Designing for Internal vs. External Discourse in Groupware for Developing Critical Discussion Skills

Daniel D. Suthers
Learning Research and Development Center
University of Pittsburgh
Pittsburgh, PA 15260
+1-412-624-7036
suthers+@pitt.edu

ABSTRACT

Groupware for learning may differ from that for other software in a crucial way: optimizing software support for subtasks can degrade overall collaborative learning. This point is illustrated in the context of a software environment for supporting student's learning to engage in critical discussion of competing scientific theories. Our experience suggests that in applications where a nontrivial portion of the discourse is external to the software, some exceptions to locally useful interface designs may be encountered, even for supposedly well understood subtasks.

KEYWORDS: Collaborative Argumentation Environment, Educational Application, Design

INTRODUCTION

As future "consumers" of science, today's students need to understand the process by which the claims of science are generated and revised. Towards this end, we experimenting with a software environment, called "Belvedere" [3], for supporting student's learning to engage in critical discussion of competing scientific theories. A number of researchers have experimented with hypertext systems and graphical interfaces for supporting argumentation [1, 2, 6]. For the most part, these systems are designed to provide either a medium for a generic competent reasoner, or support for a specialized expert user in a specific professional practice. The "Belvedere" effort seeks to support the development of scientific argumentation skills in young students. These students can't be presumed to have either general skills of constructing arguments or the specific knowledge of a domain. Therefore, the design of Belvedere has had to address the cognitive and motivational limitations and requirements of these unpracticed beginners, as presented in the psychological literature and as we encountered them in formative testing with students.

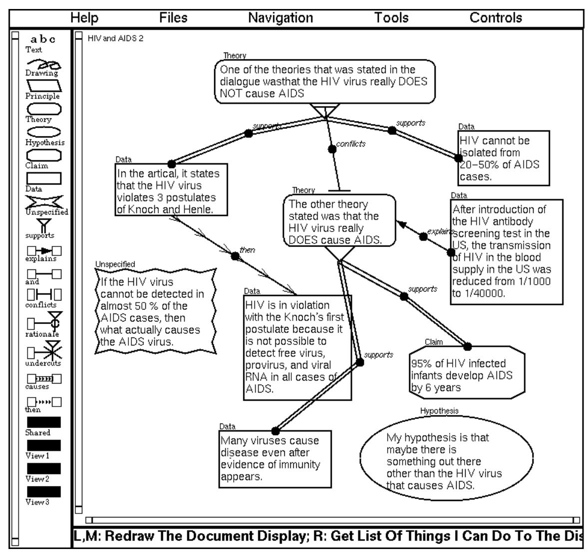

Figure 1: Diagram in Belvedere by a student dyad.

Briefly, these limitations include (1) difficulty recognizing abstract relationships implicit in scientific theories and arguments, (2) difficulty focusing attention on particular problems encountered in the construction and evaluation of complex arguments, (3) lack of domain knowledge, and (4) lack of motivation. Belvedere addresses these limitations by (1 and 2) giving arguments a concrete diagrammatic form, (2) providing an automated, on-demand argumentation advisor, (3) providing access to on-line information resources; and (3 and 4) supporting students working in small groups to construct documents to be shared with others [4, 5].

Superficially, Belvedere is networked groupware for constructing representations [3]. The interface looks like a drawing program, but using it feels more like assembling components into desired configurations. However the utility of Belvedere's representations are in their stimulus value as well as in their modeling value. When Belvedere stimulates productive discourse, in some student groups much of this discourse occurs external to the representations that result. Because of this, our emphasis is primarily on designing representations *the production and inspection of which stimulate critical discussion*, and secondarily on representations that are adequate in themselves as a medium of communication or as the basis for final-product representations of a debate.

EXAMPLES OF DESIGN ISSUES

The emphasis on stimulating critical discussion complicates the criteria for interface design. Although we design to make it easy to construct diagrammatic representations of the dialectical aspects of science, we also design to stimulate external discourse that need not be recorded in the diagram. We have found that the latter goal can overrule the utility features we would otherwise provide in support of the former. This point is illustrated with a few examples.

Statements in Belvedere are embedded in shapes that represent their epistemological status (e.g., as "theory," "claim," or "data;" see figure 1). Users often discuss the epistemological status of a statement before representing it in the diagram. An object can only have one shape at a time; therefore their discussion of the epistemological status cannot be part of the diagram. Is this a design flaw of the graphical language? Should we use an epistemologically noncommittal representation for statements, and provide annotations with which users can record any disagreement concerning the epistemological status of a statement? If the goal is to "push" all discussion into the interface, perhaps these questions are answered in the affirmative. However, it may be useful to force a decision prior to entry in the diagram precisely *because* it stimulates discussion towards making the decision. Otherwise students might never care to discuss the difference between "data," "claims," "hypotheses," and "theories." We have not resolved this issue, but the present point is that it is a nontrivial issue. It illustrates the danger of assuming that optimizing a representation with respect to criteria of epistemological adequacy will constitute an optimization of the representation with respect to the larger task of interest.

Enforcement of semantic constraints provides another example. In some versions of Belvedere, semantic constraints on the links are not enforced. For example, an "explains" link can be drawn from data to theory as well as from theory to data (figure 1). (Instead of enforcement, we provide an "Advisor" that, at the user's request looks for these and other semantic anomalies that can be detected on a purely formal basis, and makes suggestions for improvements.) If Belvedere were a CAD tool for use by expert members of some community of practice, there would be no point in allowing users to make such errors. However, Belvedere cannot be a CAD tool because its users do not yet share standard terminology and practice in argumentation. Furthermore, in a learning environment we must consider the role of "errors" in the learning process. Some errors may be so superficial that they are not likely to result in a useful learning experience. The interface should be designed to make these errors impossible. Users may more quickly develop a shared terminology based on constructs whose semantics are enforced. On the other hand, delayed or absent feedback may be more appropriate for incomplete or problematic *patterns* of argumentation. Immediate "correction" could prevent users from engaging in processes of theory criticism and revision that are encountered in the real world. Investigation of whether immediate enforcement of semantics, feedback on request, or no feedback at all has a better qualitative effect on the user's discussions is ongoing. The point, again, is that this is a nontrivial issue, illustrating that interface features that are considered to be "good" for certain subtasks may be suboptimal for the larger task of interest.

CONCLUSIONS

The specific requirements of our application may be somewhat unusual, but the lesson can be generalized. Design to support discourse processes must transcend the representational environment of the software itself, even in software that specifically relies on the utility of online representations for discourse. User's discourse processes take place in the social environment as well as within the representational and computational resources provided by support software. Thus, the utility of software features should be evaluated in terms of how well they stimulate the right kind of activity in the total human-computer system. We do not assume that local optimization of software support for isolated subtasks (e.g., making "correct" argument diagrams) always optimizes overall task performance. Rather, our main question is: What kind of discourse is facilitated or stimulated by each feature of the interface and of the task posed to the students, and what kind of discourse is inhibited?

ACKNOWLEDGMENTS

This research was conducted while supported by grant MDR-9155715 from the NSF Applications of Advanced Technology program. The author also thanks Violetta Cavalli-Sforza, John Connelly, Alan Lesgold, Massimo Paolucci, and Arlene Weiner for valuable input and support.

REFERENCES

1. J. Conklin and M.L. Begeman. gibis: A hypertext tool for team design deliberation. In *Hypertext '87*, pages 247–252, Chapel Hill, NC, November 1987.

2. G. Fischer, R. McCall, and A. Morch. Janus: Integrating hypertext with a knowledge-based design environment. In *Hypertext '89*, pages 105–117, Pittsburgh, PA, November 1989.

3. M. Paolucci, D. Suthers, and A. Weiner. Belvedere: Stimulating students' critical discussion. In *CHI95 (to appear)*, Denver, CO, May 1995.

4. M. Scardamalia and C. Bereiter. Higher levels of agency for children in knowledge building: A challenge for the design of new knowledge media. *The Journal of the Learning Sciences*, 1(1):37–68, 1991.

5. R. E. Slavin. *Cooperative Learning: Theory, Research, and Practice*. Prentice-Hall, Englewood Cliffs, NJ, 1990.

6. P. Smolensky, B. Fox, R. King, and C. Lewis. Computer-aided reasoned discourse, or, how to argue with a computer. In R.Guindon, editor, *Cognitive Science and its Implications for Human-Computer Interaction*. Lawrence Erlbaum, 1987.