Using the Web to Support Cooperative Learning in Software Engineering

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

Abstract: This article reports on the successful use of the World Wide Web to support cooperative learning by undergraduates working in teams on a software engineering course that is concerned with process management and improvement.

1 Collaboration and Negotiation in System Development

The curriculum developed for CPSC451, a required software engineering course for Computer Science majors, is centered on projects that involve the students playing roles in teams representing customer and supplier organizations. 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 for a different project. 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 Project 1 in 1996 was:

> *Write a specification for a system to keep track of a business forms salesman. It must work out the routing (you may assume the route is linearly ordered). A list of which customers are likely to reorder must be produced, where they are located, what type of products they may want, and what reordering rates they have.*

More details about the operation of the course are given by Shaw and Gaines (1996).

It has been natural to introduce web tools into the operation of CPSC451 to support the group collaboration and the customer-supplier negotiations which in previous years were communicated only through documents. All the course materials are already made available to the students through the web, and it is a simple extension for students 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.

2 Requirements Tracking Through the Web

This section reproduces the notes given to students on how to use World Wide Web facilities in a simple and natural way to manage their projects.

Using the Web for Requirements Tracking

The World Wide Web supports the sharing of structured multimedia documents with links to one another and to other material, not necessarily in HTML format. This makes it a suitable base technology on which to develop a requirements tracking system for software engineering that tracks the relationship between requirements, analysis, design, implementation, application and maintenance.

There are complex requirements tracking tools being developed that use the full client-server capabilities of the web to provide highly integrated systems. For CPSC451 we will use the web more simply to develop a set of linked documents tracking the customer group requirements specification, the negotiation with the supplier group to clarify them, the design specification, the customer commentary on this, the implementation, acceptance procedures, and so on.

Rather than use special-purpose servers we will take advantage of the web's capability to link documents held in different locations, and generate a sequence of documents held in the directories of those in the customer and supplier groups. A single master index managed by someone in one of these groups will be used to make the distributed set of documents available to everyone in the groups.

Linking and Annotating Material
By putting all the customer and supplier documentation in the web directories of members of the customer and supplier teams it automatically becomes available to everyone. However, it is not linked together and it may not be obvious how the documents relate to one another. Two things can be done to improve the situation.

First, someone in the customer group should agree to become documentation manager for the entire project and maintain an index in their web directory of all the documentation from both customers and suppliers. This index will contain pointers to the distributed set of documents allowing them all to be accessed from one place. Anyone needing to look at the project documents will start with the index and click on a document in it to fetch that document. Anyone generating a new document that should be entered in the index should email its url to the documentation manager specifying where it belong in the index.

Second, while you cannot edit a document in someone else's account, you can fetch it with a web browser, save its source, annotate that source with your comments, and make the annotated document available in your account. Thus, the index will grow to be a sequence or a tree of documents such as a customer specification, a version annotated by the suppliers with queries about the specification, a version of this annotated by the customers with answers to queries, and so on. Similarly, an initial design specification from the suppliers can be annotated by customers.

Style of Annotation in HTML
You will need to establish some style guidelines for documents so that you can distinguish annotations from the original. One good technique is to use a horizontal rule before and after annotation. Tags for bold and italic can be used to mark particular sections of text to which you are making reference. There is no obvious or standard way of doing this. Experiment with HTML and decide on an approach for your group (but don't spend more time on an annotation scheme than you do on the actual software engineering project!).

What Should be Documented on the Web
Any documents that you can generate on paper you can also put up on the web, so that your project web should finally provide a complete account of all the activities of the project. For example, you can link to design material that includes screen-dumps as GIF images. In a real-world project that used the web itself to implement a client-server system you would even be able to link to the working system so that the requirements and design information was available for maintenance and enhancement support. If your implementation is on the departmental unix system then you should be able to provide a link that starts a live demonstration.

3 Case History—Project 1 Business Forms Salesmen Tracking

The task definition for Project 1 was given at the beginning of Section 2. Figure 1 shows the supplier group's home page—both customers and suppliers generally invent company names, structures and logos to provide a realistic infrastructure for their activities. The lower part of this page (not shown) gives the project structure as: secretary and contact; 2 editors; document manager; a coordinator and 6 members as user interface design team; and a coordinator and 3 members as database design team.

Figure 2 is a concept map of the major part of the web structure developed for the project created automatically through our KMap tool (Gaines and Shaw, 1995). The customer and supplier groups have each established home pages with links to their own documents and to relevant ones in the other group.

The customer group developed an informal requirements specification to which the supplier group responded with a functional specification for a "Business Forms Salesmen Project." The customers took this and annotated it (Figure 3). The suppliers used the annotated document to generate an overall design which the customers again annotated, leading to a detailed design and user manual, again annotated, and to a customer product evaluation and a supplier project evaluation of lessons learnt.

Figure 3 shows the customer annotation of part of the proposed user interface in the supplier's functional specification. The upper part is copied from the supplier's document and the lower part is added in a different type face by the customers.

Figure 4 shows the diagram produced of the proposed user interface when the link to "Diagram" at the top left of Figure 3 is followed.

Figure 1 Home page of supplier group

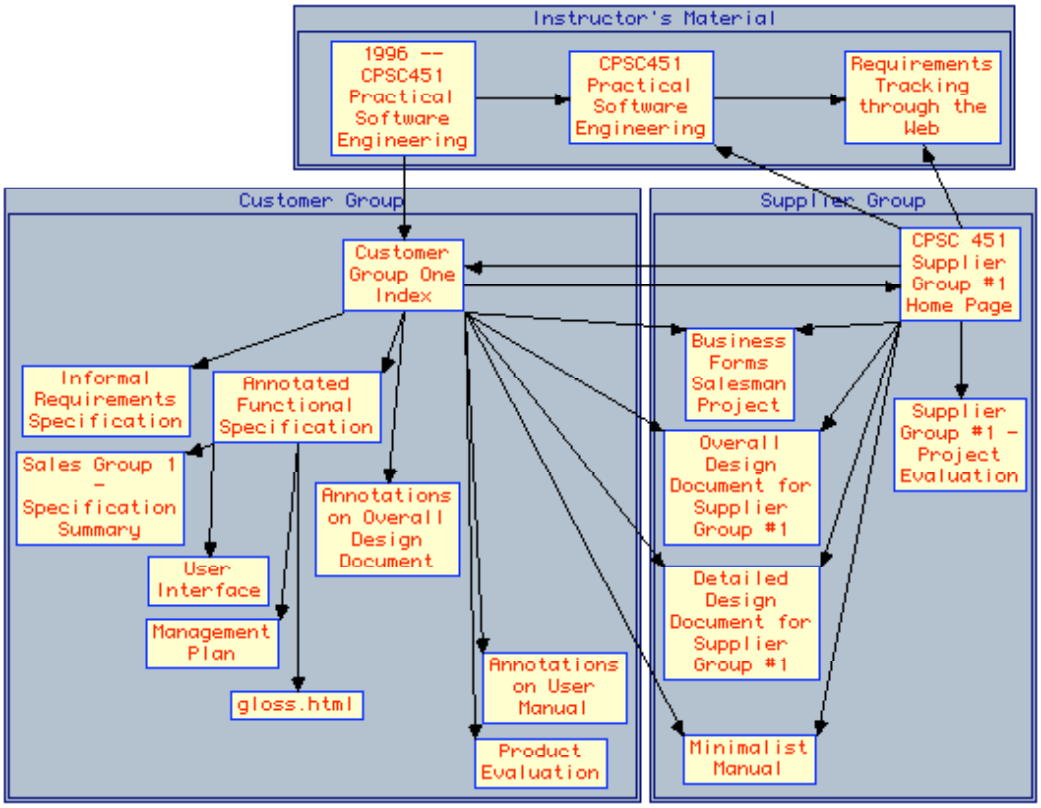

Figure 2 Concept map of customer and supplier webs

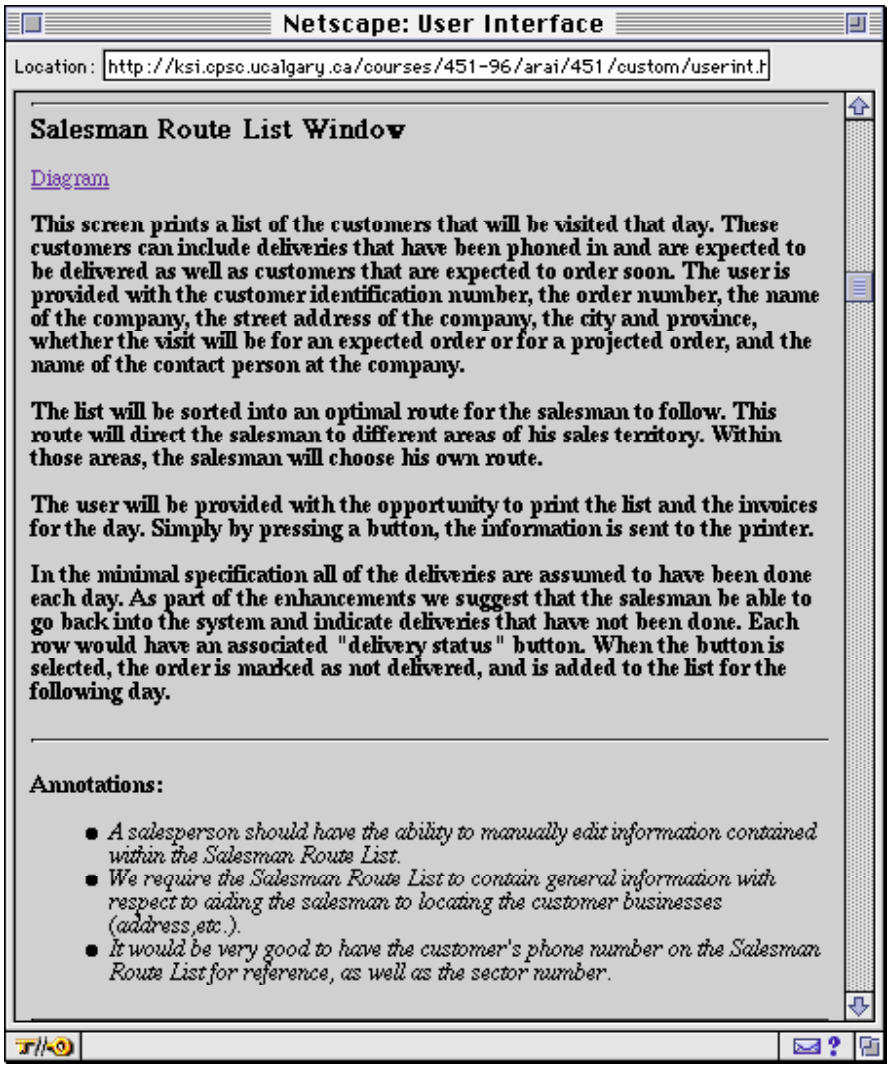

Figure 3 Customer annotation of supplier functional specification

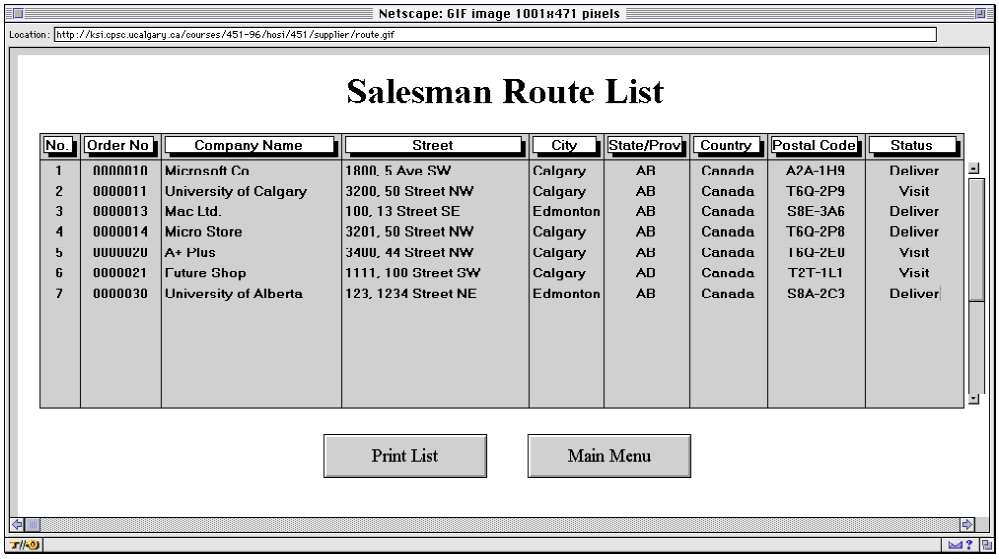

Figure 4 Supplier diagram of screen layout

Thus the two groups were able to communicate their designs and coordinate their negotiations using standard web tools in a simple and natural fashion.

4 Student Evaluation of Lessons Learnt from the Project

Part of the task of the supplier group is to write an evaluation of lessons learnt and place it on the web:

> *The final document required is a project evaluation (about 10-15 pages) which should include a discussion of how well your project satisfies your original functional specifications or what has been added or subtracted; how useful the design process was and what differences (if any) you propose for the design assignment; how the test plan worked, how the real management structure resembled the planned one; comparison of actual times taken for each part; how you would have done it differently if it were a "real" project (dissenting opinions in the group may submit an additional statement); and in what way you could make the project less work if you were to do it again. Include a section on "lessons learned" from doing this project.*

The following subsections are extracts from this evaluation for Project 1.

The Good (What went right):

- Fortunately, we all seemed to get along with one another, and we seemed to avoid any personality conflicts which could have had a detrimental effect on our progress. The members of the group usually communicated well with one another.
- The members of the group had a high level of commitment to the project, so everyone did the work assigned to them. With the exception of losing a group member and not finding out until weeks later, people finished what they said they were going to do. It was good to know that we could depend on people, since one person messing up would have reflected poorly on the entire group. Despite the fact that each of us had a different schedule and other commitments outside of this project, meetings were well attended, and we usually had little trouble finding time to work on the project.
- Everyone stayed involved throughout the project. Since almost everyone was involved to some extent in each phase of the development, we didn't have to spend a lot of time familiarizing people with recent developments.
- The loss of a group member didn't slow us down. The remaining members managed to pick up the slack and make up for the missing person.

The Bad (What went wrong):

- The group lacked clear leadership and structure. We had a wide range of different skills and talents in the group, but unfortunately we had no one who took charge immediately. There was no one in the group who seemed comfortable assigning work to others, especially since all of the group members were essentially equals. It may have been a good idea to choose a leader, although it seems unfair to thrust leadership upon a reluctant victim, and a reluctant leader may have been worse than no leader at all.
- Sometimes people were reluctant to volunteer for tasks. Again, having a leader to assign tasks would have been good. This may have ensured that the workload was spread more evenly amongst the group members.
- Lots of redundant or unnecessary documentation. At times, we felt that meeting the length requirements for certain documents meant that we had to pad our documents with information that wasn't really necessary or appropriate. The time spent adding to the documentation might have been better spent doing other things.

Lessons Learned

- Clearly define a group structure early in the project. This would probably include choosing a group leader, and possibly breaking down the group into smaller subgroups. It is very inefficient for the entire group to participate in each aspect of the project, and meetings of the entire group are often long and tedious. Smaller groups are better able to find time to meet.
- Make sure that as many people as possible are familiar with the chosen programming language. It should be possible to choose a language fairly early, so if only a few people know the language the rest should attempt to gain some experience with it.
- Define standards for formatting documentation. Since it is impossible for one person to do all of the documentation, several people will be contributing to each document. If each person uses their own set of rules then the editor will have to do a lot of extra work to combine the sections into a consistent final form. It creates a lot of wasted time if several people have to write and rewrite the same thing.
- Set a structure for meetings. Regular meetings may be necessary. Free form meetings are a bad idea. There should be an agenda so that everything that needs to be discussed gets taken care of, and everyone knows

what they need to do to prepare for the meeting. After the meeting everyone should be aware of their responsibilities.

- Set deadlines and stick to them. Since many parts of the project pass through several sets of hands on their way to completion, it is essential that each step along the way is reached on time, or else the entire project could be delayed. Make sure to allow flexibility in case of emergency, however.
- This project proved to be a good experience for us. Most of us didn't have much experience working in a group of this size, so there were lots of unanticipated problems that arose. As the project progressed we became more adept at dealing with these problems.

5 Conclusions

This article has reported on the successful use of the World Wide Web to support cooperative learning by undergraduates working in teams on a software engineering course that is concerned with process management and improvement. The supporting technology is very simple and already widely available to educational institutions. CPSC451 used only standard web servers, browsers and text editors. No special-purpose tools were used. The students were third year computer scientists and hence had familiarity with computer use and text editors, but most had no prior experience in the use of the World Wide Web. We have found in other courses with students across a range of non-computing disciplines that the capability to prepare documents for the web is easily acquired in a few hours experience. The approach described in this paper can be used in any discipline.

The web created by CPSC451 students in 1996 is part of the learning environment supplied to CPSC451 students in 1997. Thus collaborative learning is being encouraged not only within cooperative groups, and across negotiating groups, but also longitudinally between classes in different years. The students who are learners in one year know that they are also teachers for the next, and hence can take pride in knowing that their achievements will be of value to others, not just exercises for the instructor to assess and then forget.

CPSC451 is an important course for students seeking employment in industry, since most potential employers see computer science students as lacking in the group collaborative skills that CPSC451 develops. The archiving of the materials on the web automatically creates a "portfolio" for each student through which they may show to a potential employer that they have had experience both in specifying requirements for others and in developing systems through teams that have negotiated requirements with others.

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 archives for 1996: http://ksi.cpsc.ucalgary.ca/courses/451-96/
Related articles: http://ksi.cpsc.ucalgary.ca/articles/

References

Gaines, B.R. and Shaw, M.L.G. (1995). WebMap: concept mapping on the web. **World Wide Web Journal 1**(1) 171-183.

Shaw, M.L.G. and Gaines, B.R. (1996). Experience with the learning web. Carlson, P. & Makedon, F., Ed. **Proceedings of ED-TELECOM 96 : World Conference on Educational Telecommunications**. pp.320-325. Charlottesville, VA, Association for the Advancement of Computing in Education.