From Mirroring to Guiding:
A Review of State of the Art Technology
for Supporting Collaborative Learning

AMY SOLLER

Institute for Defense Analyses

4850 Mark Center Drive

Alexandria, VA 22311 U.S.A.

phone: 703-845-2139; fax: 703-931-7792
email: asoller@ida.org

ALEJANDRA MARTÍNEZ MONÉS

Dept. of Computer Science

University of Valladolid

47011 Valladolid, Spain

amartine@infor.uva.es

PATRICK JERMANN

Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
Patrick.Jermann@epfl.ch

MARTIN MUEHLENBROCK

German Research Center for Artificial Intelligence (DFKI)
66123 Saarbruecken, Germany
Martin.Muehlenbrock@dfki.de

ABSTRACT

We review a representative selection of systems that support the management of collaborative learning interaction, and characterize them within a simple classification framework. The framework distinguishes between mirroring systems, which display basic actions to collaborators, metacognitive tools, which represent the state of interaction via a set of key indicators, and coaching systems, which offer advice based on an interpretation of those indicators. The reviewed systems are further characterized by the type of interaction data they assimilate, the processes they use for deriving higher-level data representations, the variables or indicators that characterize these representations, and the type of feedback they provide to students and teachers. This overview of technological capabilities is designed to lay the groundwork for further research into which technological solutions are appropriate for which learning situations.

From Mirroring to Guiding:
A Review of State of the Art Technology
for Supporting Collaborative Learning

AMY SOLLER[1], ALEJANDRA MARTÍNEZ MONÉS[2], PATRICK JERMANN[3], & MARTIN MUEHLENBROCK[4]

[1]*Institute for Defense Analyses, 4850 Mark Center Drive, Alexandria, VA 22311, U.S.A., asoller@ida.org*
[2]*Dept. of Computer Science, University of Valladolid, 47011 Valladolid, Spain, amartine@infor.uva.es*
[3]*Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland,*
    Patrick.Jermann@epfl.ch
[4]*DFKI, German Research Center for Artificial Intelligence, 66123 Saarbruecken, Germany,*
    Martin.Muehlenbrock@dfki.de

Abstract: We review a representative selection of systems that support the management of collaborative learning interaction, and characterize them within a simple classification framework. The framework distinguishes between mirroring systems, which display basic actions to collaborators, metacognitive tools, which represent the state of interaction via a set of key indicators, and coaching systems, which offer advice based on an interpretation of those indicators. The reviewed systems are further characterized by the type of interaction data they assimilate, the processes they use for deriving higher-level data representations, the variables or indicators that characterize these representations, and the type of feedback they provide to students and teachers. This overview of technological capabilities is designed to lay the groundwork for further research into which technological solutions are appropriate for which learning situations.

INTRODUCTION

Over the past decade, we have seen a remarkable increase in the development and adoption of network-based technologies that enable traditional and non-traditional distance learners alike to learn collaboratively. These environments enhance traditional distance learning curricula by giving students the opportunity to interact with other students online, on their own time, and wherever they are located, to share knowledge and ideas. But especially for domains in which teamwork is critical, do these collaborative tools provide the kind of supportive environments learning groups need? Is it possible to design environments in which each team of students learns in the presence of a facilitator who helps to manage and guide the collaboration, providing clear goals as to what is expected from the group process? In this paper, we review a representative selection of tools and methodologies that support collaborative learning interaction, and characterize them within a simple conceptual framework. The framework serves to organize and explain the array of available collaborative support options.

Understanding and evaluating collaborative learning tools and methodologies is not a trivial task. During collaborative learning activities, factors such as students' prior knowledge, motivation, roles, language, behavior, and group dynamics interact with each other in unpredictable ways, making it very difficult to measure and understand learning effects. This may be one reason why the focus of collaborative learning research shifted in the nineties from studying group characteristics and products to studying group process (Dillenbourg, Baker, O'Malley, & Blaye, 1995; Jermann, Soller, & Muehlenbrock, 2001). With an interest in having an impact on the group process in modern distance learning environments, the focus has recently shifted again – this time from studying group processes to identifying computational strategies that positively influence group learning. This shift toward mediating

and supporting collaborative learners is fundamentally grounded in our understanding of the group activity described by our models of collaborative learning interaction.

Because distance learners adapt their interaction to the features and capabilities of the available tools, their interaction may also differ from that of face-to-face learners, and the way in which we support their interaction may differ too. Online collaborative learning environments may never offer the same kind of supportiveness found in the face-to-face classroom, and might never need to, but they must still provide students with the kind of rich learning experiences they might otherwise obtain in the classroom. In this paper, we explore the advantages, implications, and support possibilities afforded by various technologies and computational models in an array of contexts.

We begin in the next section by describing our conceptual framework, the *Collaboration Management Cycle*. This framework will help to organize the technology support options that we describe in the third section. All four authors of this article have recently completed their doctoral dissertations in this area, and each has contributed from his or her experiences to the discussion of the critical questions and open issues for future research in the fourth and fifth sections. These sections might be used in the development of future theses, to identify key unanswered research questions and gaps.

THE COLLABORATION MANAGEMENT CYCLE

Managing collaborative interaction means supporting group members' metacognitive activities related to their interaction. It may be facilitated through activities such as providing on-line dynamic feedback to students, or off-line analyses of the students' collaboration to instructors. The students, instructors, or system might then recommend actions to help students manage their interaction by reassigning roles, addressing conflicts and misunderstandings, or redistributing participants' tasks, given their levels of expertise.

In distributed computer-supported collaborative learning (CSCL) environments, the process of collaboration management is assisted and informed by one or more computational models of collaborative learning interaction (Soller, Jermann, Muehlenbrock, & Martinez-Mones, 2004). These models provide functional computer-based representations that help us understand, explain, and predict patterns of group behavior, and support group learning processes. They can help us determine how to *structure* the environment in which the collaboration takes place, or *regulate* the student interaction during the learning activities (Jermann, Soller, & Lesgold, 2004). We very briefly describe the role of computational models in structuring the group learning environment, and then focus the remainder of our discussion on their role in regulating interaction.

The Role of Computational Models in Structuring and Regulating Interaction

Structuring approaches aim to create favorable conditions for learning by designing or scripting the situation *before* the interaction begins (Dillenbourg, 2002). For example, we might structure the learning experience by varying the characteristics of the participants, the size and composition of the group, or the definition and distribution of student roles. We might also strategically select a subset of learning tools, activities, and communication media with desired characteristics, or change the appearance of the environment based on the nature of the task (e.g. writing, problem-solving) or the configuration of the group. A computational model, describing students' prior behavior under similar conditions might be used to strategically construct learning teams and activities, or plan mediation schemes. Approaches to structuring the learning situation are often based on educational principles or theories, and intended to encourage certain types of interaction, such as argumentation or peer tutoring.

Regulation approaches support collaboration by taking actions *once* the interaction has begun. Interaction regulation is a complex skill that requires a quick appraisal of the situation based on a comparison of the current situation to a model of desired interaction. In the classroom, the regulation of student interaction is performed by a teacher, taking into account complex variables such as the observed student interaction, various experiences from years of teaching, and knowledge of the students'

personalities and typical behaviors. The difficulty in eliciting the knowledge needed to account for these complex variables, and determining the manner and degree to which each contributes to the collaborative learning outcome, has presented significant challenges to the computational modeling, analysis, and assessment of group learning activities. How might a computer assess the quality of knowledge sharing, or measure the degree of constructive conflict between students? It is too early to tell whether or not we will ever be able to offer the supportiveness of a human teacher, online; however, a few research projects have begun to explore the possibilities of enriching CSCL environments with tools to support and enhance collaboration management through interaction regulation.

Before leaving our discussion of structuring and regulating approaches, we note that these methods need not be exclusive, and may even be applied in concert. For example, a system might mediate the group by dynamically structuring the environment, while the students, at the same time, attempt to regulate their own interaction. We now move to a discussion of the four phases in the collaboration management cycle, designed to organize the array of state-of-the-art functionality for supporting interaction regulation.

Phases of the collaboration management cycle

In this section, we present a framework for describing the process of collaboration management, building upon the work of Jermann, Soller, and Muelhenbrock (2001) and Barros and Verdejo (2000). Collaboration management follows a simple homeostatic process, illustrated in Figure 1, that continuously compares the current state of interaction with a target configuration (the desired state). Pedagogical actions are taken whenever a perturbation arises, in order to bring the system back to equilibrium. Because the definition of the desired state may not be fully known, and may also change during the course of group activity, the framework presented here provides a general description of the activities involved in computer-supported collaboration management, rather than a means for predicting collaborative learning outcomes.

The framework, or *collaboration management cycle* is represented by a feedback loop, in which the metacognitive or behavioral change resulting from each cycle is evaluated in the cycle that follows. Such feedback loops can be organized in hierarchies to describe behavior at different levels of granularity (e.g. operations, actions, and activities). The collaboration management cycle is defined by the following phases:

Phase 1: Collect Interaction Data
The data collection phase involves observing and recording the interaction. Typically, users' actions (e.g. 'user1 clicked on I agree', 'user1 changed a parameter', 'user1 created a text node') are logged and stored for later processing. An important decision that must be made in phase 1 as to whether the eventual model will call for an activity-based analysis, requiring a historical log of student actions across time, or a state-based analysis, requiring the logging of "snapshots" of collaborative interaction, without history information (Gassner, Jansen, Harrer, Herrmann, & Hoppe, 2003).

Phase 2: Construct a Model of Interaction
The next phase involves selecting and computing one or more higher-level variables, termed *indicators*, to represent the current state of interaction. For example, an agreement indicator might be derived by comparing the problem solving actions of two or more students, or a symmetry indicator might result from a comparison of participation indicators.

Phase 3: Compare the Current State of Interaction to the Desired State
The interaction can then be "diagnosed" by comparing the current state of interaction to a desired model of interaction. We define the desired model as a set of indicator values that describe productive and unproductive interaction states. These indicators typically correspond to features of collaborative interaction which might positively influence learning. For instance, we might want learners to be verbose

(i.e. to attain a high value on a verbosity indicator), to interact frequently (i.e. maintain a high value on a reciprocity indicator), and participate equally (i.e. to minimize the value on an asymmetry indicator).
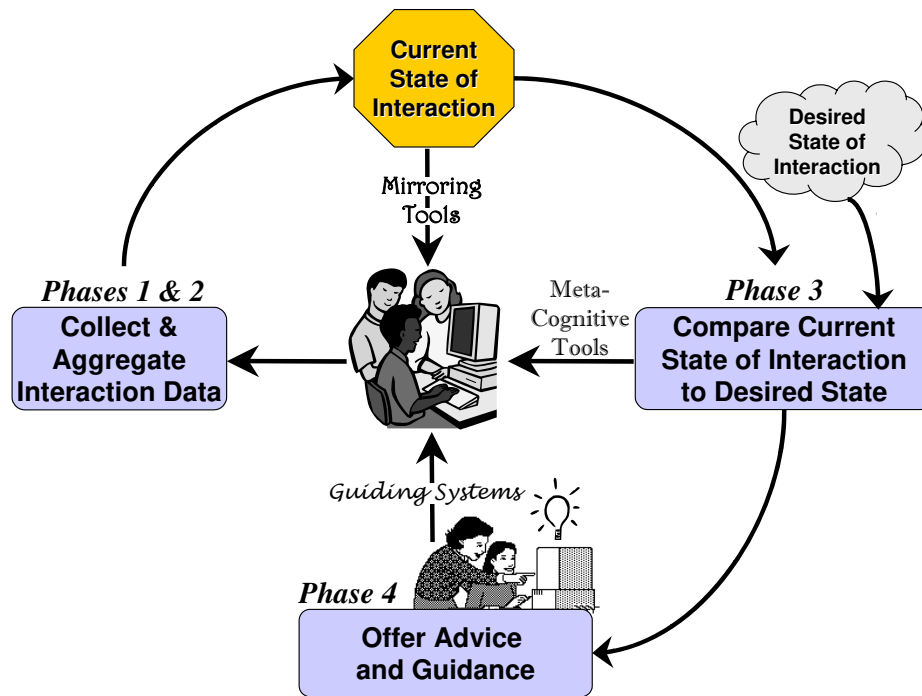

Figure 1. The collaboration management cycle, showing points at which the responsibility for analyzing and guiding the interaction might shift from the collaborators to the system.

From an implementation standpoint, the difference between phases 2 and 3 does not seem significant. From a philosophical perspective, however, these phases describe the difference between a system that reflects the group's activities back to the members, and requires them to manage their own interaction, and a system that prepares interaction data so that it can be assessed by computer models, or analyzed by researchers in an effort to understand and explain the interaction.

Phase 4: Advise/ Guide the Interaction
Finally, if there are discrepancies between the current state of interaction (as described by the indicator values) and the desired state of interaction, some remedial actions might be proposed. Simple remedial actions (e.g. 'Try letting your partner have control for a while') might result from analyzing a model containing only one indicator (e.g. word or action count), which can be directly computed from the data, whereas more complex remedial actions (e.g. 'Try explaining the concept of generalization to your partner using a common analogy') might require more sophisticated computational analysis.

Phase 4 is not the final phase in this process. Remediation by the system or human instructor will have an impact on the students' future interaction, and this impact should be re-evaluated to ensure that it produced the desired effects. The arrows that run from phase 4 back through the illustration representing the logging of learners' actions, to phase 1 indicates the cyclic nature of the collaboration management cycle, and the importance of evaluation and reassessment at the diagnostic level.

*Phase 5: Evaluate Interaction Assessment and Diagnosis*

After exiting Phase 4, but before re-entering Phase 1 of the following collaboration management cycle, we pass through the evaluation phase. Here, we reconsider the question, "What is the final objective?", and assess how well we have met our goals. Some systems are aimed exclusively at analyzing and evaluating student activity. Their objective is to explain why students may be experiencing trouble collaborating and learning, and help an instructor or online coach target those difficulties. In some cases, evaluation may be performed off-line, taking complete courses of interaction as the units of analysis. Off-line evaluation removes the temporal constraints that are present in dynamic, on-line coaching and evaluation scenarios, although such evaluation procedures also introduce some delay in the feedback, evaluation, and remediation loop. Off-line evaluation may be performed by either the system, or a human evaluator. In the first case, the system improves its own ability to diagnose student performance by directly analyzing students' actions (e.g. Soller, 2002; Soller & Lesgold, 2003). In the second case, a human may intervene in the process to alter the method of facilitation or even the model of desired interaction. In control theory and in cognitive science, cognitive architectures are described as a hierarchy of referents that starts at the level of a sensation up to the level of concepts (Robertson & Powers, 1990). The fifth phase in our model corresponds to a higher level of control: it allows adjusting the desired state of interaction in the management cycle. For the sake of simplicity, Figure 1 only represents the four first phases of the management cycle.

When these five phases are realized in a system, they might form more of a theoretical base than embody physical system components or human-controlled tasks. In some systems, the phase durations and boundaries may vary significantly, making the phases difficult to identify, whereas in other systems, the phases might be implemented as concrete, identifiable, software modules. For instance, the first phase – collection of interaction data – could be realized as either the collection of a single new datum that immediately triggers the cycle, or the accumulation of interaction data over a long period of time, that must be completed before entering the next phase. Systems that involve humans 'in the loop' who give advice or guide the interaction tend towards the latter because human resources are often not immediately available.

**The Locus of Processing: from mirroring to guiding**

Research in distributed cognition suggests that cognitive and metacognitive processes might be spread out and shared among actors in a system, where these actors may constitute both people and tools (Hutchins, 1995; Salomon, 1993). Following this idea, computers might offer support for any or all of the four phases described in the previous section. The locus of processing describes the location at which decisions are made about the quality of the student interaction, and how to facilitate this interaction. Depending on the requirements and goals of the learning activity, the locus of processing may be located anywhere on a continuum between the system, instructors, and collaborating students. For example, a teacher, or the group members themselves, might observe the interaction, compare its current state with implicit or explicitly agreed upon referents, and propose changes to the communicative rules or division of labor. In this case, the locus of processing is in human hands. Alternatively, parts of this process might be managed by a computer system, thereby shifting the locus of processing towards the computer.

Systems that collect interaction data and construct visualizations of this data tend to place the locus of processing at the user level, whereas systems that advise and coach aggregate and process this information directly. In the remainder of this section, we describe three computer-based support options that arise as the computer takes over various phases of the collaboration management process presented in the previous section.
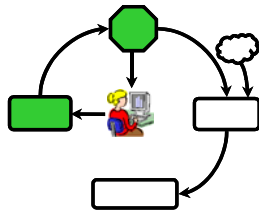
Mirroring Tools

Mirroring tools automatically collect and aggregate data about the students' interaction (phases 1 and 2 in Figure 1), and reflect this information back to the user, for example, as graphical visualizations of student actions or chat contributions. These systems are designed to raise students' awareness about their actions and behaviors. They place the locus of processing in the hands of the learners or teachers, who must compare the reflected information to their own models of desired interaction to determine what remedial actions are needed.
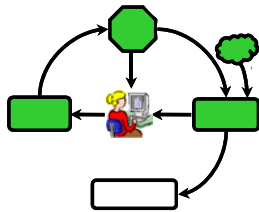
Metacognitive Tools

Metacognitive tools display information about what the desired interaction might look like alongside a visualization of the current state of indicators (phases 1, 2 and 3 in Figure 1). These systems provide the referents needed by the learners or human coaches to diagnose the interaction. Like mirroring tools, users of metacognitive support tools are responsible for making decisions regarding diagnosis and remediation.
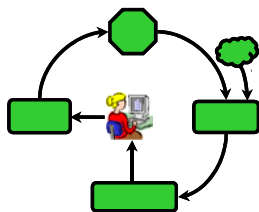
Guiding Systems

Guiding systems perform all the phases in the collaboration management process, and propose remedial actions to help the learners. The desired model of interaction and the system's assessment of the current state are typically hidden from the students. The system uses this information to make decisions about how to moderate the group's interaction.

Fundamentally, these three approaches rely on the same model of interaction regulation, in that first data is collected, then indicators are computed to build a model of interaction that represents the current state, and finally, some decisions are made about how to proceed based on a comparison of the current state with some desired state. The difference between the three approaches above lies in the locus of processing. Systems that collect interaction data and construct visualizations of this data place the locus of processing at the user level, whereas systems that offer advice process this information, taking over the diagnosis of the situation and offering guidance as the output. In the latter case, the locus of processing is entirely on the system side.

Selecting and designing the most appropriate computational approach for supporting group interaction means evaluating the instructors and learners' needs and assessing the available computational resources. Each of the three support options described in this section presents different advantages and disadvantages (described in more detail in the next section), and many combinations of approaches can be complementary. For example, imagine a system that progressively moves the locus of processing from the system side to the learner side: a guiding tool that becomes a metacognitive tool and finally a mirroring tool. As students observe the methods and standards that the system uses to assess the quality of the interaction, they might develop a better understanding of the system's process of diagnosis, allowing the responsibility for interaction regulation to be progressively handed over to the students. Once the students have understood (internalized) these standards, simply displaying the indicators in a mirroring tool might be sufficient.

Models designed to assist human instructors in coaching students might look different from those intended to guide students directly, even if the locus of processing looks similar. Dimitracopoulou & Komis (2004) found that teachers' most important consideration was their ability to track multiple groups of students as they learn synchronously, and identify individual and group difficulties. Supporting these teachers might mean providing them with automated analysis tools, targeted at their specific concerns, or tools that enable them to reconstruct and analyze sequences of past collaborative student activity.

Students, on the other hand, might initially lack the skill and insight to interpret the models correctly, and may consequently develop biases about what constitutes effective interaction. For example, students

might rely on implicit social norms (status, equality) to manage the interaction by remaining silent while their more knowledgeable peers perform difficult tasks. Partners may spend unnecessary time worrying about whether or not they are participating equally, thinking that equal participation leads to equal credit. Collaborative learners guided by mirroring and metacognitive tools may need to follow a more introspective process to develop an understanding of their interaction than those who are guided by a teacher or computer-based coach. The advantage of these tools is that those learners who struggle and succeed without intervention may more rapidly develop a understanding of their interaction, and how to improve their own interaction skills.

A REVIEW OF SYSTEMS THAT SUPPORT COLLABORATIVE LEARNING

In this section, we discuss representative examples of three types of supportive collaborative learning systems in the context of the collaboration management cycle. In the previous section, we described mirroring systems as those that reflect actions because they collect activity data in log files and display it to the collaborators. We described metacognitive tools as those that monitor the state of interaction because they maintain a model of the group activity, and either diagnose the interaction or provide collaborators with visualizations that they can use to self-diagnose their interaction. These visualizations typically include a set of indicators that represent the state of the interaction, possibly alongside a set of desired values for those indicators. Finally, we explained that coaching or advising systems guide the collaborators by recommending actions they might execute to enhance the interaction. We begin this section with a brief discussion of the options available for collecting and structuring interaction data, in preparation for collaboration analysis. We then turn to a deeper discussion of the technology options for each phase of the collaboration management cycle, and review a number of key systems within each category.

Collecting & Structuring Interaction Data

The first step in designing and developing collaboration support tools is determining how student actions should be logged by the system. This means making decisions about the granularity of data to collect (mouse movement, clicks, or object manipulation), how often actions should be logged, where (e.g. in a logfile, database, or internal data cache), and in what format. While a standard data format that would allow researchers to share and reuse analysis tools across different CSCL systems more easily, this might also limit our ability to customize tools for specific user groups, or to apply special methods for analyzing particular combinations of data. In this section, we briefly introduce the work of a few notable researchers who have seriously considered CSCL data collection issues, and then discuss how a variety of mirroring tools have taken advantage of these data collection efforts.

The Object-oriented Collaboration Analysis Framework (OCAF) (Avouris, Dimitracopoulou, & Komis, 2003) defines a model that represents the items of the students' solution (including those that have been discussed and eventually rejected) as a sequence, $P_i f_j$, where P_i represents the actor, and f_j, the *functional role* related to a particular part of the solution (e.g. the insertion or proposal of an item, the rejection of a proposal). The functional roles are determined by automatically analyzing the logs of student actions, and manually analyzing the logs of student dialog. Collecting and structuring data in this way enables the researcher to analyze student workspace actions from the point-of-view of the shared objects rather than the student actions. OCAF considers objects as entities that carry their own history, and are owned by actors (learners) who have contributed, in varying degrees, to the solution. They independently compile statistics on their use, and contribute to the definition of indicators describing their owners' collaborative behavior. Because of this *object-orientation*, OCAF is restricted to systems in which students construct solutions composed of well distinguished objects.

Avouris, Komis, Margaritis, and Fiotakis (2003) have developed a system called Synergo, which represents OCAF-modeled activity textually or diagrammatically. While the former is suitable for automatic processing, the latter is intended to provide a human with a view of the items and their history. The system also includes a web interface that logs student actions, and displays various views of the

model such as a history of events, or an object-oriented view of every object that has been inserted in the system. With the help of this tool, the researcher can inspect different aspects of the model, such as the activity of each actor, or the structure of the solution.

To facilitate the automatic data collection process, Martínez-Monés, Guerrero, and Collazos (2004) define the concept of the *collaborative action* in context as an action that can affect the collaborative process, and can be perceived by group members. They explain how a CSCL environment can model and implement collections of collaborative actions by adapting the standard software engineering *command* design pattern. The design pattern is a general solution that modularizes the data collection process, enables data customization, and can be used to define logging functionality in any type of application (not only those that are designed to mediate collaborative activity).

The storing and processing capabilities of computers have long been seen as an opportunity for research and evaluation in collaborative learning (Dillenbourg, 1999). For example, computer-generated log files may be combined with more traditional sources of data such as ethnographic observation and audio tapes. Neale and Carrol (1999) present a complex evaluation methodology that combines automatic tools with field work to evaluate distance learning activities. One modern adaptation of this concept uses web server logs as the object of analysis. The Server Log File Analyzer (SLA) (Wasson, Guribye, and Mørch, 2000) is a tool that analyzes web server logs to determine when it is possible for a team of students to collaborate synchronously (i.e. when two or more of the team's members are logged on at the same time). SLA also highlights if one team member does not log on for a significant period of time, thus identifying periods when even asynchronous collaboration is not possible. Log file analysis tools, such as SLA, act as mirroring tools by showing team members representations of their activities. In the next section, we see how these representations may help students determine what behaviors would most likely promote a successful collaboration.

Systems that Reflect Actions

The most basic level of support a system might offer involves making the students or teachers aware of participants' actions, without abstracting or evaluating these actions. Actions taken on shared resources, or those that take place in private areas of a workspace may not be directly visible to the collaborators, yet they may significantly influence the collaboration. Raising awareness about such actions through mirroring tools may help students maintain a representation of their teammates' activity. A better representation of the partner's whereabouts might allow to prevent coordination problems and enhance one's own metacognitive abilities (Amy, this means: seeing the partner's actions allows me to better reflect upon my own actions).

PENCACOLAS (PEN Computer Aided COLlAborative System) (Blasco et al., 1999), a system designed to teach collaborative writing, is an example of an environment that facilitates formal evaluation while reflecting users' actions. PENCACOLAS enables groups of students, and a teacher, to generate text synchronously. It models compositions as problem-solving situations that follow a recursive process involving a series of phases (e.g. brainstorming, planning, writing and revision). Students using the system may also interact asynchronously, by revising their peers' compositions, or exchanging short messages. PENCACOLAS records all the students' writing events. These logs are used both to analyze the student activity, and to enable the review, correction, and evaluation of previous composition phases. Reviewing students' intermediate writing steps may provide valuable insights regarding the evolution of their writing, and their cognitive development. To facilitate formative evaluation, the system automatically generates filenames that identify users, sessions, and phases, thus allowing evaluation of both collaborative and individual work. This may also allow the teacher to perform a self-evaluation in which she reviews her pedagogical interventions.

Actions may also be represented along a timeline. For example Plaisant, Rose, Rubloff, Salter, and Shneiderman (1999) describe a system in which students learn the basics of vacuum pump technology through a simulation. As the learner manipulates the controls of the simulation, he can view a history of his actions displayed graphically beneath each target variable (e.g. pressure). The display shows a series of boxes along a timeline, indicating the intervals in which the user is taking actions, and the system's

messages. The data displayed to the student does not undergo any processing or summarizing, but directly reflects the actions taken on the interface. Although Plaisant et al. did not design the system to be used by two persons at the same time, the learning history might be used to mirror a collaborative situation by displaying the actions of the learners side-by-side, and offering a representation of concurrent actions to help students coordinate their activity.

The graphical records of actions that Plaisant et al.'s (1999) system constructs might be sent to a tutor or a peer learner, or replayed by the learner to examine his own performance. Goodman, Geier, Haverty, Linton, and McCready (2001) have taken advantage of the capacity of replay and reviewing tools to serve as mirroring devices. They have developed the Asynchronous Replay Tool (ART), which when integrated in a larger system called SAILE (Synchronous and Asynchronous Interactive Learning Environment), provides support for both synchronous and asynchronous interactions. ART allows an asynchronous learner to become a full participant in a problem solving session by enabling her to replay (fast review, step-by-step, or replay action segments delimited by chat events) and experience the collaboration process of the other group members that have been working on the same problem.

Although online chat facilities pervade distance collaborative learning systems, many still present limitations, such as the lack of visual and audio cues (e.g. gestures and voice tone). Some researchers have addressed these limitations by developing creative extensions (see Looi, 2001, for an array of examples). For example, chat awareness tools such as Chat Circles (Donath, Karahalios, & Viegas, 1999), can help users keep track of ongoing conversations. Chat Circles is a graphical interface for synchronous chat communication that reveals the social structure of the conversation (see Figure 2).
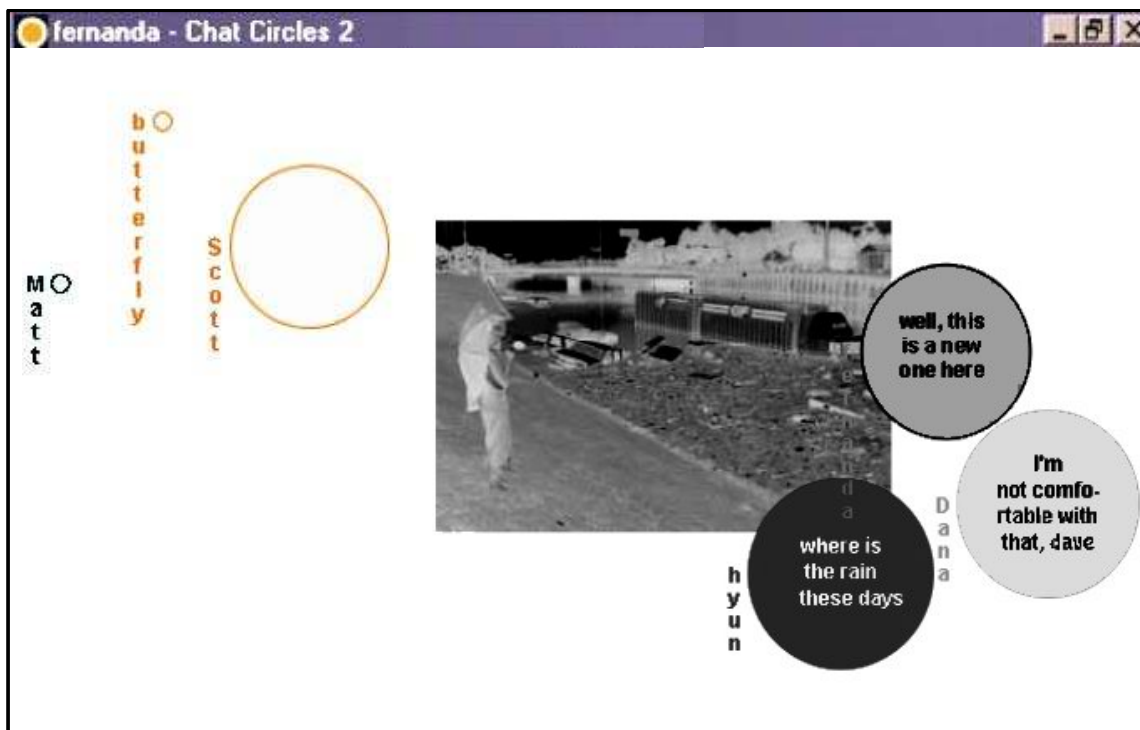

Figure 2. Chat Circles: A mirroring tool that helps users keep track of ongoing conversations Obtained by email from Fernanda Veigas (viegas@media.mit.edu). Note: figure enlarged and colors enhanced for printing. Printed with permission.

Each participant is represented by a coloured circle on the screen in which his or her words appear. The tool is based on an *auditory* metaphor: while one can see all the participants at once, one can only "hear" (that is, read the words) of those one is sufficiently close to. Distances between messages (circles) are used to represent who is talking to whom, hence the tool represents social structure through spatial

proximity. A participant's circle grows and brightens with each message that he sends, and fades in periods of silence. The circles, however, do not completely disappear while the participant is still connected to the chat. Viewed over time, Chat Circles creates a visual record of conversational patterns. Each user is made aware of the other active, animated participants and can watch the emergence and dissolution of conversational groups. The developers of Chat Circles also developed an archival interface, "Conversation Landscape", that graphically displays chat logs in an intuitive format. This format maintains the information that is normally lost in log files, such as pauses and turn-taking behavior. The conversation landscape is a two-dimensional model of the conversation, showing the postings of the participants (again identified by color) as horizontal lines. The width of the lines is proportional to the lengths of the messages. The viewer can interact with this visualization to see individual conversations, and read the postings. In this way, it is possible to explore what has happened in an intuitive way.

Other systems reflect actions, but are not geared specifically toward learning, and hence will be covered only briefly here. For example, one of the awareness tools (Gutwin, Stark, & Greenberg, 1995) in the Groupkit system (Roseman & Greenberg, 1992) contains a shared scrollbar to display the section of text each participant is looking at, allowing students to locate their partner's focus of attention. Some groupware systems use a room-based paradigm to inform users of the virtual locations of their peers. They may also show users which objects their peers are viewing or manipulating. (See NCSA Habanero, CUSeeMe, IWS, Microsoft NetMeeting, and Groove for some other examples.)

Systems that Monitor the State of Interaction

Systems that monitor the state of interaction fall into two categories: those that aggregate the interaction data into a set of high-level indicators, and display these indicators to the participants, and those that internally compare the current state of interaction to a model of ideal interaction, but do not reveal this information to the users. In the former case, the learners are expected to manage their interaction themselves, assuming that they have been given the appropriate information to do so. In the latter case, this information is either intended to be used later by a coaching agent, or analyzed by researchers in an effort to understand and explain the interaction.

Systems that Display High-Level Indicators

Our first group of systems models the state of interaction through a set of indicators that are displayed to the users. Jermann (2004) has developed a system that displays participation rates to the collaborators as they are solving a traffic light tuning problem. The indicators on the display represent the number of messages each student has sent with respect to the number of problem-solving actions he and his teammates have taken (see Figure 3). The system displays a color-coded model of desired interaction next to the observed interaction state. The colors indicate that desirable interaction includes a greater proportion of talk relative to simulation-directed actions. The students use this standard to judge the quality of their interaction and determine whether or not to take remedial actions. Jermann found that the metacognitive display encourages students to participate more through the chat interface, in particular to engage in more precise planning activities.

Tools like this might have a positive impact on a group's metacognitive activities by aiding in the construction and maintenance of a shared mental model of the interaction. This mental model may encourage students to discuss and regulate their interaction explicitly, leading to a better coordination of the joint effort to reach a solution. The notion of desirable interaction might also change during the learning process, causing the target values of the indicators to be dynamically updated, and encouraging the learners to improve in different ways.
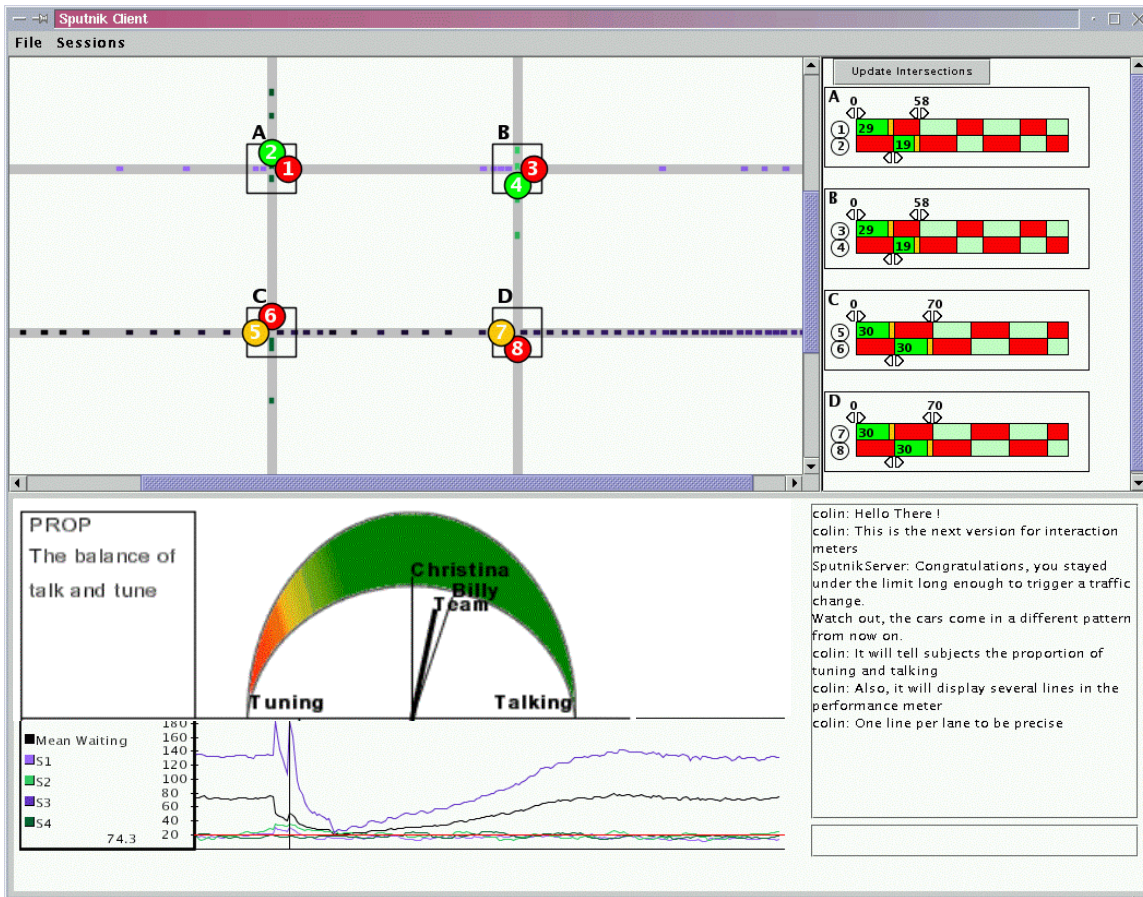
Figure 3. While students solve a traffic light tuning problem, they can visualize and compare their chat and problem solving behavior to that of their teammates. The color of the Pie ranges from red on the left side to green in the center and right side. The needles indicate the Talk Tune Proportion (TTP) for each subject (Christina and Billy) as well as the average for the group (Team).

Some metacognitive tools also include specialized displays for teachers and facilitators, to help them regulate the collaborative activities of their students. These systems usually avoid complex computation because they are designed to dynamically provide intuitive displays that make the assessment process as efficient as possible. In classrooms with hundreds of students, such tools may be invaluable in helping teachers monitor and facilitate group work. The Synergo system (Avouris, Komis, Margaritis, & Fiotakis, 2004) presented in the previous section, and the tool developed by Fesakis, Petrou, and Dimitracopoulou (2004) compute the quantitative indexes CF (Collaborative Factor) and CAF (Collaborative Action Function), respectively. Both calculate a value for collaboration by taking into consideration the actions performed by the user in the environment through the different collaboration channels (e.g. chat, shared workspace). In the classroom, teachers have described these tools as useful for monitoring the state of on-line collaborative activities, reflecting on their own activities off-line, planning new lessons, and configuring new group structures. The impact of these kinds of tools can be significant; they have the capability to focus a teacher's attention toward some groups, and away from others that may really need the extra help. In the design of these tools, especially those with a limited pedagogical basis, we should therefore remember to respect the teacher's authority in the classroom, and focus our efforts on developing tools that assist the teacher in managing the classroom, rather than tools that assume the responsibility for evaluating the collaborative activity.

Metacognitive tools designed for use in educational forums may be used to support asynchronous discussions over indefinite periods of time. Gerosa, Gomes Pimentel, Fuks, and Lucena (2004) describe

the tree-like visualization tools they have developed that compute and display the statistics and linkages between forum messages. The tools provide information about the structure of the student discourse, accounting for factors such as the length of messages related to each message category, discussion depth, percentage of leaves, percentage of messages in each category, and frequency of messages per hour. Depending on the style of interaction (reflective discussion or brainstorming), either the length of messages of their frequency is better suited to measure the intensity of interaction.

Simoff (1999) proposes an interesting way to estimate students' potential for learning by analyzing the graphical representation of student participation in an educational forum. His system uses nested boxes to visualize discussion threads. The thickness of the boxes' edges represents the number of messages produced in response to the opening message for a particular thread. In an educational environment, thicker boxes containing task-oriented content might mean deeper conversations, hence deeper understanding. To study of the content of the messages, Simoff applies a semi-automatic content analysis method. Common words such as articles and prepositions are discarded, and then the occurrences of the remaining terms are counted, noting the co-occurences of the most frequent words. This technique is used to build a seminar thesaurus, and an individual thesaurus for each participant. The comparison between these thesauri gives an indication of each participant's contribution to the seminar. The content of the discussions is also used to generate a semantic net (through the use of the Text Analyst tool), which indicates the relevance of each term.

Talavera & Gaudioso (2004) apply data mining and machine learning methods to analyze student messages in asynchronous forum discussions. Their aim is to identify the variables that characterize the behaviour of students and groups by discovering clusters of students with similar behaviours. A teacher might use this kind of information to develop student profiles and form groups. Talavera and Gaudioso's approach is general enough to be useful within many different types of collaborative environments, however the output may need further refinement to be usable by teachers, who are not expected to be experts in data mining techniques.

The last group of metacognitive tools we discuss in this section apply the formal concepts and techniques of Social Network Analysis (SNA) (Wasserman & Faust, 1996) to study and display the structure of group activity. SNA is based on strong mathematical and sociological foundations, and provides a set of methods and measurements for discovering and describing patterns of relationships among actors, and understanding how these patterns affect people and organizations. SNA methods operate on structures called *social networks* that describe a set of actors and their relationships. Social networks can take many different forms. For example, e-mail interchange networks describe which actors have sent e-mail to which other actors. Although social networks typically represent relationships between people, they can be extended to include relationships between the users and resources. Indirect interaction networks describe which actors have created and shared documents with which other actors, and who has taken actions on these shared documents. Many possibilities exist depending on the environment, the issues being studied, and the available data processing capabilities.

Social Network Analysis supports the study of the relationships at different levels, namely the individual, the group and the community. For example, at the individual level, the *degree centrality* measures the prominence of actors in the network, and helps to identify those who are the most active or most peripheral. At the group level, *cohesive subgroups* can be identified as groups of actors with strong, direct, frequent ties. And, at the community level, the *network density* computes the percentage of actual links with respect to the number of possible links in a network, describing the level of activity in the network (Wasserman & Faust, 1996; see Martínez-Monés, Dimitriadis, Rubia-Avi, Gómez-Sánchez, & Fuente-Redondo, 2003, for CSCL-specific examples).

Metacognitive CSCL tools have used SNA to both display and measure interaction during collaborative activities. In Gassner's (2004) approach, social networks of filtered e-mail data convey information about group roles, individual personality factors, and even evidence of cooperation. While this system is still a prototype, the design of meaningful filters for building social networks is an promising line of research that may add more content and meaning to social network analysis studies. Other systems, such as SAMSA (Martínez et al, 2003), and that by Nurmela, Lehtinen, & Palonen, (1999) were developed to filter a large amount of data off-line and point evaluators to key collaborative learning

events that need further study. The follow-up research is then carried out manually through content inspection or qualitative analysis.

Ogata, Matsuura and Yano (2000) extend the notion of the social network through a special metacognitive tool called a Knowledge Awareness Map that explicitly represents the content of the interaction and the objects students manipulate in the network. This tool can be seen as a specialized social network that also includes "knowledge pieces" describing information that is linked to participants. The Knowledge Awareness Map graphically shows users who else is discussing or manipulating their knowledge pieces. In this case, the distance between users and knowledge elements on the map indicates the degree to which users have similar knowledge.

In this section, we discussed the role of metacognitive tools that display high-level indicators in supporting students' collaboration and awareness, teachers' monitoring and assessment, and researchers' analysis and evaluation. Social Networks may benefit all of these communities by making the interaction in group structures visually explicit, and grounding the analysis in solid procedures based on mathematics and graph theory. SNA does not naturally represent the evolution of interaction, and may best be combined with other methods and techniques that can model how interaction and relationships change over time.

Systems that Internally Compare the Current State to a Model of Productive Interaction

The systems that we have discussed so far cover the first two stages of the collaboration management cycle (Figure 1), described at the beginning of this paper. We now turn to a discussion of systems in which the locus of processing (and the responsibility for analyzing the interaction) gradually shifts from the user to the system. These systems not only analyze, but also "diagnose" the student interaction in an attempt to deduce or infer where the students might be having trouble. This is generally done by internally comparing the current state of interaction to a model of ideal, or productive, interaction. The main challenges present during this process are (a) defining, as best possible, the model of desired interaction, and (b) designing algorithms that measure the degree to which the current model of interaction meets the requirements of the desired model, which may be uncertain or unstable. The result of the comparison in these systems is not displayed to the users, but instead used later by a coaching agent, or analyzed by researchers in an effort to understand and explain the interaction.

In the CSCL and AI-ED communities, we commonly think of "productive" interaction as interaction that facilitates learning. Models of productive interaction are therefore built from factors that are thought to positively influence learning. These factors are qualitative in nature, and involve analyzing the semantic aspects of interaction and the patterns of student actions. The systems in this section include internal representations that model aspects of collaborative learning such as coordination, conflict, and knowledge sharing.

Our first system was developed by Muehlenbrock and Hoppe (1999), two of the first researchers to propose sequences of multi-user actions in shared workspaces as a basis for qualitative analysis. The shared workspaces they consider for the automatic analysis provide semi-structured graphical representations for various types of domains and tasks. The users 'act' on these shared graphical representations by adding new textual and pictorial objects, and relations between these objects. The users can also remove or revise existing structures that result from previous joint problem solving, or that are drawn from sample material.

In contrast to student dialog, user actions have clearly defined operational semantics in terms of changes to the graphical structures. Hence they are directly available for automatic analysis, and do not require an intermediate labeling step, which might introduce a certain degree of error. Action-based collaboration analysis (Muehlenbrock, 2001) observes user actions in the temporal context of other users' actions as well as in the structural context of the graphical representations. It provides higher-level descriptions of the group activities (overview), and signals alerts when relevant events, such as task-related conflicts and coordination activities, have been detected (indicators). The analysis system was implemented as a plug-in component for the generic framework system CARDBOARD, which includes intelligent support components (CARDDALIS). A recent version (see Figure 4) has been used in a

psychological study for examining the influence of a feedback function to the behavior of the group (Zumbach, Muehlenbrock, Jansen, Reimann, & Hoppe, 2002).
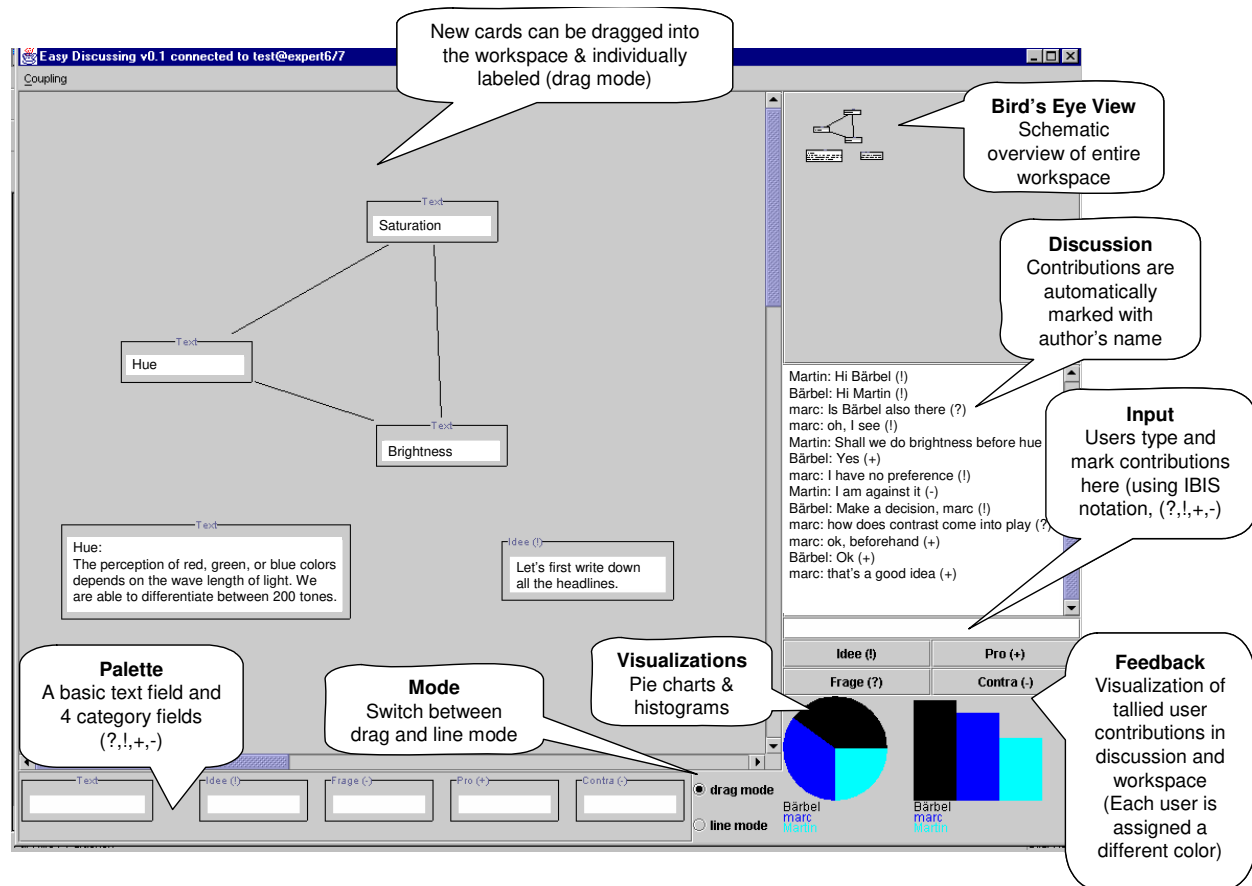

Figure 4. The CARDBOARD/CARDDALIS interface shows a large shared workspace for co-constructive activity using pre-defined cards (categories: text, idea, question, pro, contra), a chat interface (center right) with sentence makers (categories: idea, question, pro, contra), and a subtle feedback area (lower right) generated by the interaction analysis component.

The visualization of the model of interaction (e.g. the illustration of coordination vs. conflict in Figure 4) may be displayed to the students, as described in the previous section, or hidden from the students, and instead used by an instructor or computer-based facilitation agent in advising the students. It is worth noting, however, that if the model is sufficiently complex, containing a large number of interdependent variables with varying degrees of uncertainty, its construction will likely contain a margin of error. Such a model may be inappropriate to display to the user, and perhaps more meaningful to an automated facilitation agent that can abstract the relevant aspects of the model on which to base its advice. The models of interaction developed by the CARDBOARD/CARDDALIS system, and the EPSILON system which we describe next, are intended to be used by a coaching agent (in the future) in advising and guiding the group interaction.

Analyzing complex indicators, such as conflict and knowledge construction, may require sophisticated computation involving advanced modeling or natural language processing techniques. Interfaces that structure student conversation and activity in terms of a set of actions the system knows how to handle may facilitate the interpretation of student behavior. For example, our next system automatically analyzes structured sentence opener-based knowledge sharing conversation in the temporal context of workspace actions.

EPSILON (Soller, 2004; Soller & Lesgold, 2003) analyzes sequences of group members' communication and problem solving actions in order to identify situations in which students effectively share new knowledge with their peers while solving object-oriented design problems. In the first phase of the collaboration management cycle (Figure 1), the system logs data describing the students' conversation acts (e.g. Request Opinion, Suggest, Apologize) and actions (e.g. Student 3 created a new class). In the second phase, the system collects examples of effective and ineffective knowledge sharing, and constructs two Hidden Markov Models (HMMs) that describe the students' interaction in these two cases. A knowledge sharing example is considered effective if one or more students learn the newly shared knowledge (as shown by a difference in pre-post test performance), and ineffective otherwise. The Hidden Markov Modeling approach (Rabiner, 1989; also see Soller, 2004) is a probabilistic machine learning method that generates abstract generalizations of coded sequences of activity, in the form of nondeterministic state transitions.

At the beginning of the third phase of the collaboration management cycle, the EPSILON system has generated the HMMs describing effective and ineffective knowledge sharing, and is then prepared to dynamically assess a new group's interaction. It compares the sequences of student activity to the constructed Hidden Markov Models, and determines whether or not the students are experiencing a knowledge sharing breakdown. The system also includes multidimensional data clustering methods to help explain why the students might be having trouble, and what kind of facilitation might help.

Systems that Offer Advice

This section describes systems that analyze the state of collaboration using a model of interaction, and offer automated advice intended to increase the effectiveness of the learning process. The coach in an advising system plays a role similar to that of a teacher in a collaborative learning classroom. This actor (be it a computer coach or human) is responsible for guiding the students toward effective collaboration and learning. Since effective collaborative learning includes both learning to effectively collaborate, and collaborating effectively to learn, the facilitator must be able to address social or collaboration issues as well as task-oriented issues. Collaboration issues include the distribution of roles among students (e.g. critic, mediator, idea-generator) (Burton, 1998), equality of participation, and reaching a common understanding (Teasley & Roschelle, 1993), while task-oriented issues involve the understanding and application of key domain concepts. The systems described here are distinguished by the nature of the information in their models, and whether they provide advice on strictly collaboration issues or both social and task-oriented issues. We begin by taking a look at systems that advise the social aspects of collaborative learning.

Systems that Advise Social Aspects of Interaction

A classroom teacher might mediate social interaction by observing and analyzing the group's conversation, and noting, for example, the degree of conflict between group members' roles, or the quality of the conversation. A CSCL system that can advise the social aspects of interaction therefore could benefit from the ability to understand the dialog between group members. Barros and Verdejo's (2000) asynchronous newsgroup-style system, DEGREE, accomplishes this by requiring users to select the type of contribution (e.g. proposal, question, or comment) from a list each time they contribute to the discussion. This data satisfies the first phase of the collaboration management cycle. The system's model of interaction (phase 2 of the collaboration management cycle) is constructed using high-level attributes such as cooperation and creativity (derived from the contribution types mentioned above), as well as low-level attributes, such as the mean number of contributions. In the third phase of the collaboration management cycle, the system rates the collaboration between pairs of students along four dimensions: initiative, creativity, elaboration, and conformity. These attributes, along with others such as the length of contributions, factor into a fuzzy inference procedure that rates students' collaboration on a scale from "awful" to "very good". The advisor in DEGREE elaborates on the attribute values, and offers students tips on improving their interaction (see **Figure 5**).

MArCo (Tedesco, 2003) is a dialog-oriented system for the detection of meta-cognitive conflicts. The system adopts a dialog game approach with a limited set of possible dialog moves. User utterances must be formulated in a formal language that enables the conversation to be mapped onto a belief-based model (BDI). The analysis mechanism then detects disagreements and conflicts between users' beliefs and intentions. The mediator informs the group when it detects a conflict, and may also recommend alternative courses of action.
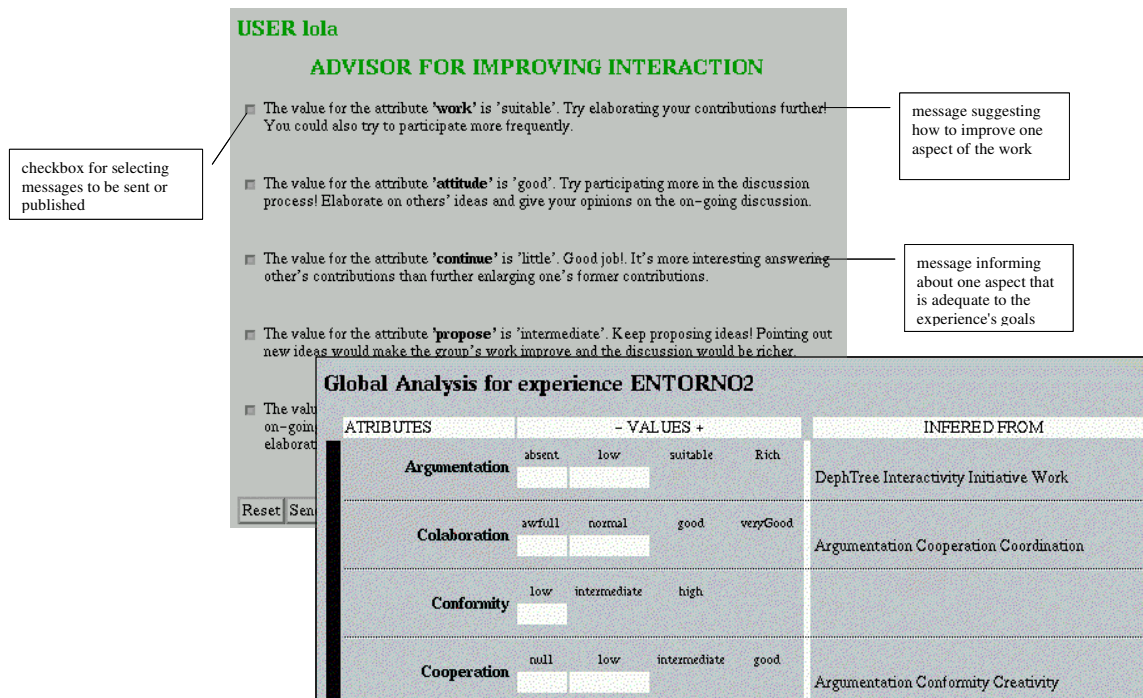

Figure 5. A snapshot of messages generated for a student in an intermediate stage of work, and below it, a general assessment of the group's behavior. From Barros, B., and Verdejo, M.F. (2000). Analysing student interaction processes in order to improve collaboration. The DEGREE approach. *International Journal of Artificial Intelligence in Education*, 11, 221-241. Copyright 2000 by the International AIED Society. Reprinted with permission.

The approaches taken by DEGREE and MArCo might be limited by their dependence on users' ability to choose the correct contribution type (proposal, comment, etc.). An alternative way of obtaining this information is to have users select *sentence openers*, such as "Do you know", or "I agree because" to begin their contributions. Associating sentence openers with conversational acts such as Request Information, Rephrase, or Agree, and requiring students to use a given set of phrases, allows a system to understand the basic flow of dialog without having to rely on Natural Language parsers. Most sentence opener approaches make use of a structured interface, comprised of organized sets of phrases. Students typically select a sentence opener from the interface to begin each contribution.

McManus and Aiken (1995) take this approach in their Group Leader system. Group Leader builds upon the concept that a conversation can be understood as a series of conversational acts (e.g. Request, Mediate) that correspond to users' intentions (Flores, Graves, Hartfield, & Winograd, 1988). Like Flores et al.'s Coordinator system, Group Leader uses state transition matrices to define what conversation acts should appropriately follow other acts; however unlike the Coordinator, users are not restricted to using certain acts based on the system's beliefs. Group Leader compares sequences of students' conversation acts to those recommended in four finite state machines developed specifically to monitor discussions about comments, requests, promises, and debates. The system analyzes the conversation act sequences,

and provides feedback on the students' trust, leadership, creative controversy, and communication skills, originally defined by Johnson, Johnson, and Holubec (1990).

The success of McManus and Aiken's Group Leader (1995) began a proliferation of systems that take a finite state machine approach to modeling and advising collaborative learners. One year later, Inaba and Okamoto (1996) introduced iDCLE, a system that provides advice to students learning to collaboratively prove geometry theorems. This system infers the state of interaction by comparing the sequences of conversation acts to four possible finite state machines. The finite state machines describe the mode of interaction; for example the model describing the query mode is used when the group is attempting to address a member's question, and the model describing the confirmation mode is used when the students are justifying or confirming an idea. Advice is generated through consideration of the dialog state and the roles of each group member. For example, iDCLE considers whether or not a group member is leading the discussion, or asking an abundance of questions, and tailors the advice appropriately.

OXEnTCHÊ (Vieira, Teixeira, Timóteo, Tedesco, & Barros, 2004) is an example of a sentence opener-based tool integrated with an automatic dialogue classifier that analyses on-line interaction and provides just-in-time feedback to both teachers and learners. During the first phase of the collaboration management cycle, the system collects the student chat logs, codified with sentence openers. During the second phase, the dialog classifier uses neural networks trained to identify productive and non-productive dialogs regarding a number of *collaborative skills,* (although the authors do not specify how the system computes these *collaborative skills*). The system also identifies off-task interactions by comparing the input with a domain ontology. During the third phase of the cycle, the system compares the students' interaction to its models and offers two types of feedback: teachers receive reports on both the group and individual students, and students may view the analysis of their contributions to the chat. The authors evaluated these reports in four experimental settings, obtaining positive results overall.

OXEnTCHÊ also includes a chatterbot (natural language agent) that acts as an advising system, attempting to maintain the dialogue focus. It interrupts the group chat when it detects an unproductive change of subject, and also attempts to motivate less participative students to engage in conversation. As of this writing, the chatterbot's functionality has not been formally evaluated, and further work is planned in order to improve its efficiency. Overall, OXEnTCHÊ goes further than many systems regarding data analysis and feedback, and also takes into account the needs of different types of users (teachers and students). The main limitations of the approach include the reliance on sentence openers and an unclear theoretical justification for the chosen collaboration skills.

Sentence opener approaches have gained popularity over the past 10 years because of their ease-of-use, and ability to efficiently reduce the amount of the amount of natural language understanding for which the system would otherwise be responsible. They are however not without their limitations. Because of the dialogical constraints of sentence openers, students may not always use them as expected. For example, it is possible to use the sentence opener, "I think", to say, "I think I disagree". The degree to which this will happen is largely dependent on the degree to which the set of sentence openers enable the students to express themselves, and the ability for them to find the phrases they need on the interface. Training students how to use the sentence openers available on the interface in contextualized situations, and running iterative human-computer interaction studies can both make a difference (Soller, 2004).

Because each sentence opener is also typically associated with only one intention (i.e. *Suggest* or *Justify*), a sentence opener-based coding scheme is only able to account for the primary intention. It cannot capture complex intentions, such as a *Discuss/Agree* act that both expresses agreement and doubt. We do not yet know to what degree a more complicated coding scheme might improve the ability to which the system can support various collaborative learning activities. Text mining methods that learn to automatically classify contributions into categories might help to address this problem, although approaches in this direction are also limited in their ability to understand natural human conversation (Linton, Goodman, Gaimari, Zarrella, & Ross, 2003; Padilha, Almeida, & Alves, 2004). Moreover, text mining techniques must be adapted to each learning domain, presenting an obstacle for their general use.

Dillenbourg (1999) describes the so-called "Berlin Wall" of collaborative learning as the notion of trying to understand and support the social aspects of collaboration separately from the task elements. Students cannot effectively learn how to collaborate outside the context of a concrete task, and cannot effectively learn how to perform the task collaboratively without attention to social and socio-cognitive factors influencing the group. A few systems have taken advantage of this idea by monitoring and analyzing students' task-based and social actions together in order to gain a better understanding of the collaboration as a whole.

Our last group of collaborative learning systems interacts with students via a set of specialized computer agents that address both social and task-oriented aspects of group learning. HabiPro (Vizcaino, 2001) is a collaborative programming environment that uses two databases – one containing words related to the domain (computer programming), and other containing potential off-topic terms. The system includes a simulated peer agent that detects off-topic words in the students' utterances, and intervenes as necessary to bring them back to work (see Figure 6).
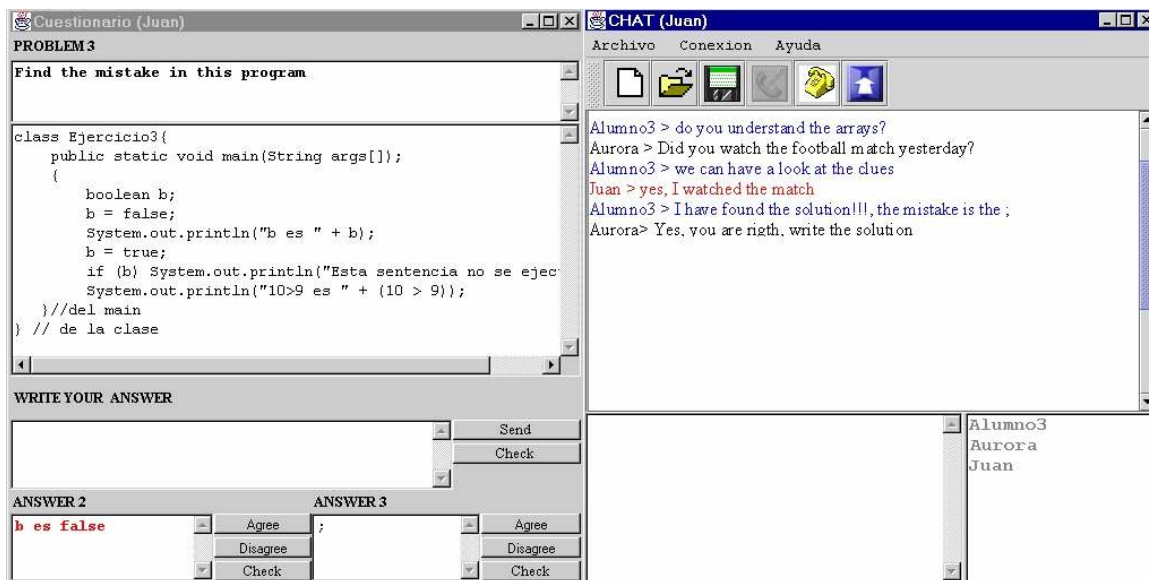

Figure 6. The simulated peer in HabiPro detects that the students are discussing the football match, and attempts to refocus them on the problem solving. Obtained by email from Aurora Vizcaino (avizcaino@inf-cr.uclm.es). Printed with permission.

HabiPro also includes a group model, and an interaction model, which includes a set of "patterns" describing possible characteristics of group interaction (e.g. the group prefers to look at the solution without seeing an explanation). During the collaborative activity, the simulated peer uses the group model to compare the current state of interaction to these patterns, and proposes actions such as withholding solutions until the students have tried the problem.

GRACILE (Ayala & Yano, 1998) is an agent-based system designed to help students learn Japanese. GRACILE's agents assess the progress of individual learners, propose new learning tasks based on the learning needs of the group, and cooperate to maximize the number of situations in which students may effectively learn from one another. In order to reach these goals, GRACILE maintains user models for each of the students, and forms beliefs about potential group learning opportunities. Group learning opportunities are defined as those that extend an individual's *zone of proximal development* (Vygotsky, 1978), which describes the potential development of a learner with the assistance of others. Following this idea, GRACILE proposes the students potential partners to collaorate with so that they maximize the opportunities to learn from each other.

The models of interaction employed by LeCS (Rosatelli & Self, 2002) and COLER (Constantino-González, Suthers, & Escamilla de los Santos, 2002) also integrate task and social aspects of interaction. LeCS is similar to GRACILE in that a set of computer agents guide students through the analysis of case studies. The agents monitor students' levels of participation, and track students' progression through the task procedure, while addressing students' misunderstandings and ensuring group coordination.

COLER (Constantino-González, Suthers, & Escamilla de los Santos, 2002) uses decision trees to coach students collaboratively learning Entity-Relationship (ER) modeling, a formalism for conceptual database design (see Figure 7). A *coach* monitors the personal and shared workspaces in order to detect opportunities for group-learning interactions. COLER draws on the socio-cognitive conflict theory (Doise & Mugny, 1984) which states that disagreements can be an opportunity for learning when students detect them and try to resolve them through reflection and elaboration.(Constantino-González et al., 2002) define three types of coaching opportunities: when there are problems in the quality of the ER group diagram, differences between individual and group ER diagrams, and differences in the levels of participation of the learners. COLER generates a set of potential advisory comments for a given situation using decision trees and chooses one of them according to a control strategy.
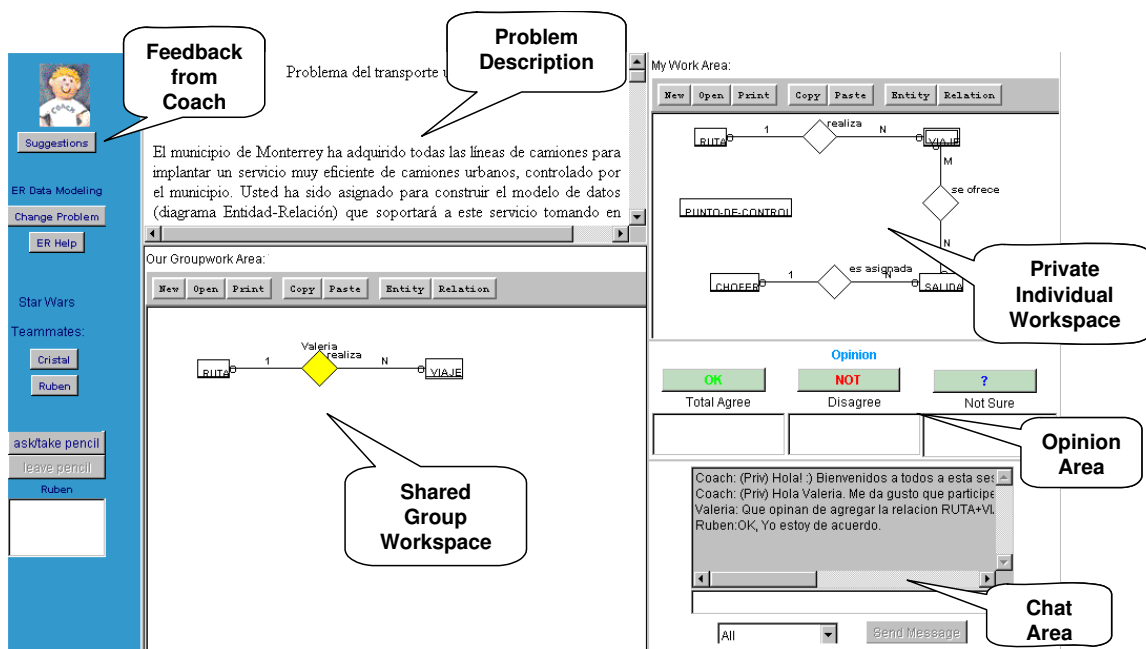

Figure 7. The COLER Workspace
[Available: http://lilt.ics.hawaii.edu/lilt/images/COLER.gif]

Goodman, Linton, Zarrella, and Gaimari (2004) present a specific example in which machine learning methods are used to train an agent-based system to recognize when students are experiencing trouble related to specific aspects of interaction. Their approach involves training neural networks with segmented, coded (speech act) student dialog and surface features (e.g. question marks and keywords). Goodman et al.'s research reminds us that choosing the appropriate technique for data analysis is critical, and depends on the conditions under which these techniques are, or are not, expected to be successful. In the next section, we summarize the various systems and methods that we have encountered in this article along the paths of the collaboration management cycle.

DISCUSSION

In the first half of this paper, we developed the collaboration management cycle from a system's

perspective. This cycle describes the actions a system can take to support online collaborative learning interaction. In the second half of this paper, we reviewed an array of systems that instantiate the stages in this model: mirroring, monitoring, and advising.

Mirroring systems record and reflect input data, while monitoring and advising systems process this input data to obtain a higher-level representation which is then either displayed to the collaborators (in the case of indicator-based systems), or used by the system or human facilitators (in the case of advising systems). This higher-level, *derived* representation may be quantitative or qualitative in nature. A quantitative derivation process might entail counting, for instance, the number of dialog or workspace actions a user has taken. A qualitative derivation process requires taking relational information into account, such as interdependencies between actions, or between actions and the application context. Tables 1-3 summarize the systems we have reviewed in this paper by the type of interaction data they assimilate, the derivation mechanism they use to produce higher-level (derived) data models, the format of the derived data model, and the way in which they attempt to achieve or maintain equilibrium (ideal collaboration).

In some cases, systems that monitor the state of interaction are not all that different from systems that provide advice. For example, suggesting that a student participate more does not require much more computation than displaying students' participation statistics; moreover both approaches may have the same effect. These systems begin to differ when the knowledge behind the indicators requires a great enough level of inferencing to warrant having a coach analyze the data to scaffold the learning process.

While reflecting on our review of systems to support collaborative learning, we noticed that there is a great diversity of approaches, and asked the question "Why?" Such diversity might be explained by the fact that each system draws upon a different theoretical perspective. But even systems that share the same view of learning employ different strategies for pedagogical intervention. For example, Table 1 shows that some systems focus on modeling features of the individual learners (learner models) in order to detect potential situations for productive interactions, while other systems that are based on similar theoretical principals, focus on analyzing collaborative interaction. GRACILE, a system that was inspired by Vygostky's zones of proximal development is an example of the first approach, whereas COLER, a system inspired by theories of socio-cognitive conflict, takes the second approach. Systems that characterize the second approach often attempt to understand how different patterns of interaction promote various aspects of collaborative learning such as knowledge sharing and construction (e.g. EPSILON, MarCo), or conversation (DEGREE). Some are more focused toward the social aspects of learning (e.g. HabiPro), while others study the structural properties of interactions within groups, such as the evolution of social roles (Salomon & Perkins, 98).

Table 1. A summary of **mirroring** systems that support collaborative learning

System	Input data	Output	Expected function
Groupkit, Gutwin (1995)	Shared workspace actions (window level)	Other users' interface actions	On-line workspace awareness
Plaisant et al. (1999)	Problem-solving actions	Actions on a timeline	Off-line analysis of the activity
Chat Circles (Donath, Karahalios & Viegas,1999)	Dialog in an unstructured virtual space	Graphical visualization	On-line social awareness
ART/SAILE (Goodman et. al, 2001)	Shared workspace actions (window-level)	Reproduction of the collaboration at different granularities	Off-line review of the collaborative process
SLA (Wasson, 2000)	Shared workspace actions (web-server level)	Users' connection times to the server	Off-line analysis of the possibilities of collaboration work

PENCACOLAS (Blasco et al., 2001)	Shared workspace actions (low-level events)	Video-like reproduction of the collaboration	Off-line review of the collaborative process for evaluation and self-evaluation purposes

Table 2. A summary of **metacognitive** systems that support collaborative learning

System	Input data	Derivation mechanism	Derived data	Output	Expected function
Sharlock II, Ogata et al. (2000)	User profile, web page access	Counting, similarity indices	Shared knowledge awareness map	Graphical visualization	Participants: Provide knowledge awareness
SAMSA, (Martínez-Monés et al., 2003)	Shared workspace actions	Social network analysis	Network density and centralisation; actors' centrality degree	Graphical visualization	Analyze participation structure
Jermann (2004)	Chat and problem-solving actions	Counting	Participation	Graphical visualization	Teachers: Analyze interaction Self-regulation for the students
Talavera and Gaudioso (2004)	Actions on a forum	Data mining and machine learning	Clusters of students with similar characteristics	Clusters of students	Identify student profiles and form groups
Nurmela (1999)	Actions on a shared workspace	Social network analysis	Actors´centrality degree	Indexes	Analyze participation structure
Simoff (1999)	Synchronous and asynchronous dialog (forum)	Counting and semi-automatic content analysis	Participation, structure of discussion	Graphical representation (Boxes)	Visualize the depth of the conversations
Action-based Collaboration Analysis, Muehlenbrock (2001)	Actions on graphical representation in shared workspaces	Activity/plan recognition	Action sequences, indicators for task-related conflicts and coordination	Graphical representation	Students' self-regulation, feedback to facilitator
EPSILON, Soller & Lesgold (2003)	Shared workspace actions, tagged dialog	Hidden Markov Models	Effectiveness of knowledge sharing	Textual assessment & explanation	Feedback to facilitator, Input to Computer-based Coach

Table 3. A summary of **guiding** systems that support collaborative learning

System	Input data	Derivation mechanism	Derived data	Output	Expected function
Group Leader, McManus and Aiken (1995)	tagged dialog	Finite state machines	Trust, leadership, communication	Coach	On-line textual feedback to the students on collaborative skills
iDCLE, Inaba and Okamoto (1996)	Tagged dialog	Finite state machines	Roles	Coach	On-line feedback to the students
DEGREE, Barros and Verdejo's (2000)	Tagged dialog	Counting and fuzzy inference	Initiative, creativity, elaboration, conformity	Coach and conversation analysis display	On-line feedback on "initiative, creativity, elaboration, & conformity"
MarCo (Tedesco, 2003)	Dialog in formal language	BDI modeling	Meta-cognitive conflicts	Conflict mediator	On-line feedback on alternatives when conflicts are detected
GRACILE, Ayala and Yano (1998)	Workspace actions, learner models	Rule-based expert system	Student helpers & learning tasks	Coaching agents	On-line proposals of group learning tasks
LeCS (Rosatelli & Self, 2002)	Shared workspace actions,	Case tree	Participation, group coordination	Coaching agents	On-line feedback of misunderstandings & coordination
COLER, (Constantino-González et al., 2002)	Shared and private actions, dialog	Decision trees	Participation, agreement with group procedure	Coach	On-line feedback of participation and workspace differences
OXEnTCHE, Vieira, Teixeira, Timóteo, Tedesco, & Barros. (2004)	Tagged dialog	Neural networks and comparison with a domain ontology	Productive and non-productive dialogs sequences	Chatterbot	On-line feedback of conversation productivity
HabiPro, Vizcaino (2001)	Shared workspace actions, student preferences, dialog	Matching group interaction "patterns", content analysis	Ideal participation, motivation, existence of off-topic conversations	Coach	Detection of off-topic interaction & on-line guidance to students

Because the systems described here are research prototypes, which tend to focus on a specific research question, they should be viewed from the perspective of that question. The collaboration management cycle, described at the beginning of this article, is intended to describe a way of understanding the capabilities available today for computationally supporting collaborative learning, rather than a way of classifying and comparing these systems. Developing a new system to support several different aspects of interaction might involve the application of research ideas from different systems, perhaps by way of re-implementation.

In this review article, we have attempted to provide an overview of the current technological capabilities, with the intention of laying the groundwork for further research that addresses the question of which technological solutions are appropriate for which learning situations. We now conclude by motivating this further research.

FUTURE WORK

The concept of *supporting* (as opposed to enabling) peer-to-peer interaction in computer-supported collaborative learning systems is still in its infancy (Jermann, Soller, & Lesgold, 2004). We have not yet

seen full-scale evaluations of the types of systems we have covered here. The evaluations that were conducted for many of these systems, if at all, were done so under closely controlled laboratory conditions. Laboratory studies are critical for developing an understanding of the various conditions that affect learning, and make sense as the first step in the assessment and redesign of the technology. If our objective is to assist students and teachers during real, curriculum-based learning activities, we must also understand how well our laboratory findings apply to natural classroom situations. This can only be done by developing and deploying robust technology in physical and virtual classrooms, and performing large scale evaluations. The feedback obtained from such evaluations should enhance the evaluation feedback loop in the collaboration management cycle, and further our understanding of which technological solutions help students, and which do not.

More studies are needed that test the utility of various strategies for computationally supporting online collaborative learning. It is probable that certain strategies are more beneficial than other strategies under various conditions, and for different domains. There is hence an important opportunity for needs analysis studies to understand which types of systems (i.e. mirroring, monitoring, or advising) are useful under various constraints (i.e. group size and ability, environment, task characteristics, availability of human instructors). Then, further analyses of computer-mediated interaction in parallel with a finer-grained needs analysis may help to determine which behavioral and pedagogical factors are influenced in what ways by the various technological features. Only then may we be in a position to recommend specific technologies for fostering established learning activities.

In some cases, a combination of technologies may be most practical. For example, analyzing visual indicators may increase students' cognitive load; moreover, some students may misinterpret the indicators. But, the interaction management skills students learn as they attempt to interpret and act upon these indicator values might transfer well to other situations. One possibility is to both display indicator values to students, and provide advice based on a deeper computational analysis of the data that was used to generate the indicators.

Many of the approaches presented in this article address effects *with* technology, rather than effects *of* technology (Kolodner & Guzdial, 1996; Salomon, Perkins & Globerson, 1991). Effects *with* technology refer to the changes in the group dynamics that are triggered by software tools, whereas effects *of* technology refer to the outcome of the collaboration, both for the individual and the collective group. These outcomes include the skills that students acquire or improve, and whether or not these skills might transfer to a new learning situation or group experience. More research is needed to determine how visual feedback through mirroring and metacognitive tools, or advice from guiding systems can lead to learning gains. In designing support for the collaborative learning process, we must still not forget to assess the product.

The techniques and systems described throughout this article use different standards for diagnosis. How might we develop modular, reusable solutions that would allow researchers to share and reuse tools in different CSCL environments? Instead of proposing new data formats and interfaces, would it be reasonable to tackle this problem in parallel with current efforts toward introducing collaboration aspects in e-learning standards? In the future, we could aim to develop reusable models of collaborative processes, based on modular architectures, that can provide the computational, theoretical, and pedagogical foundations for guiding tools, while encouraging metacognitive reflection by both teachers and students. Such models might even be used in teacher training, to help explain breakdowns in student interaction, or the dynamics of productive collaborative learning interaction.

Knowledge about how students interact in a computer-mediated environment is useful to a system only if it knows when and how to apply this knowledge to recognize specific situations that call for intervention. Classroom teachers learn to analyze and assess student interaction through close observance of group interaction, trial and error, and experience. Developing a system to analyze group conversation, however, poses its own challenges. For example, how do we go about calibrating a set of indicators that should represent a model of desired interaction, and what learning theories or experimental results allow for this calibration? This leads us to the broader issue of how to quantify and translate well-known theories from the learning and cognitive sciences into computational models that can be used to diagnose student interaction. For example, how might the principle relating elaborated explanations to learning gains (Webb, 1992) be quantified as a set of calibrated indicators that can be computed on the fly during

computer-mediated interaction? A "sufficiently elaborated explanation" might be relatively long, and refer to several domain concepts, making computer diagnosis difficult. The theoretical and experimental foundations for our models must be strengthened, justified, and assessed. Focused research in computational modeling of peer interaction in context may help in making the transition from understanding how to mediate learning groups to understanding how to train a system assist in mediating learning groups more effectively.

ACKNOWLEDGMENTS

REFERENCES

Avouris, N., Dimitracopoulou, A., & Komis, V. (2003). On analysis of collaborative problem solving: An object oriented approach. *Computers in Human Behavior*, *19*(2), 147-167.

Avouris, N., Komis, V., Margaritis, M., & Fiotakis, G. (2004). An environment for studying collaborative learning activities. *Educational Technology & Society*, *7*(2), 34-41.

Ayala, G, & Yano, Y (1998). A collaborative learning environment based on intelligent agents. *Expert Systems with Applications*, 14, 129-137.

Barros, B., & Verdejo, M.F. (2000). Analysing student interaction processes in order to improve collaboration. The DEGREE approach. *International Journal of Artificial Intelligence in Education*, 11, 221-241.

Blasco, M., Barrio, J., Dimitriadis, Y., Osuna, C., González, O., Verdú, M., & Terán, D (1999). From co-operative learning to the virtual class. An experience in composition techniques. *ultiBASE journal.* Available online: http://ultibase.rmit.edu.au/Articles/dec99/blasco1.htm.

Burton, M. (1998). Computer modelling of dialogue roles in collaborative learning activities. Unpublished doctoral dissertation, Computer Based Learning Unit, The University of Leeds.

Constantino-González, M. A., Suthers, D. D., & Escamilla de los Santos, J. G. (2002). Coaching web-based collaborative learning based on problem solution differences and participation. *International Journal of Artificial Intelligence in Education, 13*, 263.299.

Dillenbourg, P. (1999). Introduction; What do you mean by Collaborative Learning? In Dillenbourg, P., editor*, Collaborative Learning. Cognitive and Computational Approaches*, Elsevier Science Ltd., Oxford, U.K. 1-19.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL* (pp. 61-91). Heerlen, Open Universiteit Nederland.

Dillenbourg, P., Baker, M., Blaye, A., & O'Malley, C. (1995). The evolution of research on collaborative learning. In H. Spada and P. Reinmann (Eds.), *Learning in Humans and Machines*, Elsevier Science.

Dimitracopoulou, A., & Komis, V. (2004). Design principles for an open environment, supporting learning process participants on modelling and collaboration. In C. Constantinou, Z. Zacharias & P. Kommers (Guest Editors), *Int. Journal of Continuing Engineering Education and Lifelong Learning, Special Issue on the Role of Information and Communication Technologies in Science Teaching and Learning.*

Doise, W., & Mugny, G. (1984). *The social development of the intellect*. Oxford: Pergamon Press.

Donath, J., Karahalios, K., & Viégas, F. (1999). Visualizing conversation. *Journal of Computer-Mediated Communication*, 4(4).

Fesakis, G., Petrou, A., & Dimitracopoulou, A. (2004). Collaboration Activity Function: An interaction analysis tool for Computer Supported Collaborative Learning activities. *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT 2004)*, 196-200.

Flores, F., Graves, M., Hartfield, B., & Winograd, T. (1988). Computer systems and the design of organizational interaction. *ACM Transactions on Office Information Systems,* 6(2), 153-172.

Gassner, K., Jansen, M., Harrer, A., Herrmann, K., & Hoppe, U. (2003). Analysis methods for collaborative models and activities. In B. Wasson, S. Ludvigsen, & U. Hoppe (Eds.) *Designing for Change in Networked Learning Environments: Proceedings of the International Conference on Computer Support for Collaborative Learning 2003*. (pp. 369-377). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Gassner, K. (2004). Using Patterns to Reveal E-Mail Communication Structures. J. Mostow, & P. Tedesco *Designing Computational Models of Collaborative Learning Interaction, workshop at. ITS 2004* (pp. 77-82). Maceió, Brazil.

Gerosa, M. A., Gomes Pimentel, M., Fuks, H., & Lucena, C. (2004). Analyzing Discourse Structure to Coordinate Educational Forums*. Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004.* (pp. 262-272). Berlin Heilderberg: Springer.

Goodman, B., Geier, M., Haverty, L., Linton, F., & McCready, R. (2001). A framework for asynchronous collaborative learning and problem solving. *Proceedings of AIED'01, 10th International Conference on Artificial Intelligence in Education*, San Antonio, Texas, 188-199.

Gutwin, C., Stark, G., & Greenberg, S. (1995) Support for Workspace Awareness in Educational Groupware. *Proceedings of CSCL'95. The First International Conference on Computer Support for Collaborative Learning*, 147-156.

Hutchins, E. (1995). How a Cockpit Remembers Its Speeds. *Cognitive Science, 19*, 265-288.

Inaba, A., & Okamoto, T. (1996). Development of the intelligent discussion support system for collaborative learning.. *Proceedings of ED-TELECOM '96,* Boston, MA, 137-142.

Jermann, P. (2004). *Computer Support for Interaction Regulation in Collaborative Problem-Solving*. Doctoral Dissertation, University of Geneva.

Jermann, P., Soller, A., & Lesgold, A. (2004). Computer software support for CSCL. In P. Dillenbourg (Series Ed.) & J. W. Strijbos, P. A. Kirschner & R. L. Martens (Vol. Eds.), *Computer-supported collaborative learning: Vol 3. What we know about CSCL ... and implementing it in higher education* (pp. 141-166). Boston, MA: Kluwer Academic Publishers.

Jermann, P., Soller, A., & Muehlenbrock, M. (2001). From mirroring to guiding: A review of state of the art technology for supporting collaborative learning. *Proceedings of the First European Conference on Computer-Supported Collaborative Learning*, Maastricht, The Netherlands, 324-331.

Johnson, D., Johnson, R., & Holubec, E. J. (1990). *Circles of learning: Cooperation in the classroom* (3rd ed.). Edina, MN: Interaction Book Company.

Kolodner, J., & Guzdial, M. (1996). Effects with and of CSCL: Tracking learning in a new paradigm. In T. Koschmann (Ed.) *CSCL: Theory and Practice of an Emerging Paradigm* (pp. 307-320). Mahwah NJ: Lawrence Erlbaum Associates.

Linton, F., Goodman, B., Gaimari, R., Zarrella, J., & Ross, H. (2003). Student modeling for an intelligent agent in a collaborative learning environment. *Proceedings of the 9th International Conference on User Modeling (UM '03),* Johnstown, PA.

Looi, C.-K. (2001). Supporting conversations and learning in online chat. *Proceedings of AIED'01, 10th International Conference on Artificial Intelligence in Education*, San Antonio, Texas, 142-153.

Martínez-Monés, A., Dimitriadis, Y., Rubia-Avi, B., Gómez-Sánchez, E., & Fuente-Redondo, P. (2003). Combining qualitative evaluation and social network analysis for the study of classroom social interactions. *Computers and Education, 41*(4), 353-368.

Martínez-Monés, A., Guerrero, L., & Collazos, C. (2004). A model and a pattern for the collection of collaborative action in CSCL systems. In J. Mostow, & P. Tedesco (Eds.) *Designing Computational Models of Collaborative Learning Interaction, workshop at. ITS 2004* (pp. 31-36). Maceió, Brazil.

McManus, M. & Aiken, R. (1995). Monitoring computer-based problem solving. *Journal of Artificial Intelligence in Education*, 6(4), 307-336.

Muehlenbrock, M. (2001). *Action-based Collaboration Analysis for Group Learning*. Amsterdam: IOS Press.

Muehlenbrock, M., & Hoppe, U. (1999). Computer supported interaction analysis of group problem solving. In C. Hoadley & J. Roschelle (editors), *Proceedings of the Conference on Computer Supported Collaborative Learning CSCL-99* (pages 398-405). Palo Alto, CA, December. Mahwah, NJ: Erbaum.

Neale, D. & Carroll, J. (1999). Multi-faceted evaluation of complex, distributed activities. *Proceedings of the Computer Support for Collaborative Learning (CSCL) 1999 Conference*. Palo Alto, CA: Stanford University, 425-433.

Nurmela, K.A., Lehtinen, E., & Palonen, T. (1999). Evaluating CSCL log files by Social Network Analysis. In C. M. Hoadley and J. Roschelle (Eds.), *Proceedings of the Computer Support for Collaborative Learning (CSCL) 1999 Conference*. Palo Alto, CA: Stanford University, 434-444.

Ogata, H., Matsuura, K., & Yano, Y. (2000). Active Knowledge Awareness Map: Visualizing learners activities in a Web based CSCL environment. *International Workshop on New Technologies in Collaborative Learning*, Tokushima, Japan.

Padilha, T. P. P., Almeida, L. M., & Alves, J. B. M. (2004). Mining Techniques for Models of Collaborative Learning. In J. Mostow, & P. Tedesco (Eds.) *Designing Computational Models of Collaborative Learning Interaction, workshop at. ITS 2004* (pp. 89-94). Maceió, Brazil.

Plaisant, C., Rose, A., Rubloff, G. Salter, R., & Shneiderman, B. (1999). The design of history mechanisms and their use in collaborative educational simulations. *Proceedings of the Computer Support for Collaborative Learning (CSCL) 1999 Conference*., Palo Alto, CA: Stanford University, 348-359.

Rabiner, L. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257-286.

Robertson, R.J. & Powers, W.T. (1990). *Introduction to modern psychology: the control-theory view*. Gravel Switch, KY: Control Systems Group.

Rosatelli, M., & Self, J. (2002). A Collaborative Case Study System for Distance Learning. *International Journal of Artificial Intelligence in Education, 12*, 1-25.

Roseman, M., & Greenberg, S. (1992). GroupKit: A groupware toolkit for building real-time conferencing applications. *Proceedings of the ACM CSCW Conference on Computer Supported Cooperative Work*, Toronto, Canada, 43-50.

Salomon, G. (1993) (Ed.). *Distributed cognitions. Psychological and educational considerations*. Cambridge: University Press.

Salomon G., & Perkins, D.N. (1998). Individual and social aspects of learning. *Review of Research in Education, 23*, 1-24.

Salomon, G., Perkins, D., & Globerson, T. (1991). Partners in cognition: Extending human intelligence with intelligent technologies. *Educational Researcher*, *20*(4), 2-9.

Simoff, S. (1999). Monitoring and Evaluation in Collaborative Learning Environments. *Proceedings of the Computer Support for Collaborative Learning (CSCL) 1999 Conference*, Palo Alto, CA: Stanford University.

Soller, A. (2004). Computational Modeling and Analysis of Knowledge Sharing in Collaborative Distance Learning. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, 14 (4), 351-381.

Soller, A., Jermann, P., Muehlenbrock, M., & Martinez-Monez, A. (2004). *Proceedings of the 2nd International Workshop on Designing Computational Models of Collaborative Learning Interaction*, ITS 2004, Maceio, Brazil.

Soller, A., & Lesgold, A. (2003). A Computational Approach to Analyzing Online Knowledge Sharing Interaction. *Proceedings of Artificial Intelligence in Education 2003,* Sydney, Australia*, 253-260.

Talavera, L., & Gaudioso, E. (2004). Mining Student Data to Characterize Similar Behaviour Groups in Unstructured Collaboration Spaces. *Proceedings of the Artificial Intelligence in Computer Supported Collaborative Learning Workshop at the ECAI 2004* . Valencia, Spain.

Teasley, S., & Roschelle, J. (1993). Constructing a joint problem space. In S. Lajoie and S. Derry (Eds.), *Computers as cognitive tools* (pp. 229-257). Hillsdale, NJ: Lawrence Erlbaum.

Tedesco, P. (2003). MArCo: Building an Artificial Conflict Mediator to Support Group Planning Interactions. *International Journal of Artificial Intelligence in Education, 13*, 117-155.

Vieira, A. C., Teixeira, L., Timóteo, A., Tedesco, P., Barros, F. A. (2004). Analyzing On-Line Collaborative Dialogues: The OXEnTCHÊ-Chat. In J. C. Lester, R. M. Vicari, F. Paraguaçu (Eds.): *The 7th International Conference on Intelligent Tutoring Systems, ITS 2004*, Maceiò, Alagoas, Brazil, 315-324.

Vizcaíno, A. (2001). Negative situations in collaborative environments: Can a simulated student avoid them? *Proceedings of AIED'01, 10th International Conference on Artificial Intelligence in Education*, San Antonio, Texas, 610-612.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. London: Harvard University Press.

Wasserman, S., & Faust, K. (1996). *Social Network Analysis: Methods and Applications*. Cambridge: Cambridge University Press.

Wasson, B., Guribye, F., & Mørch, A. (2000). *Project DoCTA: Design and use of collaborative telelearning artefacts.* Technical report, Pedagogisk Informasjonsvitenskap, Universitetet i Bergen, Bergen, Norway.

Webb, N. (1992). Testing a theoretical model of student interaction and learning in small groups. In R. Hertz-Lazarowitz and N. Miller (Eds.), *Interaction in Cooperative Groups: The Theoretical Anatomy of Group Learning* (pp. 102-119). New York: Cambridge University Press.

Zumbach, J., Mühlenbrock, M., Jansen, M., Reimann, P., & Hoppe, U. (2002). Multi-dimensional tracking in virtual learning teams: An exploratory study. *Proceedings of the Conference on Computer Supported Collaborative Learning CSCL-2002*, Boulder, CO, 650-651.