Computer Supported Interaction Analysis of Group Problem Solving

Martin Mühlenbrock, Ulrich Hoppe

University of Duisburg, Department of Mathematics/Computer Science

Abstract: Collaboration is widely regarded as beneficial for learning. Collaboration is the mutual engagement of participants in a coordinated effort to solve a problem together. It has been conclusively argued that a focus on the *process* of collaboration is necessary in order to understand the value of working together with peers for learning.

In accordance with the approach of open distributed learning environments (ODLEs), we have developed a framework system for computer-supported cooperative learning and working. This distributed shared workspace system is well suited for monitoring and assessing group interaction in co-located and distant scenarios. It provides adaptable mechanisms for the automated micro-analysis of processes of collaboration as well as for visualization and feedback.

The system has been used in determining conflicts in focus setting as well as initiative shifts in aggregation and revision phases during a sample of collaborative sessions on puzzle problem solving.

Keywords: problem-based learning, computer-mediated communication, discourse analysis

Introduction

Recently, the nature of collaboration for learning and the dynamics of group interactions in learning environments have gained considerable interest (Baker & Lund, 1996, Burton, Brna, & Pilkington, 1997). For single user learning environments, Akhras & Self (1996) have laid out a formal framework for modeling the *process* of learning on the basis of a micro-analysis of user-environment interactions. We propose an approach for the detailed assessment of *group interactions* in a multi-user learning environment.

Since the advent of computer supported collaborative work, the investigation of computer support for collaborative learning has been of major interest. It has been conclusively argued that a focus on the processes of collaboration is necessary in order to understand the value working together with peers for learning. Collaboration is the mutual engagement of participants in a coordinated effort to solve a problem together (Roschelle & Teasley, 1991).

A number of systems have been developed to support collaborative learning, e.g. by simulation, remediation or information sharing. In principle, these learning environments also offer the opportunity to gain deeper insights in the processes of collaboration by automatically monitoring and analyzing the individual traces of the interaction moves. Presenting abstracted patterns from the collaboration protocol to the collaborators themselves is supposed to stimulate them to reflect on their interaction as well as to provide feedback to tutors.

In this paper, we present an approach and a system for the automated microanalysis of two or more persons' problem solving. In accordance with the approach of open distributed learning environments (ODLEs) first proposed in Mühlenbrock, Tewissen, & Hoppe (1997), we have developed a framework system for computer-supported cooperative learning and working. This shared workspace system seems well suited for monitoring and assessing group interaction in co-located and distant scenarios. It provides adaptable mechanisms for automated analysis of processes of collaboration as well as for visualization and feedback.

This paper is organized as follows. In the following section, different approaches on the assessment of human-system and computer supported human-human interaction will be discussed. Then we present in more detail the shared workspace system that constitutes the basis for the assessment of collaborative problem solving processes. This will finally be demonstrated by means of a worked out example application.

Assessing collaboration

It has been argued that developing integrated theories considering also environmental variables such as contextual, physical, and temporal factors could help in understanding and modeling human learning (Vosniadou, 1996). Along these lines, Akhras & Self (1996, 1999) have laid out a formal basis for modeling the *process* of learning, which is therefore characterized by its context, the activity involved, and its extension over time. Learning is regarded as the construction of knowledge through interacting in a world in which learners experience the domain and interpret their own experiences. For single-user learning environments, functions are formally defined in predicate logic

- for extracting information from the process of the learner interacting with the learning environment,
- for analyzing the process of learner-environment interactions,
- for determining the state of the learning process, and
- for changing the environment to enable changes in the state of the learning process.

The analysis is based upon notions of constructive, cumulative, reflective, and self-regulative courses of interaction. In addition, desired situations are predicted by the affordance of situations with respect to these notions.

Moreover, a repertoire of methods for analyzing user-system interactions has been developed in the field of plan recognition. Lewis (1988) investigated analysis-based generalizations of procedures in the context of human-computer interactions. He defined and empirically evaluated cognitively motivated heuristics such as the loose-ends heuristic. Hoppe (1993) investigated task models on file management that reflect the mapping of some set of meaningful external tasks onto the action primitives of a given device. Supportive functions recognize partial or complete plans for assessing the user's current task context and infer potential user goals. In contrast to the *GOMS* approach of task analysis (Kieras, 1994), plans can be directly nested.

Though frameworks and techniques for assessing single user interactions with systems already exist, they still have to be developed for *multi user* task analysis. Recently, work has begun on models of collaborative task-oriented discourse. These models put emphasis on group-oriented notions such as initiative changes (Guinn, 1998) and focus shifts in dialogues (Burton, Brna, & Pilkington, 1997). Baker & Lund (1996) investigated the implications of different communication interface layouts on collaboration and problem solving. Moreover, models of human collaboration form a basis for methodologies in distributed artificial intelligence and multi-agent systems and influence the design of communication protocols and collaboration mechanisms (see e.g. Huhns & Singh, 1997).

System architecture

The framework system is characterized by the provision of opportunities for group interaction, by the combination of intelligent support with interactive learning environments, by the provision of reusable components, and by a distributed multi-agent architecture (Mühlenbrock, Tewissen, & Hoppe, 1998). The system has been tested with several applications, e.g. for arithmetic exercises, puzzles, microworld robot control, Petri nets, and group discussions. Essentially, the system is composed of two parts, i.e.

- *CardBoard* user interfaces with distributed shared workspaces and semi-formal visual representations, so-called *card nets*, and
- *CardDalis* add-on components for the provision of operational semantics and intelligent support.

In the *CardBoard* user interface, the user can create several workspaces and freely add, arrange, and modify workspace elements. For each workspace, a specific set of possible elements (*cards*) is pre-defined (visual language). Cards can also be moved or copied between workspaces, and they can be connected by links to build up card nets. The contents of workspaces can be synchronized between two or more *CardBoards* for workspace sharing.
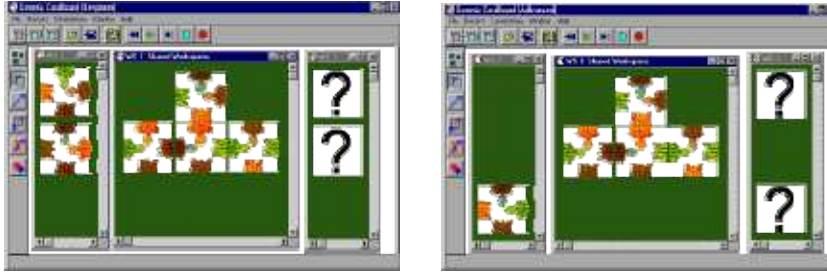
Figure 1: *CardBoard* **user interfaces with shared workspaces.**

In the example application in Figure 1, each *CardBoard* user interface contains three workspaces. The workspace in the center is shared among the two user interfaces. In this example, the task is to solve a *turtle puzzle*, i.e., to drag cards from the private palette workspaces to the shared workspace and to arrange the cards in a way to have matching turtles in color and shape. The problem can only be solved cooperatively, since the limited set of cards is initially divided up between the two or more collaborators (jigsaw design). Figure 1 shows an intermediate step of the problem solving process, where each player has dragged some cards from the private workspace on the left of his or her *CardBoard* to the shared workspace in the middle and both users have arranged the four cards in order to solve the puzzle. (The cards with a question mark can be used to activate an additional artificial player.)
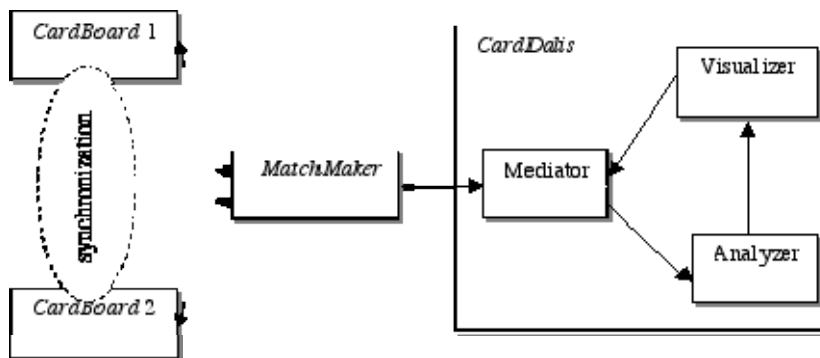

Figure 2: Architecture of the framework system.

The synchronization of shared workspaces is achieved by a replicated architecture facilitated by a synchronization manager called *MatchMaker.* Figure 2 sketches the architecture of the framework system. In order to synchronize shared workspaces, two or more *CardBoards* communicate via the central *MatchMaker* synchronization manager by TCP/IP. The communication is based on a standardized message format for card nets to minimize bandwidth. Each message contains fields for the *action* that has been performed by an *actor* on a specific *object* with an *id* and possibly further arguments (see Table 1).

Table 1: Message primitives for synchronization of workspace activities.

Further components (agents) for the provision of operational semantics and intelligent support can be added to the workspace environment by a general architecture called *Dalis* (Mühlenbrock, Tewissen, & Hoppe, 1998). *CardDalis* has been specifically developed for the automated interpretation and generation of card workspaces and their contents. Mainly, *CardDalis* comprises three components, i.e., a mediator, an analyzer, and a visualizer (see Figure 2). The mediator interfaces *CardDalis* to the *CardBoard* environment and internally represents and reconstructs the state of the workspace contents on the basis of incoming card messages as well as generates card messages to modify workspace contents itself. The analyzer takes the reconstructed data as input and derives analysis concepts on the collaboration from this data. The automatic derivation of these concepts is based on explicitly formulated rules and will be described in the next section. The visualizer processes the derived facts from the analyzer to generate further workspaces with cards that present the analysis to the collaborators or a tutor.

Task analysis of collaboration

In assessing multi user interaction and problem solving, certain aspects are of specific interest. For instance, it is likely that two collaborators have quite different methods of solving a problem at hand. The general idea is to assign collaborators to different roles depending on their initiatives in setting or shifting the focus of the problem solving process and to determine the whether next actions are chosen in accordance with the context, e.g. whether prior actions of collaborators are taken into account or are neglected.

The shared workspace system described in the previous section constitutes on ODLE that is well suited for the analysis of collaborative problem solving. The collaboration process is analyzed on the basis of an online protocol of user actions related to events in card workspaces. For instance in case of the spatial interpretation of cards and their relative positions as in the example application with the turtle puzzle (Figure 1), the analysis derives from the basic message primitives as described in Table 1 and starts with infering facts on the pairwise adjacency of cards. The adjacency of two cards is inferred from their actual positions and dimensions including some fuzziness, i.e. that adjacent cards may overlap or be apart to a certain degree. Cards that are adjacent to each other are asserted to be in the same cluster. These concepts related to the spatial arrangement of cards form the basis for the analysis of higher level concepts in a derivation hierarchy (see Figure 3).
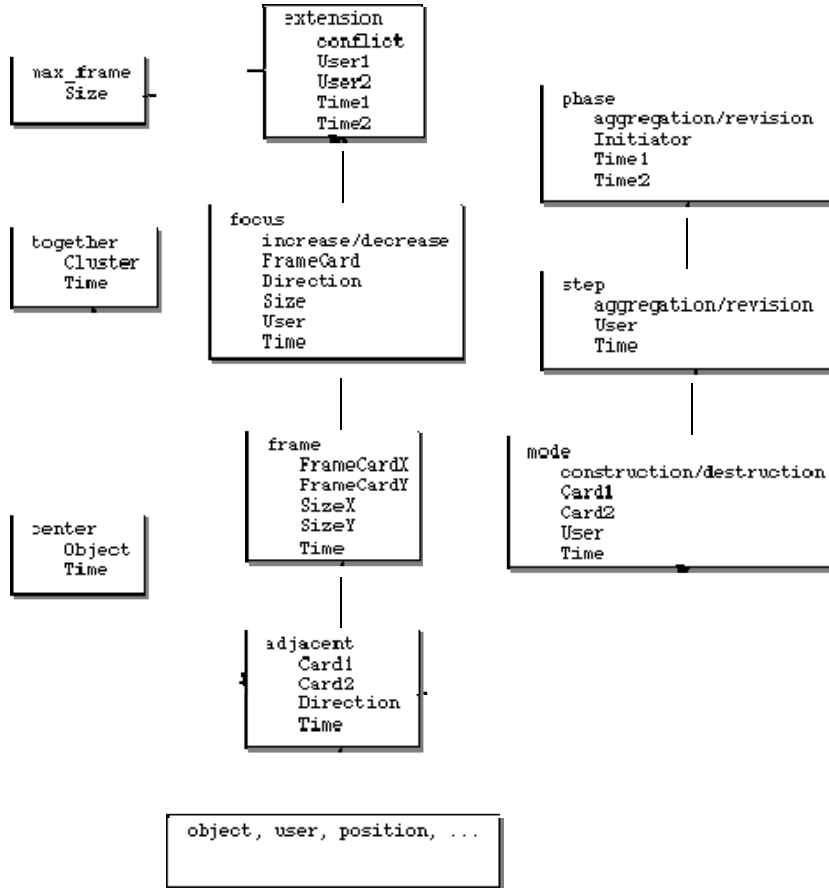
Figure 3: Derivation hierarchy of analysis concepts.

Different problem solving activities such as aggregation, conflict creation, revision, or focus shifting are assigned to the action protocol. For instance, the following rule derives the concept of a conflict in regards of different problem solving foci, e.g. in the turtle puzzle example (cf. also top of Figure 3):

```
extension(conflict,User1,User2,Time1,Time2) ←
focus(increase,FrameCard1,Direction1,Size,User1,Time1) &
focus(increase,FrameCard2,Direction2,_,User2,Time2) &
subsequent(Time1,Time2) & part_of(FrameCard1,FrameCard2) &
max_size(Size) & opposite(Direction1,Direction2)
```

According to this rule, a conflict between two users arises when they subsequently increase the outer extension of the solution, i.e. the arrangement of adjacent turtle cards, beyond a given limit and in opposite directions. For the turtle puzzle, the maximal size of the solution is three, since the final puzzle forms a square of three times three cards.

During the online processing of the action protocol, the derived facts are visualized and presented to the users through the creation of additional workspaces by the mediator.

Figure 4 shows an intermediate state of a real problem solving session where a focus conflict between the two collaborators has been detected by the rules given above.
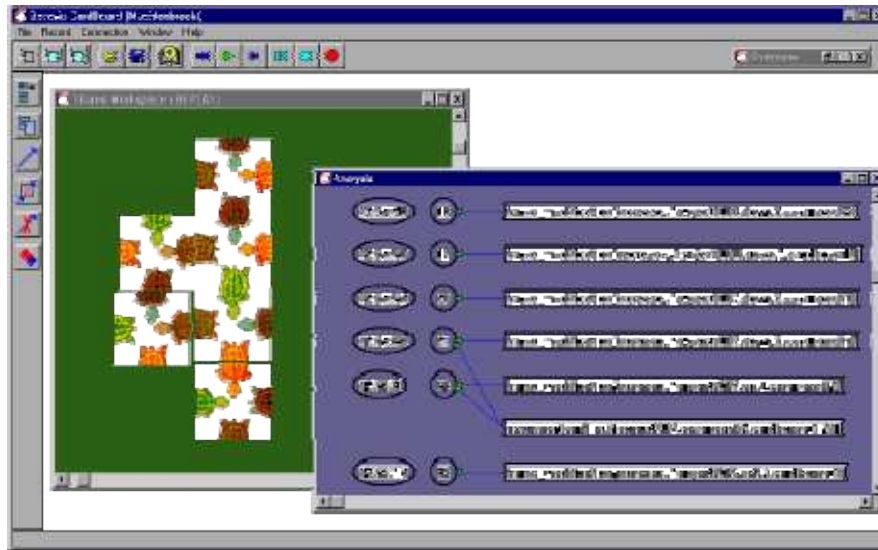

Figure 4: Visualization of analysis concepts in an additional workspace.

Furthermore, the role each user plays in the process of collaboration can generally be determined by relating initiatives and leading activities of users to problem solving steps. For instance, a user action is classified as a revision step, if the problem solution is modified in a destructive as well as constructive manner. This is usually the case when a user moves a card, which is already part of the solution, to another position to resolve a dead-end. Figure 5 and Figure 6 show the analysis of aggregation and revision steps of different collaborators. The figures have been produced with a standard spreadsheet tool directly from the data that has been generated in table format by the visualizer on the analysis.
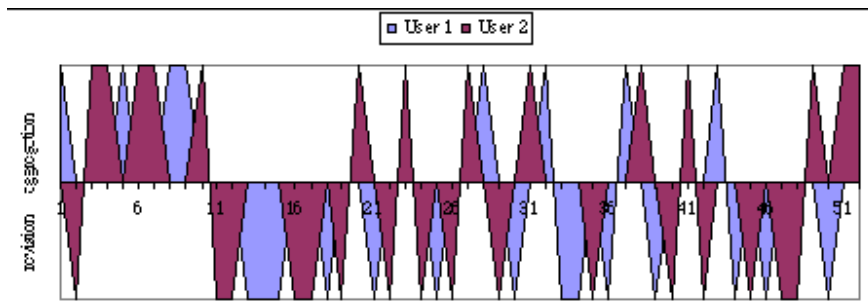

Figure 5: Aggregation and revision steps in first session of collaborative problem solving.
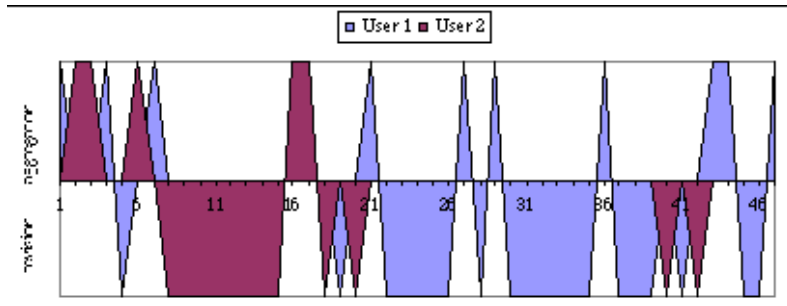
Figure 6: Aggregation and revision steps in second session of collaborative problem solving.

Both example collaborations start with a phase of aggregation steps initiated by user 1, followed by a phase of revision steps initiated by user 2. After these similar initial phases, phases of altering aggregation and revision steps occur, which are different in the two examples. Whereas in the first example, both users frequently take the initiative, in the second example the initiative remains with user 1 for the most part and aggregation and revision phases alter less frequently.

Conclusions and further work

In this paper, we have presented an approach and a system for the task-oriented analysis of collaborative problem solving. In contrast to other analysis systems, group interaction is not affected by the need to use interaction markers such as sentence openers. The framework system has been implemented in C^{++} and Prolog on a heterogeneous system platform with TCP/IP interprocess communication. A tentative set of analysis concepts including their visualization in an overview workspace has been modeled for the turtle puzzle application and has been tested with some collaborative sessions by human users.

Apart from certain concept that are specific for the turtle puzzle such as the maximum size of the solution, the analysis concepts are supposed to be applicable to diverse problems that can be solved by spatial arrangements of cards. Moreover, other structuring principles of the shared external representation such as the possibility to connect cards by typed links can be easily integrated in the derivation hierarchy in addition to the adjacency relation. Nevertheless, the approach has to be evaluated with other problems and with bigger samples of collaboration sessions.

Though the assessment of group interactions is of major interest here, immediate feedback and remediation is technically possible within the framework system. The system can be extended by rules that react to frequently occuring conflicts or to unbalanced cooperations and suggest helpful remediation. In addition, it could call the tutors attention to successful collaborations.

Currently, we are considering further techniques for task analysis in plan recognition and for analysis-based generalizations of procedures in the context of human-computer

interactions for the analysis mechanism. Further work will be dedicated to the extension and refinement of the analysis concepts that are derived. It will be investigated if validation of the concepts can be supported by audio or video protocols. This is expected to be eased by the existing time references in the action protocols.

Acknowledgements

Bibliography

Akhras, F. N. & Self, J. A. (1996). A process-sensitive learning environment architecture. In C. Frasson, G. Gauthier & A. Lesgold (editors), *Proceedings of the Third International Conference on Intelligent Tutoring Systems ITS-96* (pages 430-438). Berlin, Germany: Springer.

Akhras, F. N. & Self, J. A. (1999). Modelling the process, not the product, of learning. In S. P. Lajoie (editor), *Computers as cognitive tools: Theory change, paradigm shifts and their influence on the use of computers for instructional purposes* (volume 2). Mahwah, NJ: Lawrence Erlbaum.

Baker, M. J. & Lund, K. (1996). Flexibly structuring the interaction in a CSCL environment. In P. Brna, A. Paiva & J. Self (editors), *Proceedings of the European Conference on Artificial Intelligence in Education* (pages 401-407). Lisbon, Portugal.

Burton, M., Brna, P., & Pilkington, R. (1997). Splitting the collaborative atom: How to support learning about collaboration. In B. du Boulay & R. Mizoguchi (editors*), Artificial intelligence in education: Knowledge and media in learning systems* (pages 135-142). Amsterdam, The Netherlands: IOS Press.

Hoppe, H. U. (1993). Intelligent user support based on task models. In M. Schneider-Hufschmidt, T. Kühme, & U. Malinkowski (editors), *Adaptive user interfaces* (167--181). North-Holland: Elsevier-Science.

Huhns, M. N. & Singh, M. P. (editors). (1997). *Readings in agents.* Palo Alto, CA: Morgan Kaufmann.

Kieras, D. (1994). *A guide to GOMS task analysis*. University of Michigan.

Lewis, C. (1988). Why and how to learn why: Analysis-based generalization of procedures. *Cognitive Science*, *12*, 211--256.

Mühlenbrock, M., Tewissen, & Hoppe, H. U. (1997). A framework system for intelligent support in open distributed learning environments. In B. du Boulay & R. Mizoguchi

(editors), *Artificial intelligence in education: Knowledge and media in learning systems* (pages 191-198). Amsterdam, The Netherlands: IOS Press.

Mühlenbrock, M., Tewissen, & Hoppe, H. U. (1998). A framework system for intelligent support in open distributed learning environments. *International Journal of Artificial Intelligence in Education*, 9, 256-274.

Roschelle, J. & Teasley, S. D. (1991). The construction of shared knowledge in collaborative problem solving. In C. O'Malley (editor), *Computer supported collaborative learning* (pages 67-97). Berlin, Germany: Springer.

Vosniadou, S. (1996). Towards a revised cognitive psychology for new advances in learning and instruction. *Learning and Instruction*, 6 (2), 95-109.

Authors' addresses

Martin Mühlenbrock (muehlenbrock@informatik.uni-duisburg.de)
University of Duisburg, Department of Mathematics/Computer Science, 47048 Duisburg, Germany, Tel (+49) 203 379 3554, Fax (+49) 203 379 3557
Ulrich Hoppe (hoppe@informatik.uni-duisburg.de)
University of Duisburg, Department of Mathematics/Computer Science, 47048 Duisburg, Germany, Tel (+49) 203 379 3553, Fax (+49) 203 379 3557