GROUP SUPPORT TOOLS AS A SCAFFOLD FOR DEVELOPING GROUP SKILLS: DESIRING TO BE DISPENSABLE

Michele H. Jackson
Department of Communication
jackson@colorado.edu

William Waite
Department of Electrical and Computer Engineering
william.waite@colorado.edu

Amer Diwan
Department of Computer Science
amer.diwan@colorado.edu

University of Colorado, Boulder, USA

1. INTRODUCTION

In some types of professions or organization, decision-making depends heavily on group processes. The examples we are concerned with are the technical disciplines of computer science and engineering. Success in the current engineering environment requires individuals who are able to recognize and fill gaps in their knowledge, work effectively in groups, and use past experience to predict future performance. In all sciences, expectations for collaboration and interdisciplinary work are increasing [Cohen et al. 2001, Ziman 2001].

As a profession, engineering expects the university to inculcate students in what it means to be a professional. Thus, students should not only learn the body of scientific knowledge underpinning their particular discipline, but also the norms and behaviors of a practicing engineer, including how to work with others in knowledge acquisition and artifact creation, and how to plan their activities based on an accessible history of previous projects.

Translating these general educational objectives into specific classroom environments is a serious challenge. First, limitations of available design tools tends to constrain instructors to the use of relatively well-defined, simple problems that bear little resemblance to the problems that are faced by practicing engineers. Second, engineering curricula typically do not include courses specifically intended to develop skills needed for group work. Given that group projects are typically only one of the many components of existing courses, these projects generally are inadequate in teaching students the necessary collaborative skills. The approach we have taken in our project is to involve students in realistic projects beginning with the introductory courses but to use an infrastructure, or *scaffolding*, to selectively hide the complexity from the students. As students progress to more advanced courses, more of the scaffolding falls away, revealing more complexity until, finally, students engage in realistic projects involving significant collaborative work.

As a design objective for group support technologies, scaffolding aims to develop tools that, through their success become dispensable. As such, the principle is distinct from other technologies developed to augment or enhance the performance of an interacting group, perhaps being appropriated by the group so that it becomes integral to its functioning . In other words, group develops some dependence on the technology. In a scaffolding approach, in contrast, the group develops independence with regard to the technology.

A project currently underway at the University of Colorado, Boulder, USA, uses online tools to help engineering students improve their ability to perform effectively in normal (non-GSS supported) group interaction. In this paper, we present a first implementation of group support technology intended to scaffold procedures of decision making.

2. GOALS AND ASSUMPTIONS

The primary goal of our research is to introduce task complexity earlier in the computer science curriculum (as early as the first year of university study). Specifically, we introduce complexity in two ways: (i) through attempting 'trickier' tasks and (ii) through adopting more collaborative design and development procedures.

This report focuses on an initial implementation of a tool to improve group decision making procedures. This implementation deals with the second course of a cumulative three-course sequence not uncommon in computer science curricula: Data Structures, Programming Languages, and Compiler Construction. In terms of building collaboration skills, the first course introduces the importance of interdependence, the second provides experience in standard decision-making procedures, and the third introduces negotiation skills and stakeholder models for ambiguous problem-solving. The tools are Internet-based, used outside of the classroom, and currently developed in HTML and Java using FoxPro for database management.

Several assumptions guide our design. First, given research on the culture of engineering contexts (e.g., [Kunda 1992, Vincenti 1990]), students are ego-motivated to participate in group procedures. The sense of competition would cause students to engage in substantive discussion. Properly directed, such desire could then demonstrate the advantages of group over individual decision-making for complex problems. Second, drawing on Williams and Kessler [2000] we assumed that students are generally averse to group work, or at least that they see technical projects as more suited to solo work. Third, we assumed that students generally do not possess the skills to collaborate effectively in face-to-face groups. This assumption was based on several years' of student self-reports regarding study patterns, as well as structured interviews conducted the semester before our implementation. Students tended to be strategic in their use of groups, either decomposing the group task into a set of individual tasks, or allowing one member to take responsibility for the entire group.

We also assume certain things about the nature of group work. Most basic is the importance of developing an individual's skills to work across groups, instead of the skills of an intact group to work together. Second, specific group skills can be identified and partitioned out for emphasis and development. Third, that these skills can be gained tacitly [Polanyi 1967], through integration with actual tasks, rather than through specific, explicit training. These assumptions represent a particular view of group interaction that will be tested over the course of this research project. They also form the core principles of our scaffolding approach to the design of group support technology. To what extent can group support technologies perform the function typically performed by formal instruction or training, that of improving an individual's abilities to work in a group? This contrasts to the typical 'augmentation' approach found in group support technologies, in which technology allows groups to do things they otherwise couldn't, or to allow them to do it better. Here, the aim is to build fundamental skills that can be transferred to any task-based group context.

3. PROJECT AND SYSTEM DESCRIPTION

The first class we were able to implement was Programming Languages, the second of the three-course sequence. While the department offers several sections of PL, only one (taught by one of the authors) was redesigned. Initial course enrollment was approximately 100, declining to around 70 by the end of the semester. Three projects of increasing technical complexity were assigned throughout the semester. The first project asked students to write a simple compiler for translation of code macro-extended shtml into basic shtml. In the second project, students used the Modula-3 programming language to explore parameter passing mechanisms. In the final project, students used a functional language, SML, to implement a compiler to optimize code.

Each project divided into four segments completed sequentially over 3-4 weeks:

Stage 0: project preparation, learning the programming environment

Stage 1: writing the code

Discussion of possible alternatives and class choice of one of these

Stage 2: implementation of what was decided during the discussion

The projects also increased in complexity in terms of decision-making procedures. Technological support was used only for the discussion segment of the projects. In addition, this segment was also the only one in which students collaborated with one another. The model for the discussion segments was a variation of the standard multi-attribute utility model [Jarboe 1996] in which decision criteria are weighted in importance, alternatives are then rated on those criteria, and final rankings emerge based on a utility algorithm. According to Jarboe, the advantages of such an approach is its consistency, each alternative 'is measured against the same set of standards' (p. 360).

In the first project, the criteria and the alternatives were supplied, and students supplied the ratings for each. Because some rankings of alternatives on criteria could be seen, according to course material, to be evidently 'better' than other rankings, answers were graded according to how well they aligned with the instructor's preferred rankings. Limitations to such evaluation procedures were recognized after the project was completed, and alternate means of evaluation were used in the second and third projects.

In the second project, students first entered alternatives and provided justifications for why they supported those alternatives. Instructors next selected a subset of these options and students then followed the procedure used for the first project. Students were graded according to the clarity of their justification for their support of alternatives.

Due to other factors in the course, the third project was pressed for time and the discussion segment was truncated. In this project, in addition to entering alternatives (as they did in project 2), students also entered criteria. They also entered justifications for their proposals (or arguments against other entries), and had the ability to return to the system to respond to the entries of other students, in effect creating an online discussion. In fact, the assignment given to the students explicitly called for them to "view these comments as an online dialog between yourself and your fellow students." The remainder of the discussion proceeded as in project 2, with instructors selecting a subset of student entries for rating.

4. RESULTS

4.1 Tool Performance

Students experienced no problems in understanding the procedures for using the tools. The tools were accessible and reliable. As might be expected from a first instance, we did view limitations that stem from the implementation approach. For example, the system could not support iterations in ranking, as might be desirable. The elements of the procedure had to be strictly sequenced, sometimes resulting in a choppy or disconnected process. We will address these concerns in future implementations.

4.1 Student Performance

Student performance varied across the three projects. The first project simply asked students to rate alternatives. The resultant decision was of high quality. The second projects asked students to propose and justify alternatives. Student behavior here divided into two camps: (i) those that entered interesting options, including ones the instructor did not anticipate, and (ii) those that entered clearly inferior options that indicated that they had not taken the time to think through their contributions. The first group also provided considerable justification for their entries, beyond what was expected by the instructors. In addition, several students directly responded to and debated other student's proposals.

The third project was problematic, due primarily to issues of implementation. Because time was too compressed, fewer students entered interesting options, the discussion was less rich, and there was little to no debate and refutation. Further, the students seemed not to recognize the increased complexity of the problem. The problem *appeared* simple but upon adequate reflection proved not to be, yet many students never looked beyond the simple front.

4.3 Decision Quality

The quality of the group decision was high in the first and second projects. Indeed, in the opinion of the course instructor, the group decision for the second one was surprisingly wise. In the third project, reaching a high quality group decision required a careful and sustained discussion in order to properly perceive that the seemingly simplest answer was actually the least effective. Such discussion did not occur; consequently, the better students ranked the 'best' answer highly, but most students did not.

5. DISCUSSION

Results confirm the ability to incorporate specific group procedures into the structure of relatively complex technical work. Results also suggest that the procedures implemented in this initial tool can return high quality group decisions. Yet a number of limitations must be addressed in our next iteration. One objective will be to remove the need for the instructors to intervene in the group processes, through the selection of criteria and options for rating. A second change must be to the assignments themselves. Interviews with students suggest problems were caused by the first project having a more-or-less 'right' answer. Some students lost faith in the system as a means for authentic group decision-making, and instead attempted to guess what the instructor 'wanted.' The importance of faith in a GSS for producing authentic group results is well documented in prior research. A third change should be to the timing of the group work within each project so that deliberation and discussion occurs earlier, during the planning stages of each project rather than the selection stage. This introduces collaboration earlier in the process, where the level of ambiguity (and therefore the need for collaborative discussion) is greater.

Finally, future attention must be directed toward the importance of the group context in shaping the appropriation of the system [Poole & DeSanctis 1992]. Our concern was to make the projects nontrivial, where interviews suggested more importance in making them *ambiguous*. In the students' perception, structures prevalent in the organizational context of the computer science department--such as the course grading structure, an emphasis on 'correct' answers, and a strict policy against plagiarism (which made students wary of working in groups)--worked against the acceptance of task ambiguity and against genuine collaboration.

COHEN, L., MCAULEY, J., DUBERLEY, J. (2001), Continuity In Discontinuity: Changing Discourses Of Science In A Market Economy. *Science, Technology, & Human Values,* 26(2), 145-166.

JARBOE, S. (1986), Procedures For Enhancing Group Decision Making, In: POOLE, M.S., HIROKAWA, R., Eds., *Communication And Group Decision Making,* 2nd ed., Sage, Beverly Hills, CA, pp. 345-383.

KUNDA, G. (1992), *Engineering Culture: Control And Commitment In A High-Tech Corporation,* Temple University Press, Philadelphia.

POLANYI, M. (1967), *The Tacit Dimension,* Anchor, New York, NY

POOLE, M.S., DESANCTIS, G. (1992), Microlevel Structuration In Computer-Supported Group Decision Making, *Human Communication Research,* 19(1),5-49.

VINCENTI, W.G. (1990), *What Engineers Know And How They Know It,* Johns Hopkins University Press, Baltimore.

WILLIAMS, L.A., KESSLER, R.R. (2000), All I Really Need To Know About Pair Programming I Learned In Kindergarten, *Communications of the ACM,* 43(5), 108-114.

ZIMAN, J. (2000), Real Science: What It Is And What It Means. Cambridge University Press, Cambridge.