In the Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED 2005)
Amsterdam, the Netherlands, 18-22 July, 2005

Collaboration and Cognitive Tutoring: Integration, Empirical Results, and Future Directions

Andreas HARRER[+], Bruce M. MCLAREN[*],
Erin WALKER[*], Lars BOLLEN[+], Jonathan SEWALL[*]

[+]*University Duisburg-Essen, Duisburg, Germany*
{harrer, bollen}@collide.info
[*]*Carnegie Mellon University, Pittsburgh, PA USA*
{bmclaren, sewall}@cs.cmu.edu, erinwalk@andrew.cmu.edu

Abstract. In this paper, we describe progress we have made toward providing cognitive tutoring to students within a collaborative software environment. 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 provides a means to capture data that acts as the foundation of a tutor for collaboration but does not yet fully support actual tutoring. Second, we've performed two exploratory studies in which dyads of students used our software to collaborate in solving modelling tasks. These studies uncovered five dimensions of observed behavior that point to the need for abstraction of student actions to better recognize, analyze, and correct collaborative steps in problem solving. We discuss plans to incorporate such analyses into our approach and to extend our tools to eventually provide tutoring of collaboration.

1. Introduction

Cognitive Tutors, a particular type of intelligent tutor that supports "guided learning by doing" [1], have been shown to improve learning in domains like algebra and geometry by approximately one standard deviation over traditional classroom instruction [2]. So far, cognitive tutors have been used only for one-on-one instruction—a computer tutor assisting a single student. We seek to determine whether a cognitive tutoring approach can support and improve learning in a collaborative environment.

Collaboration is recognized as an important forum for learning [3], and research has demonstrated its potential for improving students' problem-solving and learning [e.g., 4, 5]. However, collaboration is a complex process, not as constrained as individual learning. It raises many questions with respect to cognitive tutoring: Can a single-student cognitive model be extended to address collaboration? Can a cognitive tutor capture and leverage the data available in a collaborative scenario, such as chat between mutiple students? What types of collaborative problems are amenable to a cognitive tutoring approach?

To take a step toward addressing these questions, we have integrated and begun experimentation with a collaborative work environment and a cognitive tutoring tool [6]. Our initial goals are twofold. First, we capture and analyze data from live collaboration so that we can better understand how a cognitive tutor might use that data to diagnose and tutor student action in a collaborative environment. Second, we would eventually like to directly use the data we collect as the basis for the cognitive tutor model.

To that end, we have developed an approach called bootstrapping novice data (BND) in which groups of students attempt to solve problems with a computer-based collaborative tool. While they work, the system records their actions in a network representation that combines all collaborating groups' solutions into a single graph that can be used for analysis and as the basis for a tutor. To effect the BND approach we have combined two software tools: a collaborative modeling tool, Cool Modes (Collaborative Open Learning and MODEling System) [7], and a tutor authoring environment, the Cognitive Tutor Authoring Tools (CTAT) [8]. Our work has focused on data collection and analysis; actual tutoring in the collaborative context is yet to be done but will be guided by these initial findings.

In this paper, we illustrate how we have implemented the BND methodology, describe empirical work that explores a particular type of collaborative problem and tests the BND approach, and present our ideas for extending our approach both to improve analysis and to lead to our ultimate goal of providing tutoring in a collaborative environment.

2. Realization of BND: The Integration of Cool Modes and the Behavior Recorder

In our implementation, depicted in Figure 1, Cool Modes (shown on the left) provides the user interface for the student; it includes a shared workspace that all collaborating students 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 describing students' actions (e.g., "student *A* created classification link *L*") to CTAT's Behavior Recorder (or "BR," shown on the right of Figure 1), which stores the actions in a *behavior graph.* Each edge in the graph represents a single student action, and paths through the graph represent series of student actions.

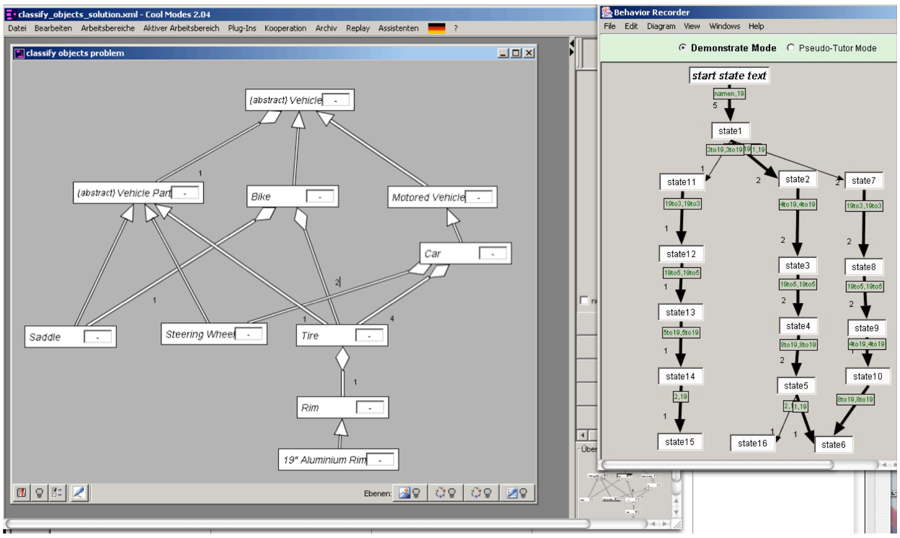

Figure 1: The student's view of the integrated Cool Modes (left) and the Behavior Recorder (right) environment. This shared Cool Modes workspace is from a vehicle classification / composition task. The behavior graph at right shows the amalgamated solutions of different collaborating groups of students.

A key aspect of the BND approach is that it counts the number of times actions are taken and displays these counts on the edges of the behavior graph. Thus, after a number of groups have used the integrated system, the behavior graph contains the actions of all student groups and reveals the frequency of common paths, both correct and incorrect. Use of this actual novice data can help to avoid part of the "expert blind spot" problem, in which experienced problem-solvers and teachers fail to identify common errors of novice students [9]. A tutor author can then use the BR to create a problem-specific tutor (or pseudo tutor, [8]) directly from the graph by labeling edges with hints and buggy messages.

We have integrated Cool Modes and the BR in a loosely-coupled fashion. Both tools remain fully operational by themselves, but can exchange messages bidirectionally using the MatchMaker communication server [10] and a "Tutor Adapter" (see Figure 2). Our earlier implementation provided one-way communication, which could support the recording of student actions but not tutoring [6]. Now, a student action causes the Cool Modes client to send an event to the MatchMaker server, which sends this event to the Tutor Adapter, which in turn forwards the event to the BR. If an author were to create a pseudo tutor and switch the BR from recording to tutoring mode, then it would respond to incoming events by sending bug messages and hints to the appropriate student or students.

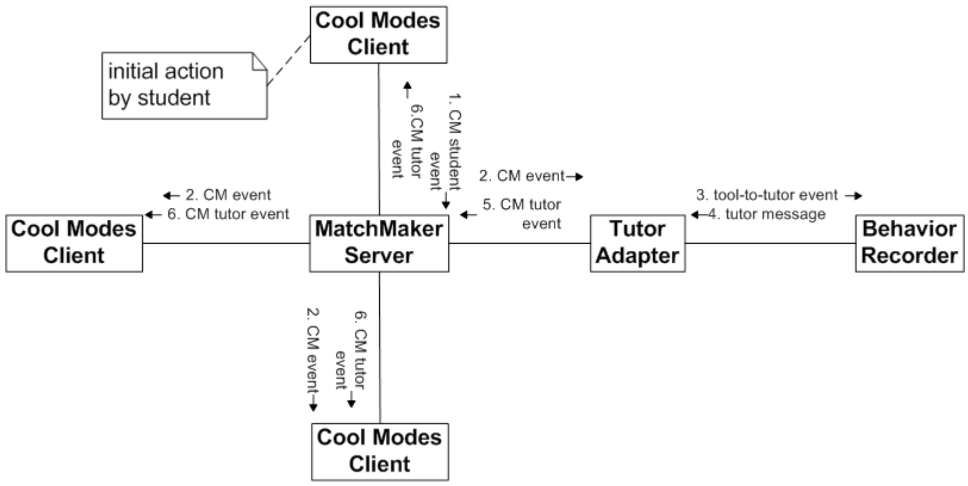

Figure 2: Collaboration diagram showing the message flow between Cool Modes and Behavior Recorder.

There are two key advantages to the BND approach. First, direct capture of student data for use in tutor building is a powerful idea. While student data has been used to guide tutor design [11] and tune tutor parameters [12], it has not been used directly as input for building an intelligent tutor. The potential time savings in data collection, data analysis, and tutoring with a single integrated tool could be significant. Second, given the complexity of collaborative learning, we thought that a 2-D visualization, in the form of a behavior graph, might allow for a better understanding and analysis of collaborative behavior when compared with, for instance, a non-visual, linear representation such as production rules.

3. Using the Behavior Recorder to Analyze Collaboration

The BR was originally designed for single-student tutoring of well-defined problems (e.g., mathematics, economics), which tend to have less possible correct and incorrect actions. In more open-ended collaborative problems, however, there are many possible sequences and alternative actions, and a given action may be appropriate in one context but not another. In this situation, a single behavior graph containing student actions is hard to interpret because higher-level processes like setting subgoals are not represented, and it is difficult to compare solutions, since on an action-by-action level most solutions will appear to be completely different. Additionally, larger group sizes also increase the state space of the Behavior Graph, because of different, yet potentially semantically equal sequences of actions by different users. Thus, early on it appeared to us that the BR would need to be extended using multiple levels of abstraction to handle the increased complexity of collaborative actions.

In preliminary experimentation with Cool Modes collaboration, we were able to identify five common dimensions of student action: *conceptual understanding, visual organization, task coordination, task coherence,* and *task selection.* Conceptual understanding refers to a pair's ability to successfully complete the task, while visual organization refers to a pair's ability to visually arrange the objects involved in an appropriate manner. Task coordination refers to skills in coordinating actions in the problem, without reference to the content of the actions. It includes sharing the work between all group members, and knowing what type of action to take at a given time (i.e., knowing when it is a good idea to reorganize the objects involved in the problem). Task coherence refers to the strategic appropriateness of the content of student actions, dealing with both task-oriented content (i.e., do adjacent phases of action deal with the appropriate objects) and collaborative content (i.e., are students providing good explanations to each other). Finally, task selection refers to students' abilities to set task-oriented and collaborative subgoals for solving the problem.

In order for the BR to process these five dimensions, it needs to handle actions at different levels of abstraction. Conceptual understanding and visual organization can be dealt with on an action-by-action basis. On the other hand, task coordination and task coherence are best evaluated through the analysis of *phases* of action, or chains of the same type of action. A chain of chat actions followed by chain of creation actions would indicate that, on a task coordination level, students have decided to discuss what objects they should create and then create some objects. This type of information is difficult, if not impossible, to extract from an action-by-action representation. Finally, task selection can be analyzed in the BR by aggregating multiple phases of action which represent high-level goals.

4. Empirical Studies

We performed two experiments to explore our assessment of the information required by the BR. Each experiment involved a visual modelling problem and tested the effect of the initial organization of objects on the collaborative problem-solving effort. In Experiment 1, we established these five elements of collaboration as relevant to the Cool Modes classification problem, and showed the need for adding support for different levels of abstraction to the BR. In Experiment 2, we verified that the five elements of collaboration are generalizable to a CoolModes Petri Net problem, and explored how the five elements could be analyzed and tutored using the BR. We will summarize the results of Experiment 1 (for a more detailed description see [13]) and describe the results of Experiment 2 in detail.

4.1 Experiment 1

In this experiment we asked 8 dyads of students to collaborate on solving a classification / composition problem (depicted in Figure 1). 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 links between objects. There were two conditions: in the *ordered* condition, the initial presentation showed related objects visually close to one another, to provide a well-organized display of the desired final network; in the *scrambled* condition, objects were positioned randomly. Groups 1 to 5 were in the scrambled condition; groups 6 to 8 were in the ordered condition. The results of the first experiment are summarized in Table 1.

The five dimensions of analysis illustrated positive and negative strategies of the participants as they related to the quality of the final solutions. Additionallly, the dimensions highlighted the connection between the organization of the start state and participants' conceptual understanding and collaborative processes.

Table 1: Solution Types and Dimensions of Analysis

	Groups 5 and 8	*Groups 2,6, and 7*	*Groups 1, 3, and 4*
Conceptual Understanding	**Good** – only trivial mistakes	**Incomplete** – only one link extended from each class	**Inconsistent** – too many links extended from each class
Visual Organization	**Good** - based on abstractions	**Overly organized** – had a tree-like structure	**Disorganized** – had long, intersecting links
Task Coordination	**Good** – good alternation of phases and distribution of work	**Hesitant –** long chat phases, formal turn-taking structure	**Impulsive –** creation before organization, informal turn-taking.
Task Coherence	**Good** - adjacent phases referred to similar objects and levels of abstraction.	**Good** - adjacent phases referred to similar objects and levels of abstraction.	**Poor** – adjacent phases referred to different objects
Task Selection	**Good** - based on abstractions	**Good** - based on abstractions	**Poor** - based on visual proximity

4.2 Experiment 2

We asked 8 dyads to solve a traffic light modelling problem using the Cool Modes / BR integrated system. Students were asked to model the coordination of car and pedestrian lights at a given intersection using Petri Nets (i.e., they were asked to draw links between traffic lights and transitions). Students could take chat, move, and creation/deletion actions, as in Experiment 1, but also *simulation* actions, firing transitions to move from one state to another. In the ordered condition of Experiment 2, the objects were organized like real-world traffic lights, with the car lights on one side, the pedestrian lights on the other side, and the transitions in the middle. In the scrambled condition, objects were placed randomly in the workspace.

We were again able to analyze the results using the five dimensions. To evaluate *conceptual understanding*, solutions were rated on a 9-point scale based on the requirements of the problem (e.g., during a simulation, the solution should never have pedestrians and cars moving at the same time). The scrambled group had significantly better solutions than the ordered group (Ms = 5.25 and 1.75). Solutions could be further divided into good (groups 1 and 2, M = 6.5), mediocre (groups 3, 4, and 5, M = 3.7), and poor (groups 6, 7, and 8, M = 1.3). The scrambled group had two good and two medium solutions, and the ordered group had one medium and three bad solutions.

The *visual organization* of the final solutions can be described in terms of two competing schemes: "real-world" (i.e., separating the car and pedestrian lights and arranging them in red/yellow/green order) versus “easy-to-follow” (i.e., having minimal edge crossings). A real-world scheme meant that the best place for the transition links were in the center of the shared visual space, creating confusing solutions because links intersected and extended in many different directions. In the ordered start state, the ideal solution corresponded to the real world, but was not easy-to-follow. Three out of the four ordered groups did not significantly reposition the objects from their original places in the start state. On the other hand, all four of the groups in the scrambled condition moved objects from their initial disorganized state to good final solutions that were relatively easy to follow. It appears that our conception of an "organized" condition may not have been as well founded for this particular problem, since an easy-to-follow arrangement seemed to relate to better solutions than a real-world arrangement.

The results for the *task coordination* differed significantly between good and bad solutions. Good groups had a significantly fewer percentage of chat actions than mediocre and poor groups (Ms = 12%, 48%, and 44%), and a significantly lower percentage of chat phases (Ms = 20%, 40%, and 39%). The good groups and the two mediocre groups in the scrambled condition also had a significantly higher percentage of move actions than the ordered groups (Ms = 28% and 8%) and significantly more move phases (Ms = 23% and 11%). There was some statistical evidence that the ordering of phases also had an effect on whether groups did well or poorly, with the optimal sequence of phases being chat->move->creation/deletion->simulation. Further, the good groups had a less balanced work distribution than the mediocre and poor groups. The ordered (and therefore less successful) groups split their time between having one person perform the whole phase (M = 37%), the other person perform the whole phase (M = 34%), or both people taking action in the phase (M = 28%). The scrambled groups had fewer phases where both people took action (M = 15%), and a less balanced distribution of individual phases (Ms = 53% and 32%). These results were surprisingly congruent with the task coordination results for Experiment 1, as reported in detail in [13].

Although *task coherence* varied between conditions in Experiment 1, there were few differences on this dimension between groups in Experiment 2. Groups refered to an average of 1.8 objects per phase in move phases, creation/deletion phases, and simulation phases. All groups tended to refer to the same objects across multiple phases.

Task selection also did not differ between groups in this experiment, but commonalities between groups provided insight into the collaborative process. Groups structured their actions based on the transitions from one state of traffic lights to the next. Creation/deletion actions were linear 79% of the time, in that the current edge being drawn involved an object used in the previous creation/deletion action. Groups tended to focus on either the pedestrian or the car lights at a given time; the current creation/deletion action tended to involve the same light class as the previous creation/deletion action 75% of the time.

In addition to the analysis of Experiment 2 based on the five dimensions, we explored how the BR could be used to analyze and tutor collaboration. For example, we used the BR to capture individual creation actions, and discovered that two groups (1 and 3) used the same correct strategy in creating the links necessary to have the traffic lights turn from green to yellow to red. This path in the graph demonstrated a conceptual understanding of how Petri Nets can be used to effect transitions. We will ultimately be able to add hints that encourage students to take this path, leveraging the behavior graph as a means for tutoring. In likewise fashion, the BR can also be used to identify common bugs in participants' action-by-action problem solving. For instance, the BR captured a common error in groups 1 and 2 of Experiment 2: each group built a Petri Net, in almost identical fashion, in which the traffic-red and pedestrian-green lights would not occur together. In situations like this, the behavior graph could be annotated to mark this sequence as buggy, thus allowing the tutor to provide feedback should a future student take the same steps.

On the other hand, it is clear that the level of individual actions is not sufficient for representing all of the dimensions. For instance, evaluating whether students are chatting "too much" or alternating phases in an "optimal" way is not easily detected at the lowest level of abstraction. To explore how we might do more abstract analysis, we wrote code to pre-process and cluster the Cool Modes logs at a higher level of abstraction and sent them to the BR. Figure 3 shows an example of this level of analysis from Experiment 2. Instead of individual actions, edges in the graph represent phases of actions (see the "CHAT", "MOVE", and "OBJEC" designations on the edges). The number to the right of each phase in the figure specifies how many instances of that particular action type occurred during consecutive steps, e.g., the first CHAT phase, starting to the left from the root node, represents 2 individual chat actions. The graph shows the first 5 phases of groups 2, 3, 5, and 8. Because the type of phase, the number of actions within each phase, and who participates (recorded but not shown in the figure), is recorded we can analyze the data and, ultimately, may be able to provide tutor feedback at this level. For instance, notice that the

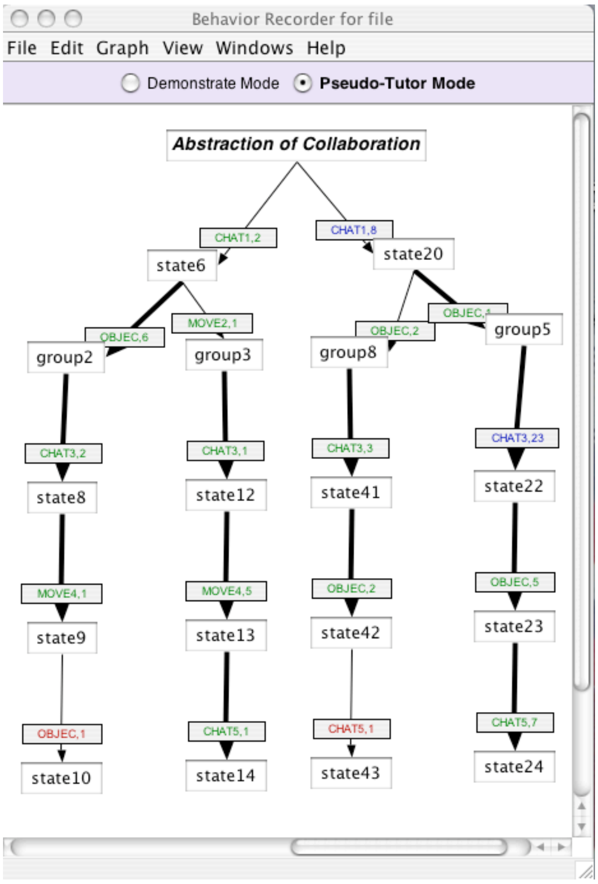

Figure 3. An Abstracted Behavior Graph

scrambled groups (2 and 3) incorporated move phases into their process, while at the same point, the organized groups (5 and 8) only used CHAT and OBJEC (i.e., creation/deletion) phases. Additionally, groups 5 and 8 began their collaboration with a lengthy chat phase, and group 5 continued to chat excessively (23 chat actions by group 5 leading to state22!). This level of data provided to the BR could help us to understand better the task coordination dimension. In addition, if provided at student time, the BR could also provide feedback to groups with "buggy" behavior; for instance, a tutor might have been able to intervene during group 5's long chat phase. In future work, we intend to further explore how this and other levels of abstraction can help us address not only the task coordination dimension but also the task coherence and task selection dimensions.

4.3 Discussion

There are two questions to answer with respect to these empirical results: Were the five dimensions valid units of analysis across the experiments? Can the BR analyze the dimensions and, if not, can the dimensions be used to guide extensions to it? The dimensions did indeed provide a useful analysis framework. The conceptual understanding dimension was helpful in evaluating problem solutions; in both experiments we were able to identify and rate the dyads based on salient (but different) conceptual features. Visual organization was important in both tasks, and appeared to inform problem solutions. The task coordination dimension provided valuable data, and the clearest tutoring guidelines of all the dimensions. The task coherence dimension provided information about object references in Experiment 1, but was not as clear of an aid in the analysis of Experiment 2. Finally, the task selection dimension was a useful measure in both experiments, but was more valuable in Experiment 1 due to the greater number of possible strategies.

With the introduction of abstraction levels, the effort to provide hints and messages to links will be greatly reduced because of the aggregation of actions to phases and sequences of phases. Even with abstraction, larger collaboration groups would naturally lead to greater difficulty in providing hints and messages, but our intention is to focus on small groups, such as the dyads of the experiments described in this paper.

5. Conclusion

Tackling the problem of tutoring a collaborative process is non-trivial. Others have taken steps in this direction (e.g., [14, 15]), but there are still challenges ahead. We have been working on capturing and analyzing collaborative activity in the Behavior Recorder, a tool for building Pseudo Tutors, a special type of cognitive tutor that is based on the idea of recording problem solving behavior by demonstration and then tutoring students using the captured model as a basis. The work and empirical results we have presented in this paper has led us to the conclusion that BR analysis needs to take place at multiple levels of abstraction to support tutoring of collaboration.

Using the five dimensions of analysis as a framework, we intend to continue to explore ways to analyze and ultimately tutor collaborative behavior. We briefly demonstrated one approach we are exploring: clustering of actions to analyze phases (of actions) and sequences of phases. Since task coordination appears to be an interesting and fruitful analysis dimension, we will initially focus on that level of abstraction. Previously, in other work, we investigated the problem of automatically identifying phases by aggregating similar types of actions [16] and hope to leverage those efforts in our present work. An architectural issue will be determining when to analyze (and tutor) at these various levels of abstraction. Another direction we have considered is extending the BR so that it can do "fuzzy" classifications of actions (e.g., dynamically adjusting parameters to allow behavior graph paths to converge more frequently).

We are in the early stages of our work but are encouraged by the preliminary results. We plan both to perform more studies to verify the generality of our framework and to implement and experiment with extensions to the Behavior Recorder.

References

[1] Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, *4*, 167-207.

[2] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8*, 30-43.

[3] Bransford, J. D., Brown, A. L., , & Cocking, R. R. (Eds.). (2000). *How people learn: Brain, mind, experience, and school.* Washington, DC: National Academy Press.

[4] Slavin, R. E. (1992). When and why does cooperative learning increase achievement? Theoretical and empirical perspectives. In R. Hertz-Lazarowitz & N. Miller (Eds.), *Interaction in cooperative groups: The theoretical anatomy of group learning* (pp. 145-173). New York: Cambridge University Press.

[5] Johnson, D. W. and Johnson, R. T. (1990). Cooperative learning and achievement. In S. Sharan (Ed.), *Cooperative learning: Theory and research* (pp. 23-37). New York: Praeger.

[6] 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.

[7] 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.

[8] 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 *Proceedingsof ITS*, Maceio, Brazil, 2004.

[9] Nathan, M., Koedinger, K., and Alibali, M. (2001). Expert blind spot: When content knowledge eclipses pedagogical content knowledge. Paper presented at the *Annual Meeting of the AERA*, Seattle.

[10] Jansen, M. (2003) Matchmaker - a framework to support collaborative java applications. In the *Proceedings of Artificial Intelligence in Education* (AIED-03), IOS Press, Amsterdam.

[11] 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.

[12] Corbett, A., McLaughlin, M., and Scarpinatto, K.C. (2000). Modeling Student Knowledge: Cognitive Tutors in High School and College. *User Modeling and User-Adapted Interaction*, 10, 81-108.

[13] McLaren, B. M., Walker, E., Sewall, J., Harrer, A., and Bollen, L. (2005) Cognitive Tutoring of Collaboration: Developmental and Empirical Steps Toward Realization; Proceedings of the *Conference on Computer Supported Collaborative Learning*, Taipei, Taiwan, May/June 2005.

[14] Goodman, B., Hitzeman, J., Linton, F., and Ross, H. (2003). Towards Intelligent Agents for Collaborative Learning: Recognizing the Role of Dialogue Participants. In the *Proceedings of Artificial Intelligence in Education* (AIED-03), IOS Press, Amsterdam.

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

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