ClassroomImaging: a tool to code and visualize compley classroom learning processes

Ingo Kollar, A. Harrer

ClassroomImaging:
A Tool to Code and Visualize
Complex Classroom Learning Processes

Ingo Kollar, Ludwig-Maximilians-Universität München, ingo.kollar@psy.lmu.de
Andreas Harrer, Clausthal University of Technology, andreas.harrer@tu-clausthal.de

Abstract: Data from classrooms is typically complex as it includes the presence of multiple actors (e.g., 25 students and a teacher), possible activities (e.g., explaining, questioning, monitoring), social levels on which learning takes place (e.g., individual learners vs. small groups), and artifacts that are used by the participants. Further, all these aspects may change over time. Yet, few tools are available that are capable of both coding such data in an integrative manner and of creating intuitive, dynamic visualizations of the learning process. This paper describes ClassroomImaging, a tool to code and visualize complex classroom learning processes by resorting to (a) a spatial representation of the classroom, (b) a symbolic representation of the participants (teacher, students), artifacts, (c) teaching and learning activities, (d) the social level on which learning is realized and (e) the temporal dynamics of the above properties during the learning situation.

Keywords: classroom learning, coding, data analysis, data visualization, learning processes

Introduction

Over the past years, a lot of CSCL research has dealt with how to effectively orchestrate complex classroom learning (see Dillenbourg, 2013). In that research, emphasis is given to the constantly changing conditions and processes of classroom learning. For example, the teacher may switch between different teaching and learning activities, which may be situated at the plenary, the small-group or the individual level (Dillenbourg & Jermann, 2007). Further, learners may interact with artifacts (e.g., chairs, desks, laptop computers) and with other learners as they are learning.

Empirical research is interested in the effects of such orchestrations on the learning process of the class. E.g., research might focus on the effects different orchestrations have on the frequency or quality of explanations (van Boxtel, van der Linden & Kanselaar, 2000), questions (King, 2007) or arguments (Weinberger, Stegmann & Fischer, 2010) that are produced by the learners. The effects of one orchestration may then be compared to the effects of an alternative orchestration (see Kollar, Wecker, Langer & Fischer, 2013). To perform such analyses, researchers typically do two things: (a) *Coding data*: They develop a coding scheme that captures a certain process dimension (e.g., argumentation quality) of the data of each individual actor (which may exist in the form of videos, audio recordings, chat logs, etc.), and then code the data accordingly. (b) *Visualizing results*: To demonstrate the effect of a certain orchestration on a certain process variable, they compute and visualize the frequency of the occurrence of certain learning process variables (e.g., the frequency of warranted arguments), e.g., in bar or line diagrams.

Yet, researchers face several challenges during this process. With respect to *coding*, the data sources are often spread across several files with different formats (such as audio recordings of talk and logfile data). That way, it is complicated to quickly switch between the multiple data sources available, which makes coding of the learning processes of all possible actors inconvenient. With respect to *visualizing* the effects of different orchestrations, a reliance on bar or line diagrams that aggregate frequencies of particular process categories obscures what particular aspects of the learning environment actually evoked those learning processes and whether single individuals were differentially affected by those aspects (e.g., a certain teacher activity). This is even more problematic when acknowledging the dynamic nature of complex classroom scenarios (Mercer, 2008), in which a given learner may find herself in a face-to-face discussion within a small group during one phase of learning, and learning individually with a laptop in the next.

Interaction analysis has been a topic in CSCL research for several years, and has resulted in many helpful tools such as interaction awareness tools (Gassner et al., 2003) and video coding tools (Avouris et al., 2007). A lot of these approaches support analysis at an abstract and general level, such as the contingency graph analysis (Suthers, 2009) that associates users, messages and their semantic relations or trace analysis tools (Settouti et al., 2009, or the Tatiana toolsuite). In contrast, our approach stresses the physical aspect of learning within a classroom and its orchestrations, information that is not easily detectable automatically if not using wearable technology (Hernández-Leo et al., 2013). Thus, we see interesting synergies with the mentioned approaches in the future, yet have to create our own dedicated approach and tool that captures classroom interaction data before integrating it with other tools.

This paper therefore presents ClassroomImaging, a software tool that is supposed to afford (a) the coding of classroom data stored in different files and formats in a convenient and integrated way, and (b) the visualization of the flow of activities within complex scenarios in a dynamic, non-aggregated fashion, thereby providing an intuitive understanding of the learning processes that go on in complex classroom scenarios.

Introducing ClassroomImaging: Code and visualize complex classroom data

The basic goal that guided the development of ClassroomImaging was to create an integrated tool that helps researchers to (a) conveniently code learning processes that occur in complex classroom scenarios and to (b) create dynamic and intuitive visualizations of those processes. To do so, we argue that the representation used for coding as well as for the visualization ideally should mirror the actual classroom situation.

Preparing for coding within ClassroomImaging

In a first step, ClassroomImaging asks the user to re-build the classroom from which the data is coming (and for which appropriate recordings must exist) in a symbolic, albeit sufficiently realistic way. For that purpose, ClassroomImaging helps the user create a bird's-eye-view-representation of the classroom. This includes the possibility to choose from a set of objects that are commonly used in classrooms (such as desks, whiteboards, laptops) and to freely drag and drop them to the appropriate positions. Furthermore, users can represent every single actor (i.e., each student plus the teacher) and position them in the representation (using circles as representations for students, triangles for teachers) according to their position in the video recording (or photo/drawing, if no video exists). To make every actor identifiable, each actor can be given a code (e.g., a number) that is "sticked" to him or her (see fig. 1 for an example of the initial setup of a classroom situation).

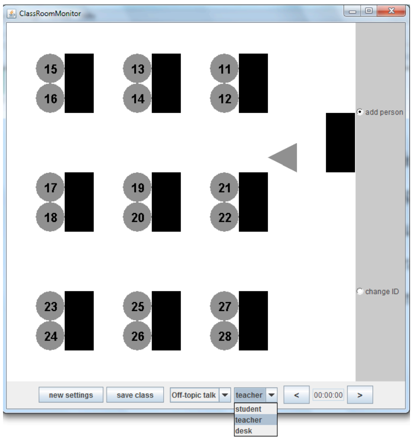

Figure 1. Editing the classroom layout and initial position of students and teacher.

For future versions of our tool we plan that after the initial set-up of the classroom is completed, available data sources for each actor are associated with the physical representation of that actor. I.e., if for one student video plus logfile data is available, a double-click on the symbol that represents that actor will bring up those files so that they can immediately be coded. If multiple files exist, the user can choose which one to use for coding. This extension will enhance the integrated and interactive coding we describe in the next paragraph.

Coding activities and social settings within ClassroomImaging

After the classroom representation is created, the actual coding of the activities the single actors engaged in can begin. Precondition of course is that the user has already developed a coding scheme that captures those process categories that are of interest. ClassroomImaging particularly supports two objectives of coding, (a) the coding of the kinds of activities an actor engages in (e.g., arguing, listening, explaining, etc.), and (b) the social setting that is realized (i.e., whether the actor is part of a group or not; groups can range from dyads to the whole classroom, to also differentiate between small group and plenary phases). The coding of the activities of an actor can be done by assigning a code that can be chosen from a choice-list (fig. 2 middle) based on a schema file the tool is started with and that can be changed easily without any programming knowledge. The information about the social setting is planned to be possible in different ways, such as multi-select to group actors or selecting via area selection/framing with a free curve like in a drawing tool.

If video or audio data exist, coding can be done as the recording is running, and the assigned codes are stored together with a time stamp that indicates when the respective activity was shown by the learner. If logfile data exists, and if these carry time stamps, these time stamps can be transferred to each code that is assigned as well. The ClassroomImaging prototype can be configured with the time interval the researcher finds appropriate, e.g. ten-second-segments for each of which one code for each coding dimension can be given. A similar approach has been used in other video-based coding software, such as Videograph® (Rimmele, 2004).

Resulting data visualization within ClassroomImaging

To visualize the flow of activities within the classroom, ClassroomImaging's visualization output is dynamic, i.e. a flipbook-like representation that can be "played in sequence" that basically includes four features.

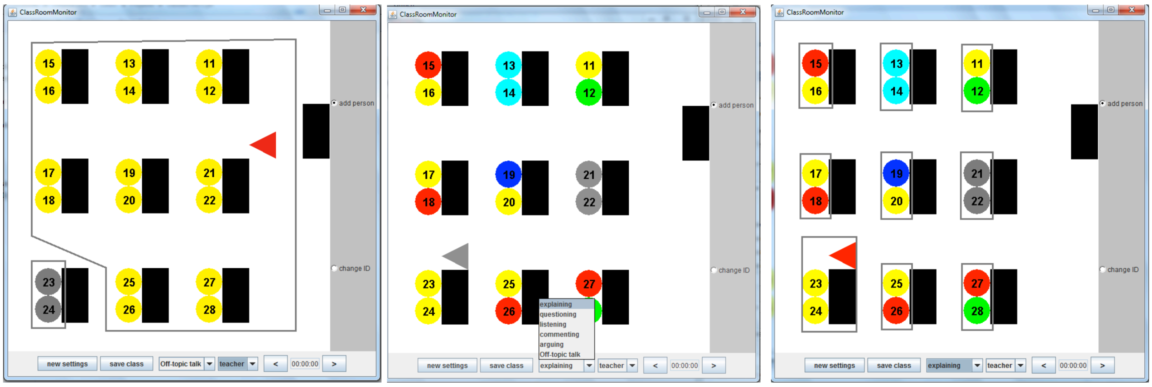

Figure 2. Coding and visualizing classroom situations.

First, based on the coding on the activity dimension, the symbol representing a particular student or teacher is color-coded. E.g., suppose that the coding scheme includes the categories "arguing", "explaining", "questioning" and "listening" (visible in fig.2 middle). ClassroomImaging associates each of these categories with a color, and whenever a student is coded as engaging in one activity, the circle with the number of the student is painted accordingly. The leftmost screenshot of fig. 2 shows a teacher (represented as triangle) explaining (color-coded in red) to the class in a plenary situation with most students listening (color-coded in yellow) and two students in the corner chatting off-topic (color-coded in grey). In the middle screenshot, the teacher's activity is coded using a choice-list of the used coding schema; when choosing the "explaining" code for the teacher, the color of the teacher symbol changes accordingly, which can be seen in the right screenshot of fig. 2. As the flipbook is running, the colors of each actor thus change dynamically, depicting the flow of activities that each single actor has engaged in over the course of the lesson.

Second, to depict changes regarding the social plane (Dillenbourg & Jermann, 2007) on which learning takes place (plenary, small group, individual), ClassroomImaging uses a "frame" representation to bind those actors together who currently form a group. E.g., if two students sitting next to each other form a group, the two circles that represent these students can be connected at the appropriate step of the flipbook (fig. 2 right). If students collaborate online, and if collaborators within a dyad are seated in different areas of the classroom, those two students are connected. If there is a plenary activity, all students who are attentive to this activity are connected with each other (fig. 2 left).

Third, to depict changes regarding the physical position of the actors (e.g., to account for situations in which the teacher is moving from one desk to the next, or in which new groups are formed that temporarily sit together to discuss) and the objects they currently use, ClassroomImaging affords the opportunity to drag and drop actors and objects at any time point of the flipbook. Stopping the run enables interactive manipulation of the respective time segment as described above in the coding section. For example, figure 2 left and middle show that the teacher has moved from the classroom center to the lower left corner where the chatting students are located.

Fourth, to provide the viewer with information on time, the resulting representation includes a time stamp in the bottom right corner (fig. 2). The interval steps in which the animation proceeds is oriented at the size of the segments that were initially set by the user.

Promises and challenges

We believe that ClassroomImaging bears a strong potential to help researchers code complex classroom data and capture the dynamic flow of learning activities within "messy" classroom situations. Regarding coding, ClassroomImaging offers useful features to integrate data from multiple data sources and to access them all through the software itself. The resulting data visualizations can be used for in-depth, descriptive analyses of the learning processes that go on within a given classroom and thus make the effects of certain "orchestrations" intuitively visible. E.g., it would be very interesting to see the immediate effects of a certain teacher activity (such as modeling) on the learning processes of the individual students by simply looking at the changes of the dynamic visualization right after the teacher has exhibited this activity. This detection of impact could also be supported by formal metrics for the differential between two time points. Such metrics can be easily implemented and might be helpful to identify "critical moments" in the classroom flow. Further, resulting visualizations might serve as material for teacher training. By inspecting and discussing a prototypical visualization based on the coding of a real classroom situation, teachers may for example become aware of possible, unwanted side-effects of certain teaching activities and be trained in their development of strategies to prevent those unwanted side-effects from materializing.

Yet, we also see some challenges ahead of us. First, coding remains cumbersome, as every single student's activity at each point in time needs to be coded by aid of a to-be-developed coding scheme. Thus, ClassroomImaging can only relieve time and effort in that sense that it offers opportunities for conveniently accessing multiple data sources within one tool. Yet, combining ClassroomImaging with tools for natural language processing that are able to automatically code verbal data (e.g., Rosé, Wang, Arguello, Stegmann, Weinberger & Fischer, 2008) seems promising. A second challenge is that the richness of the resulting dynamic visualizations is difficult to transfer to static publications, e.g. journal articles. While it is possible to create moment-to-moment visualizations as in fig. 2, this can only cover a very short time sample of a lesson. Currently, ClassroomImaging is much better suited to visualize classroom learning in more interactive formats (e.g., during conference talks). More work is needed to make its potential visible in static publication formats.

Despite these and other challenges, we believe that ClassroomImaging might become a powerful tool to analyze and visualize complex classroom learning that recently received a lot of attraction in the Learning Sciences. In principle, the tool could even be used on-the-fly while observing classroom situations, similar to paper-based observation sheets for group interactions. Such a "real-time mode" would have to be designed and evaluated carefully, but is an interesting extension of the current usage scenario.

References

Avouris, N., Fiotakis, G., Kahrimanis, G., Margaritis, M., & Komis, V. (2007) Beyond logging of fingertip actions: analysis of collaborative learning using multiple sources of data. *Journal of Interactive Learning Research, 18*(2), 231-250.

Dillenbourg, P. (2013). Design for classroom orchestration. *Computers and Education. 69,* 485-492.

Dillenbourg, P. & Jermann, P. (2007). Designing integrative scripts. In F. Fischer, I. Kollar, H. Mandl & J. M. Haake (Eds.), *Scripting Computer-Supported Collaborative Learning: Cognitive, computational, and educational perspectives* (pp. 275-301). New York: Springer.

Gassner, K., Jansen, M., Harrer, A., Herrmann, K. & Hoppe, U. (2003). Analysis methods for collaborative models and activities. In *Proc. of CSCL 2003*.

Hernández Leo, D., Nieves, R., Carrascal, J.P., & Blat, J. (2013). Signal Orchestration System for Face-to-Face Collaborative Learning Flows. *EC-TEL 2013*: 560-564.

King, A. (2007). Scripting collaborative learning processes: A cognitive perspective. In F. Fischer, I. Kollar, H. Mandl, & J. Haake (Eds.), Scripting Computer-Supported Collaborative Learning: Cognitive, computational, and educational perspectives (pp. 13-37). New York: Springer.

Kollar, I., Wecker, C., Langer, S. & Fischer, F. (2013). When instruction supports collaboration, but hinders learning – the case of classroom and small group collaboration scripts in the CSCL classroom. In Proc. of CSCL 2013 (pp. 256-263). International Society of the Learning Sciences.

Mercer, N. (2008). The seed of time: why classroom dialogue needs a temporal analysis. *The Journal of the Learning Sciences, 17*(1), 33-59.

Rimmele, R. (2004). *Videograph.* IPN Kiel. Retrieved from http://www.ipn.uni-kiel.de/aktuell/videograph/enhtmStart.htm

Rosé, C. P., Wang, Y. C., Arguello, J., Stegmann, K., Weinberger, A., & Fischer, F. (2008). Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *IJSCL, 3,* 237-271.

Settouti, L.S., Prié, Y., Marty, J.C., & Mille, A. (2009). A Trace-Based System for Technology-Enhanced Learning Systems Personalisation. *ICALT*, 93-97.

Suthers, D., Dwyer, N., Medina, R., & Vatrapu, R. (2010). A framework for conceptualizing, representing, and analyzing distributed interaction. *IJCSL*, 5(1), 5-42.

Tatiana Interaction Analysis Tool: https://code.google.com/p/tatiana/ (last access 2014-12-09)

van Boxtel, C., van der Linden, J., & Kanselaar, G. (2000). Collaborative learning tasks and the elaboration of conceptual knowledge. *Learning & Instruction, 10,* 311-330.

Weinberger, A., Stegmann, K., & Fischer, F. (2010). Learning to argue online: Scripted groups surpass individuals (unscripted groups do not). *Computers in Human Behavior, 26* (4), 506-515.

Acknowledgments

We thank Kilian Stampfer from Catholic University Eichstätt-Ingolstadt for his support in implementing the prototype of the ClassroomImaging tool.