T. Koschmann, D. Suthers, & T.W. Chan (Eds.), *Computer Supported Collaborative Learning 2005: The Next 10 Years!*
Mahwah, NJ: Lawrence Erlbaum Associates.

Cognitive Tutoring of Collaboration: Developmental and Empirical Steps Towards Realization

Bruce M. McLaren*, Lars Bollen+, Erin Walker*, Andreas Harrer+, Jonathan Sewall*

*Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA USA
bmclaren@cs.cmu.edu,
erinwalk@andrew.cmu.edu, sewall@cs.cmu.edu

+Collide Research Group
University Duisburg-Essen
Duisburg, Germany
harrer@collide.info,
bollen@collide.info

Abstract. In this paper, we describe developmental and empirical steps we have taken toward providing Cognitive Tutoring to students within a collaborative software environment. We have taken two important steps toward realizing this goal. First, we have integrated a collaborative software tool, Cool Modes, with software designed to develop Cognitive Tutors (the Cognitive Tutor Authoring Tool). Our initial integration does not provide tutoring *per se* but rather acts as a means to capture data that provides the beginnings of a tutor for collaboration. Second, we have performed an initial study in which dyads of students used our software to collaborate in solving a classification / composition problem. This study uncovered five dimensions of analysis that our approach must use to help us better understand student collaborative behavior and lead to the eventual development of a Cognitive Tutor for collaboration. We discuss our plans to incorporate such analysis into our approach and to run further studies.

Keywords: Collaborative learning, Cognitive Tutors, jigsaw design, spatial effects on problem solving

INTRODUCTION

Intelligent Tutoring Systems (ITS) have long been used to provide one-on-one (machine-to-student) instruction (Wenger 1987). We are interested, however, in using a software tutor to instruct *multiple* students collaborating on a single problem. There have been steps toward providing tutoring in a collaborative environment (e.g., Goodman *et al*. 2003; Suthers 2003; Lesgold *et al*. 1992), but many difficult challenges remain. For instance, the space of possible actions among collaborating users is huge for even the simplest of problems, thus making the analysis of learner behavior much more difficult for collaborative tasks than for the single-student case.

As a step toward addressing the complexities of a collaborative environment, we have created a tutor-development methodology that leverages actual problem-solving data not only to guide ITS design, as has been done in past work (e.g., Koedinger and Terao, 2002), but also to contribute directly to tutor implementation (McLaren *et al*. 2004a; McLaren *et al*. 2004b). Using this approach, called *bootstrapping novice data (BND)*, groups of collaborating students attempt to solve problems with a computer-based tool. While they work, the system records their actions in a graphical representation that combines all of the groups' solutions into a single graph that can be used as the basis for building a tutor and analyzing the collaboration. Our initial BND implementation is realized through the integration of a collaborative modeling tool, Cool Modes (Collaborative Open Learning and MODEling System) (Pinkwart 2003), and a tutor authoring environment, the Cognitive Tutor Authoring Tools (CTAT) (Koedinger *et al*. 2004).

Our ultimate research aim is to develop better support for collaborative learning through cognitive tutoring. In this paper, we describe two steps we have taken toward realizing this ambition: (1) an initial implementation of the BND methodology and (2) a study using the BND approach, including the results and the implications for further development of the methodology. The study we have performed reveals some interesting aspects of the way dyads solved a particular collaborative problem. More importantly, it has pointed us in the direction of improving our implementation of the BND methodology and realizing cognitive tutoring in a collaborative environment.

In our initial implementation of the BND methodology, depicted in Figure 1, Cool Modes (shown on the left) provides the graphical user interface, including a shared workspace that all collaborators in a session can view and update, a palette with objects that users can drag onto the workspace, a chat area, and a private workspace. Cool Modes sends messages about students' actions (e.g., "create an IS-A link") to CTAT's

Behavior Recorder (also referred to as the "BR" and shown on the right of Figure 1), which stores the actions in a *behavior graph.* Edges in the graph represent student actions and paths through the graph represent attempted solutions to the problem. The current approach keeps track of the number of times actions are taken by the various collaborating groups and presents these "traversal counts" on the edges of the behavior graph, e.g., 3 student dyads took the action from the "start state" Classification-Composition to State1. Using CTAT, a tutor author can subsequently transform the generated behavior graph into a *Pseudo Tutor*, or problem-specific tutor (Koedinger *et al.* 2004), by adding or deleting edges, labeling correct or buggy behavior, and adding hints to the edges. To use the finished graph as a tutor, the BR is switched to "model-tracing" mode in which student actions are compared to the graph, instead of recorded, and error messages and hints are delivered to the student. While our ultimate aim is use this approach to provide cognitive tutoring within Cool Modes, as well as other collaborative environments, our initial focus is somewhat more modest: We want to analyze data that was collected using the BND methodology to help us better understand both collaborative behavior and how we can enhance the BND methodology to provide more useful analysis of that behavior. The preliminary study that we have performed is an example of such an analysis.

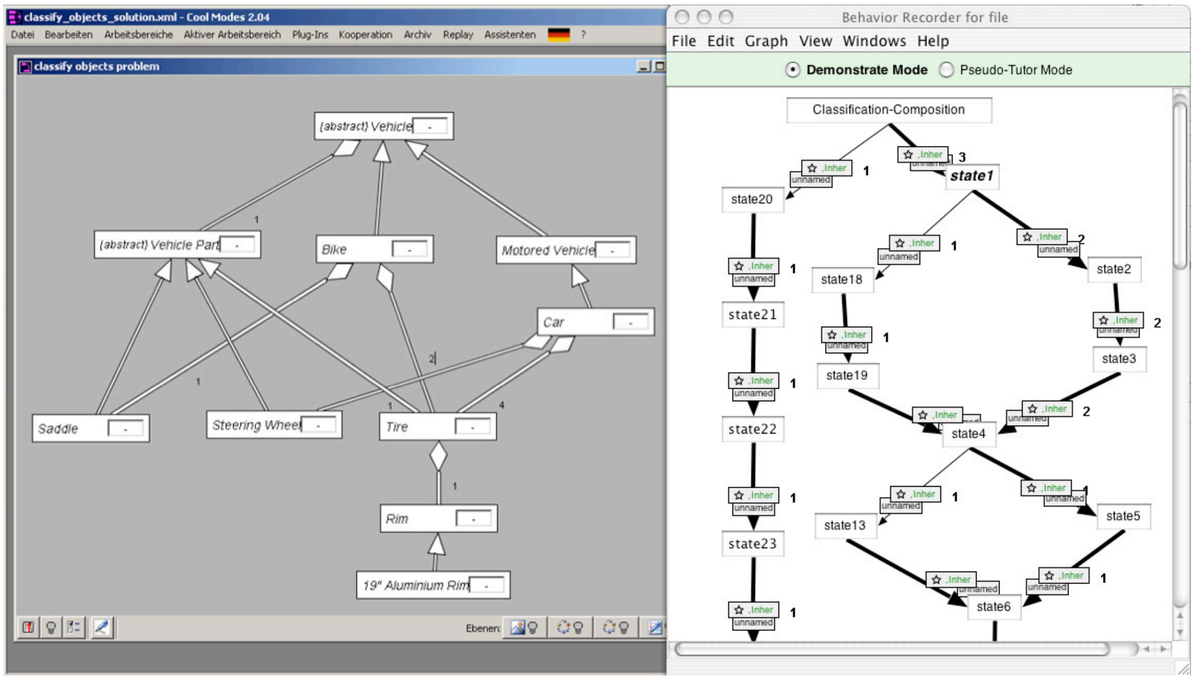

Figure 1: The student's view of the integrated Cool Modes (left) and the Behavior Recorder (right) environment. This shared Cool Modes workspace is from a vehicle classification / composition task that was completed by a dyad of collaborating students. The behavior graph at right shows the amalgamated solutions of different collaborating groups of students.

DESCRIPTION OF THE STUDY

The research question in the preliminary study was whether, in a graphical problem-solving domain, an organized arrangement of objects leads to quicker and better collaborative solutions than a disorganized arrangement. We also wondered whether student rearrangement of the objects facilitates quicker and better results and how this rearrangement might be conducted in a collaborative scenario. To explore these questions and test how the BND methodology might be a useful analysis tool, we assigned 16 students to 8 dyads and asked each dyad to solve an object-modeling problem using the Cool Modes / BR integrated system (one subject was a class assistant). The objects in the given problem were vehicles (e.g., "Car") and parts of vehicles (e.g., "Tire"). The student dyads were asked to relate the objects using classification and composition links. The students were volunteers from a "Modeling Techniques in Computer Science" course at the University of Duisburg, Germany. Seven of the students (pairs 6, 7, 8 and one in pair 5) had had previous experience with Cool Modes. All students received approximately 5 minutes of instruction on how to use the system before the experiment. The student pairs worked at separate workstations, back-to-back in the same room. They shared a single Cool Modes workspace.

To specify IS-A (i.e., classification) and PART-OF (i.e., composition) links between objects in the workspace, the students used the Unified Modeling Language, a graphical modeling technique. To stimulate collaboration we used an unequal resources design akin to jigsaw experiments (Aronson et al. 1978). One student was provided with IS-A links only and one with PART-OF links only, so that no student could solve the problem alone. Students communicated by typing statements into a chat box. The only other communication permitted was the actual composition and repositioning steps taken by the students in the shared workspace.

The 8 groups were randomly assigned to two experimental conditions. In Condition 1, pairs attempted to solve a problem in which related objects were close to one another, providing an organized visual display of the final network. For example, the two abstract classes “Vehicle” and “Vehicle Part” were located near the top of the visual space, and most of the subclasses were located near their respective super classes. In Condition 2, pairs solved a problem for which the objects were positioned in the workspace without any clear organizational principle.

STUDY RESULTS AND ANALYSIS

All 8 dyads completed the task. Students’ solutions can be divided into three categories: ***good*** solutions (groups 5 and 8), ***incomplete*** solutions (groups 2, 6, and 7), and ***poor*** solutions (groups 1, 3, and 4). All three poor solutions were in the disorganized condition, and two of the incomplete solutions were in the organized condition. While one of the good solutions was in the organized condition, one was in the disorganized condition. There were negligible differences in the time required to complete the task between the different solution categories. The informal results suggested that a clearly organized problem state does not necessarily lead to quicker and better solutions than a disorganized problem state, but does lead to different types of errors. It is of course impossible to draw statistical conclusions about the relationship between the solutions and starting conditions with such a small sample size, but we nevertheless have made some interesting informal findings. We now describe those findings.

The conceptual distinctions between good, incomplete, and poor solutions were related to the errors committed in solving the classification / composition problem. In the good solutions, errors reflected misunderstandings about the meaning of classes. Otherwise, students correctly divided the objects into subclasses and super classes and correctly placed the inheritance and composition edges. In the incomplete solutions, students only connected one inheritance or composition link from each class. As a result, they had too few links, and left out key relationships. In the poor solutions, students would often connect a class to multiple ancestor nodes of the same lineage. For example, in Group 3, students connected "Car" to both "MotorVehicle" and "Vehicle." The poor solution groups also typically created too many edges and were logically inconsistent about the connection decisions they made.

Solutions can also be characterized by the way students tended to move nodes in the shared workspace. In the good solutions, the students separated the two abstract classes, placing one at the top of the workspace and the other at the bottom. IS-A and PART-OF links flowed in opposite directions and crossed only when necessary. Objects were organized in rows that reflected their level of abstraction. In the incomplete solutions, the two abstract classes were placed relatively close to one another, all links pointed in one direction, and there were no crossed links. The objects tended to be clustered together and were organized into fewer rows than in the good solutions, but the rationale behind the organization wasn't as clear. Finally, the poor solutions had the abstract classes positioned without an obvious rationale. They had much longer edges pointing in all directions and frequently intersecting with one another. As a result, the poor groups tended to use the entire shared workspace.

We then analyzed the processes associated with the development of each solution. While working on the problem, students could take three types of actions: *chat* actions, "talking" to a partner in a chat window, *move* actions, repositioning an object in the shared workspace, and *creation/deletion* actions, creating or deleting edges. Solution types showed differences in collaborative and task-oriented behavior. In terms of collaboration, the students with the good solutions had different approaches. Group 8 worked completely collaboratively. Members would take turns moving and creating objects and for a given group of objects, one member would reposition objects while the other would create the edges. Conversely, the members of Group 5 worked completely in parallel, coordinating their actions only to correct their partner's mistakes. In the incomplete and poor groups, pair members shared the work. The poor groups were informal in their turn taking, while students in the incomplete groups would alternate taking the initiative. With the exception of Group 5, groups decided on their actions using the chat window and ensured that both members agreed on the actions being taken.

The three solution groups coordinated phases of chatting, moving, and creating/deleting differently. In the good solutions, the approaches of two pairs were dissimilar. Group 8 collaborated by alternating chat phases, move phases, and creation/deletion phases. Group 5 alternated between move phases and creation/deletion phases, and primarily communicated visually. In both groups, adjacent phases referred to the same objects and levels of abstractions, displaying a coherent problem-solving strategy. Both pairs were the only groups to have more move actions than chat actions. The incomplete groups had fewer move actions, longer phases of chatting, and fewer deletions. They adopted the inefficient strategy of discussing many future actions, creating a single edge, and then repeating the discussion. On the other hand, they would consistently reorganize objects before creating edges, which may have contributed to the tree-like organization of the classes. The poor groups engaged in long phases of chatting or moving, but showed less coherence between the objects they were discussing, objects they were creating, and objects they were moving. They deleted a lot of edges and tended to create particular edges before repositioning them. This disorganized approach probably led to the conceptual and visual disorganization of their final solutions. Another difference was in terms of the selection of objects: Both the good and the incomplete solutions would focus on objects based on their level of abstraction; they would manipulate a given superclass and all its subclasses, and then move on to another group of objects. On the other hand, students in the poor solutions appeared to let the problem organization guide their actions. They would draw edges based on classes that were close to one other in the shared workspace, rather than classes that were semantically related. Differences in object selection probably related to differences in the consistency of the final solutions.

These results indicate that we should focus on five elements when evaluating students' performance (at least on this particular task): *conceptual understanding, visual organization, task coherence, task coordination,* and *task selection,* (see Table 1). These elements were chosen because they represent different relevant aspects of student action that appeared to inform groups' solutions. Students within each solution type (good, incomplete, and poor) tended to make the same types of mistakes within each of these categories.

Conceptual understanding refers to a pair's ability to correctly place the inheritance and composition edges, while visual organization refers to a pair's ability to visually arrange the classes and edges in an appropriate manner. These two elements are linked; conceptual understanding of the problem was often reflected in the visual organization, and therefore conceptual steps tended to parallel organizational steps. For example, students in the incomplete solution groups appeared to believe that they could create only one inheritance or composition link extending from each object, and their solutions thus tended to take on rigid tree structures.

	Conceptual Understanding	***Visual Organization***	***Task Coherence***	***Task Coordination***	***Task Selection***
Good Solutions	Translation mistakes	Based on abstractions	Adjacent phases referred to the same objects	Balanced phases and work distribution	Based on abstractions
Incomplete Solutions	Overly restricted definition of super class / subclass relationships	Based on a rigid, tree-like structure	Chat phases referred to far more objects than subsequent phases	Strict turn-taking and overly long chat phases	Based on abstractions
Poor Solutions	Inconsistent definition of super class / subclass relationships	Disorganized	Little correspondence between selected objects in adjacent phases	More informal turn taking and overly long phases (particularly of deletion)	Based on proximity

Table 1: Solution Types and Elements of Analysis

Task coherence, task coordination, and task selection reflect student strategies for collaborating on the problem. Task coordination refers to skills in coordinating actions, without reference to the content of the actions. It includes distributing the work among group members, and spending appropriate amounts of time in each phase. The good groups exhibited successful task coordination, in part because they spent more time moving objects than talking about them. Task coherence refers to the appropriateness of the content of student actions. Students in the incomplete groups showed poor task coherence by chatting at length about many different links and then creating a single link. Task selection refers to a student's ability to set subgoals for solving the problem by drawing edges between classes in a sensible order. A breakdown in task selection leads to disorganized and incoherent solutions, as seen in the poor group. These three skills should be assessed by looking at the problem-solving process.

IMPLICATIONS FOR IMPROVING THE BND METHODOLOGY AND CONCLUSIONS

One of the key goals of this study was to determine how to enhance the BND methodology to provide more helpful data and analysis. In theory, the BR should be able to facilitate analysis of all the skills from Table 1. However, these five elements are not supported in the current BR, making it difficult to classify behavior and provide tutoring support. In an attempt to produce convergent paths in the behavior graphs, we restricted input to the BR to creation and deletion actions. Unfortunately, analyzing creation and deletion appears too limited to be useful. Although the BR records sequences of actions at a single level of generality, the nature of this problem indicates that student skills at different levels of abstraction need to be addressed, and the BR needs to be able to create hierarchies of behavior graphs.

We intend to address this problem by modifying the BR to support recording at different levels of abstraction. Single chat, move, or creation/deletion actions, made by a particular user and referring to a particular object, are at the lowest level of abstraction. Separate behavior paths can be generated for the creation/deletion and move actions, and used for analyzing the conceptual understanding and visual organization. The middle level of abstraction involves the analysis of *phases* of action, or chains of the same type of action, and can deal with task coherence and task coordination. The highest abstraction level addresses sequences of phases, or the characteristics of the current phase in relation to previous and subsequent phases. Skills related to task selection will be evaluated and supported at this level. An approach to classifying actions and action sequences in Cool Modes has been described in Harrer and Bollen (2004), and can be used to process actions at different levels of abstraction.

Given these changes, we believe the modified BR will be a much more effective tool for analyzing collaboration and providing tutoring support in future experiments. Besides the preliminary study we have already performed, we plan to perform two more experiments in the near term. In these studies, we will test how students collaborate to solve a Petri Net problem. We wish to determine how well the five elements of analysis we have uncovered generalize to other graphical collaboration tasks and to less structured problems. We then intend to enhance the BND methodology to allow us to compare and classify student problem solving strategies. The resulting annotated behavior graphs will provide the basis for cognitive tutor development within Cool Modes.

ACKNOWLEDGMENTS

We thank Ken Koedinger who was responsible for bringing us all together and for some of the early ideas. Partial support was provided by Deutsche Forschungsgemeinschaft (DFG) and Kaleidoscope "Network of Excellence" (IST-507838).

REFERENCES

Aronson, E., Blaney, N., Stephin, C., Sikes, J., & Snapp, M (1978). The Jigsaw Classroom. Sage Publishing Company. Beverly Hills, CA.

Bollen, L, Harrer, A., & Hoppe, U. (2004). An Integrated Approach for Analysis-Based Report Generation. *Proceedings of Advanced Learning Technologies* (ICALT 2004), IEEE Press, Los Alamitos, CA, USA, pp. 1094-1095.

Goodman, B., Hitzeman, J., Linton, F., & Ross, H. (2003) Towards Intelligent Agents for Collaborative Learning: Recognizing the Role of Dialogue Participants. In *Proceedings of the International Conference on AI in Education*.

Harrer, A. & Bollen, L. (2004) Klassifizierung und Analyse von Aktionen in Modellierungswerkzeugen zur Lernerunterstützung. In Workshop-Proceedings Modellierung 2004 . Marburg, 2004.

Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B. M., & Hockenberry, M. (2004) Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration. In *Proceedings of ITS*, Maceio, Brazil, 2004.

Koedinger, K. R. & Terao, A. (2002) A cognitive task analysis of using pictures to support pre-algebraic reasoning. In C. D. Schunn & W. Gray (Eds.), *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, 542-547.

Lesgold, A., Katz, S., Greenberg, L., Hughes, E., & Eggan, G. (1992) Extensions of Intelligent Tutoring Paradigms to Support Collaborative Learning. In S. Dijkstra, H. Krammer, & J. van Merrienboer (Eds.), *Instructional Models in Computer-Based Learning Environments*. Berlin: Springer-Verlag, 291-311.

McLaren, B. M., Koedinger, K. R., Schneider, M., Harrer, A., & Bollen, L. (2004a) Bootstrapping Novice Data: Semi-Automated Tutor Authoring Using Student Log Files; *Proceedings of the Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes*, ITS 2004

McLaren, B. M., Koedinger, K. R., Schneider, M., Harrer, A., & Bollen, L. (2004b) Toward Cognitive Tutoring in a Collaborative, Web-Based Environment; *Proceedings of the Workshop of AHCW 04*, Munich, Germany, July 2004.

Pinkwart, N. (2003) A Plug-In Architecture for Graph Based Collaborative Modeling Systems. In U. Hoppe, F. Verdejo & J. Kay (eds.): *Proceedings of the 11th Conference on Artificial Intelligence in Education*, 535-536.

Suthers, D. D. (2003). Representational Guidance for Collaborative Learning. In the *Proceedings of Artificial Intelligence in Education* (AIED-03), IOS Press, Amsterdam.

Wenger, E. (1987) *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann Publishers, Inc.