Designing Computational Models of Collaborative Learning Interaction: Introduction to the Workshop Proceedings

Amy Soller\textsuperscript{1}, Patrick Jermann\textsuperscript{2}, Martin Mühlenbrock\textsuperscript{3}, Alejandra Martínez\textsuperscript{4}

\textsuperscript{1}ITC-IRST, via Sommarive 18, 38050 Povo, Trento, Italy
\textsuperscript{2}Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland
\textsuperscript{3}German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
\textsuperscript{4}Dept. of Computer Science, Universidad de Valladolid, Valladolid, Spain
Email: soller@itc.it, Patrick.Jermann@epfl.ch, Martin.Muehlenbrock@dfki.de, amartine@infor.uva.es

Abstract: Computational models of collaborative learning interaction provide functional computer-based representations that help us understand, explain, and predict patterns of group behavior, and support group learning processes. These models may assist students and teachers in managing and guiding the interaction during learning activities. In this paper, we introduce the proceedings of the 2\textsuperscript{nd} International Workshop on Designing Support for Collaborative Learning Interaction by exploring the advantages, implications, and support possibilities afforded by various types of computational models, in the context of a conceptual framework that we have developed.

Keywords: regulate, mediate, mirroring, metacognitive, guiding, collaboration management

Introduction

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. The papers included in these workshop proceedings explore the advantages, implications, and support possibilities afforded by the various types of computational models of collaborative learning processes.

Computational models of collaborative learning interaction provide functional computer-based representations that help us understand, explain, and predict patterns of group behavior, and support group learning processes. These models 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, 2002). 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.

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.

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. In this paper, we describe a few of these tools, in the context of a conceptual framework that we have developed to organize and describe the array of available collaborative support options. Our objective in presenting this conceptual framework is to help structure discussions during this workshop, and provide a focus for the papers in these proceedings. We first describe a general framework that we have found useful for understanding the process of computer-supported collaboration management, highlight a few of the workshop’s themes in the context of this framework, and present open issues for discussion and future research.

Designing support for collaboration management

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 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: 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.
- Phase 2: 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: 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. 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).

- Phase 4: 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.

**Understanding the locus of processing and computer-based support options**

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 when the computer takes over various phases of the collaboration management process presented in the previous section.

**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** 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** 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 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 throughout these proceedings), 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.
Workshop highlights from a collaboration management perspective

In this section, we highlight a few of the themes in these workshop proceedings with regard to the collaboration management cycle.

Phase 1

The first phase in each cycle is the data collection phase, in which the student interaction is recorded to a logfile, database, or internal data cache. While this step might seem simple, it still poses significant challenges. On one hand, the establishment of a standard data format might enable researchers to share and reuse analysis tools across different CSCL systems. On the other, such standards may limit the level of customization we can provide to users, who may want to choose specific combinations of data to analyze. Regarding the first issue, we do not know of a widely accepted data format that could facilitate the reuse and sharing of analysis methods developed by different researchers; however, some of the papers in these proceedings present contributions in this direction.

For example, Avouris, Margaritis, and Komis propose a system, Synergo, that builds on the Object-oriented Collaboration Analysis Framework (OCAF). Their system is unique in that student interaction and workspace actions are analyzed from the shared objects’ point-of-view. The objects that students manipulate independently compile statistics on their use, and contribute to the definition of indicators describing their owners’ collaborative behavior. OCAF includes a formal definition of events as tuples that include time, actions, objects, and configurable types of events. These elements are also considered in the collaborative action model, proposed by Martínez, Guerrero, and Collazos, that includes aspects related to the context of the interaction (Martinez, Dimitriadis, & de la Fuente, 2003). Drawing on this model of collaborative action, the authors address the problem of customizing the collection of data through the use of the command design pattern for the implementation of the data collection, modularizing it and enabling the desired customization. This design pattern is a general solution that can be used to define logging functionality in any type of application (not only those that are designed to mediate collaborative activity). Castro and colleagues present a similar architecture based on the Model-View-Controller design pattern, designed to support collaborative model construction.

Phase 2

The second phase involves computing one or more higher-level variables, termed indicators, to represent the current state of interaction. Information about the interaction takes many forms, from low-level user interface events (e.g. mouse clicks or movements, drag & drop actions, keystrokes) to actions that carry meaning in terms of the task (e.g. message posts, utterance type or category selections, graph node or edge deletions). The aggregation process carried out in this phase may lead to cognitive interpretations of certain data combinations, enabling researchers to assign value to these aggregated actions in terms of learning or problem solving (e.g. propose counter-argument, refine simulation setting, complement schema, explain strategy).

Some of the systems presented at the workshop implement indicators that aim at helping their users (teachers or students) understand the state of the collaboration specifically in terms of their indicators. For example, Avouris, Margaritis, and Komis propose the collaboration factor (CF), as an aggregation of other, quantitative, lower-level indicators, and Padihla, Almeida, and Alves propose performance reports based on a set of quantitative and qualitative indicators. Both of these indicators are graphically displayed on the time axis, facilitating the analysis of collaboration over a set time period. Vassileva, Cheng, Sun, and Han’s system aggregates the different types of contributions users have made to a virtual community over time, and shows users visualizations of their membership level in the community. The possibility of achieving a silver or gold membership status encourages users to contribute to their community.

Phase 3

During the third phase of the collaboration management cycle, the interaction is “diagnosed” by comparing the current state of interaction to a desired model of interaction. The main challenges present during this step 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.
In preparation for this phase, the derivation of indicators from raw data during the second phase might benefit from the action or dialog classification techniques described by Tedesco and Rosatelli. For instance, plan recognition techniques might be used to identify and distinguish different problem solving strategies, Markov modeling might be used to recognize or predict interaction patterns (e.g. Soller & Lesgold, 2003), pattern recognition techniques might allow sequences of events to be grouped into more general behavioral units (as in Gassner’s contribution), Bayesian modeling might be useful for describing the relationships between actions and their causal effects (as in Muehlenbrock’s contribution), and filtering techniques might help in determining which actions are meaningful, and which should be disregarded as “noise” (e.g. rearranging the nodes in a graph for aesthetic reasons).

Although these learning and classification techniques help in understanding, modeling, and assessing student activity, the problem of analyzing unstructured student dialogue is still an open issue. Padihla, Almeida, and Alves discuss how various text and data mining techniques might address this issue, and Goodman, Linton, Zarrella, and Gaimari present a specific example in which machine learning methods are used to train a 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).

Costa and Dimuro propose an alternative approach to diagnosing the state of interaction, based on the notion of social exchange values, as defined by Piaget and Homans. Their contribution explains the concept of exchange values, and how they can be used to help assess the quality of the interaction. Underlying this approach is the idea that effective social exchange should tend towards equilibrium. Suthers also presents an interesting proposal, building on Muehlenbrock and Hoppe’s (1999) analysis of actions in shared workspaces, in which collaborative learning interaction patterns might be assessed by modeling mode shifts between domain-oriented verbal and task actions.

Phase 4

If there are discrepancies between the current state of interaction (as described by the indicator values) and the desired state of interaction, the system may enter the fourth phase, in which it alerts a human facilitator as to the nature of the discovered discrepancies, or directly takes remedial actions in the collaborative virtual space. For example, the system described by Borges and Baranauskas alerts a facilitator when it detects critical periods in the student interaction, recommending intervention.

Tedesco and Rosatelli review several different systems in which a computer-based coach provides guidance to the learning group. For example, the COLER system (Constantino-Gonzalez, Suthers, & Escamilla de los Santos, 2002) detects differences between the students’ personal and shared workspaces, and differences between students’ participation levels in order to identify opportunities for facilitating group learning interactions. Tedesco’s MarCo system models and monitors the group dialog, intervening with recommendations when it detects meta-cognitive conflicts.

In some cases, metacognitive tools 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.

Open Issues

In these workshop proceedings, we see how Artificial Intelligence techniques such as pattern and plan recognition, and data mining may be valuable in the construction of indicators from raw interaction data, and how guiding systems might diagnose interaction, proposing recommendations to learners or teachers. The theoretical and experimental foundations for our models, however, must be strengthened, justified, and assessed. What does it mean when we calibrate a set of indicators to constitute 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?
interaction? A “sufficiently elaborated explanation” might be relatively long, and refer to several domain concepts, making computer diagnosis difficult.

The techniques and systems described throughout these proceedings 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 hope 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.

Many of the approaches presented in these proceedings 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 not forget to assess the product.

References


