been one of the most heavily subscribed of our 500-level courses attests to its perceived value by students. It provides a bridge from their roles as students to their roles as industrialists, managers, researchers, members of, and contributors to, our rapidly changing post-modern age and information society.

The learning web tools are used in CPSC547 to support the conceptualization, requirements analysis, and presentation for the small group projects which are, however, broader than those of CPSC451 in that they involve the analysis of major areas of computer technology and its impact rather than the development of specific systems. Each group develops its own web site with a view to its being available to others both within the course and, internationally, across the web. They search for, and link to, relevant materials on the web, much of which now comes from other students in related courses or undertaking graduate work at other institutions. Thus, the notion of collaboration is already broadened to transcend institutional boundaries. In the preparation for the Winter 1996 course, for the first time, it has been possible to drop the circulation of reprints of published papers as course materials because adequate background materials for all course topics are now available on the web. This generation of students is already taking electronic, open publication as the norm and only using paper-based materials as a last resort.

However, the most important transition in CPSC547 is that the learning web tools are used also to encourage meta-reflection about the course and the nature of the learning experiences involved. The agenda for the course outlined above is made explicit to the students at the beginning of the course—they become *owners* of the course at every level from day one. The web of previous student projects and commentaries is available to them so that there is collaboration not only within the course but across instances of it, a sense of a continuing learning community where their contributions will be valued by the next generation of students.

The repertory grid tools are used to construe the course topics, and the instructors' grids are made available using the facility of WebGrid (Shaw and Gaines, 1995) to compare grids and present a graphical analysis of similarities and differences in construction. Figure 1 shows a student's grid being compared with that of an instructor. However, the grids of other students are also available so that the instructors' views have no especial priority, and the primary focus is on the discussions generated among students when they find that they have widely different constructs for what appear to be well-defined topics or are using well-established terminologies in very different ways.

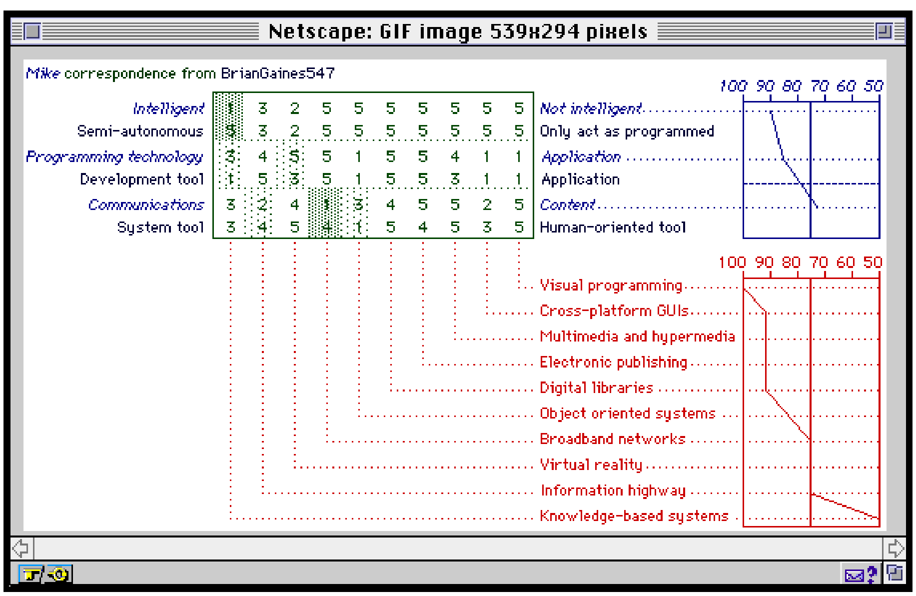

Figure 1 Comparison of student grid on course topics with that of an instructor

5 Conclusions

This article has decsribed our experience in introducing the learning web approach into existing undergraduate courses, including the restructuring of existing pedagogical approaches, de-emphasizing received wisdom and authority of the instructor, and emphasizing collaborative learning and meta-reflection on the learning process. Once the fundamental restructuring has taken place, technology can be used to provide effective support for the learning web objectives on the scale needed which goes beyond the capabilities of individual instructors. We are already seeing evidence of the positive feedback processes that are the basis of major innovations in the way that our students are using resources provided by other students world-wide and are increasingly seeing themselves as contributing to those resources rather than merely ‘taking a course.’ It is only a matter of time before ‘guest students’ from other universities are attracted to become contributors to what were initially list servers for a local course. It is only a matter of time before students graduating and taking up employment return electronically to use the list server associated with a course, either because they have come across an interesting idea or technology to contribute, or because they wish to raise a question that might be answered within the class. We have found that just making the learning web tools available on the Internet and linking them to brief descriptions on our home pages has already led to substantial world-wide use. The learning web, like the Internet and the World-Wide Web where it is now implemented, is a self-propagating culture, each use of which and each extension of which generates new uses and extensions. It has already gone beyond our capabilities to predict or control, and we are only contributors to an inexorable process of change in the structure of learning processes and institutions in our society.

Acknowledgments

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada.

URL’s for Materials

CPSC 451: http://www.cpsc.ucalgary.ca/~mildred/457/
CPSC 547: http://www.cpsc.ucalgary.ca/~mildred/547/
WebGrid: http://tiger.cpsc.ucalgary.ca/WebGrid/
Related papers: http://ksi.cpsc.ucalgary.ca/articles/

References

Brooks, F.P. (1975). **The Mythical Man-Month: Essays on Software Engineering**. Reading, Massachusetts, Addison-Wesley.

Ekins, P. and Max-Neef, M., Ed. (1992). **Real-Life Economics: Understanding Wealth Creation**. London, Routledge.

Humphrey, W.S. (1989). **Managing the Software Process**. Reading, Massachusetts, Addison-Wesley.

Mayrhauser, A. von (1990). **Software Engineering Methods and Management**. Boston, Academic Press.

Norrie, D.H. and Gaines, B.R. (1995). The learning web: An agent-oriented network supporting knowledge access,collaboration and simulation. Maurer, H., Ed. **Proceedings of ED-MEDIA'95 : World Conference on Educational Multimedia and Hypermedia**. pp.500-505. Charlottesville, VA, Association for the Advancement of Computing in Education.

Papert, S. (1980). **Mindstorms: Children, Computers and Powerful Ideas**. New York, Basic Books.

Piaget, J. (1972). **The Principles of Genetic Epistemology**. London, Routledge.

RE (1993). **Proceedings of First International Symposium on Requirements Engineering**. New York, IEEE Computer Society Press.

Rogers, C. (1961). **On Becoming a Person**. London, Constable.

Schön, D.A. (1983). **The Reflective Practitioner**. New York, Basic Books.

Sculley, J. (1991). The relationship between business and higher education: a perspective on the 21st century. Dunlop, C. and Kling, R., Ed. **Computerization and Controversy: Value Conflicts and Social Choices**. Boston, Academic Press.

Shaw, M.L.G. and Gaines, B.R. (1994). Requirements acquisition. **Proceedings of Workshop on Requirements Elicitation for Software-Based Systems**. Keele, University of Keele.

Shaw, M.L.G. and Gaines, B.R. (1995). Comparing constructions through the web. Schnase, J.L. and Cunnius, E.L., Ed. **Proceedings of CSCL95: Computer Support for Collaborative Learning**. pp.300-307. Mahwah, New Jersey, Lawrence Erlbaum.

Warnecke, H.J. (1993). **The Fractal Company: A Revolution in Corporate Culture**. Berlin, Springer.

Wheatley, M.J. (1993). **Leadership and the New Science**. San Francisco, Berrett-Koehler.