Supporting Agile Classroom Orchestration With a Live Teacher Dashboard

Sven Strickroth[1] [0000-0002-9647-300X], Melanie Kreidenweis[1] [0000-0001-8659-525X], and Andreas Götzfried[1]

Institute for Informatics, LMU Munich, Oettingenstraße 67, 80538 Munich, Germany
sven.strickroth@ifi.lmu.de,

Abstract. This paper presents a real-time teacher dashboard optimized to support agile classroom orchestration based on a Kanban board. The implemented prototype allows teachers to prepare and implement lessons and monitor student progress in real-time. Teachers can easily duplicate Kanban boards for multiple groups and identify groups of students who are falling behind or excelling and may need special support. Students can access and use the prepared Kanban boards in a privacy friendly way, raise their hand virtually, and upload their work for review. The prototype was used and evaluated in two case studies. The results indicate that the developed prototype with its features including the dashboard is usable and helpful for preparing, implementing, and reviewing lessons. Implications and future work are discussed.

Keywords: Teacher Dashboard, Real-time Dashboard, Agile Teaching, Teaching Support Tool, Learning Analytics

1 Introduction

In schools, teachers are responsible not only for delivering the curriculum, but also for monitoring and guiding learning activities and student interactions. Task-driven learning strategies, such as station teaching and structured work plans, can promote student autonomy and personalized learning [1,2]. This allows students to self-regulate, control the pace of their learning, and adapt to their individual needs and preferences. Hence, such strategies allow for internal differentiation [2] and can not only be used for individual learning activities but also to orchestrate group work. However, gaining and maintaining an overview of the progress of individual students or groups in such a cooperative/collaborative learning scenario where students simultaneously work on (different) assignments is a significant challenge for teachers. An overview is essential to effectively provide mentoring, guidance, and feedback to students ([3] outlines the importance of feedback).

To address this overview issue, we proposed in [4] a teaching method based on agile principles – specifically Scrum and a Kanban board – and developed a digital computer-supported collaborative learning (CSCL) tool *AgileBoard4Teaching* to support the implementation of the method. The agile teaching method works as follows (cf. [4]): There is a Kanban board for each student/group that is split into three columns “ToDo”, “In Progress”, and “Done”. The teacher prepares (prioritized) tasks for the “ToDo” column. This can be repeated for multiple planned rounds. Then, the board is duplicated and distributed to the student groups. Group members select and commit themselves to tasks by moving these to the “In Progress” column and start working on them. After all (mandatory) tasks meet the criteria of the definition of Done (i. e., are moved to the “Done” column) and an (optional) teacher review was fulfilled, a new round starts as part of the iterative-incremental process with a new set of prepared tasks. The tool supports teachers as well as students in every step: Teachers can easily prepare the learning scenario and tasks (for multiple groups and rounds) using an authoring mode. Using the tool, both teachers and students can get a quick overview on the progress directly in class. However, the tool was designed to work completely offline on students’ devices. This was a design choice due to (very strict local) data protection regulations in schools (i. e., General Data Protection Regulation, GDPR). As a result, the teacher had to manually distribute the prepared boards to all students, walk around the classroom to check on student boards, and collect the final results at the end of the session. Therefore, the existing tool could not be used in scenarios where students were distributed in different locations as no central overview showing all group boards and no synchronization was available.

The main idea of this paper is to extend the existing offline tool with networking capabilities so that the teacher gets a real-time overview of the progress (i. e., the state of all assignments) of all student groups. This allows the teacher to easily identify groups and/or students who are falling behind and may need special support, or excelling groups who at a certain point may need less structured instruction from the teacher to create solutions beyond the mandatory tasks. This paper answers the following three research questions:

1. What features must the networked tool provide?
2. How should the user interface and the real-time overview be designed to be usable in class?
3. How is the networked tool used and perceived by teachers and students?

We extended the existing *AgileBoard4Teaching* tool and implemented a prototype with a real-time dashboard, an advanced online authoring mode for teachers, and other features to support in-class communication. Furthermore, the developed tool was evaluated in two settings, a field study in an 8^{th} grade IT course and a study with experienced teachers and teacher trainers. Therefore, this paper has two main contributions: First, the design rationale of the extended tool, and second, the evaluation results of the authentic use of the implemented prototype.

This paper is organized as follows: First, related research on tools and dashboards for teachers is reviewed and the research gap is identified. Second, the requirements for the networked prototype as well as the design, the features, and the dashboard are outlined. Third, the evaluation and its results are presented. The paper concludes with a discussion, a summary, and future research.

2 Related Research

This section discusses related research on specialized tools that support teachers in preparing and delivering their courses/lessons in CSCL settings, teacher dashboards, and existing (professional) tools for orchestrating agile projects.

Many specialized tools have been developed to support teachers in planning and implementing their lessons [5]. The *PLATON* tool supports teachers in planning (traditional) lessons with a graphical planning view and automatic feedback [5,6]. Since *PLATON* focuses only on the planning phase, a complementary tablet app called *munter* has been developed [7,8]. This app allows teachers to access their lesson plan and materials during class, take notes, and set a timer, but it does not provide a student interface, nor does it allow for orchestrating assignments or tracking student progress. To not only plan and model lessons, but also to implement them, Learning Design tools have been developed [5]. A notable example is *LAMS* [9], which enables teachers to design lessons with a set of predefined activities, conditions, branching, and group assignments. During the lesson, students use the system and work together step by step through the planned activities. *LAMS* also offers a view for teachers to track student progress. Additionally, there are other Learning Design tools (e. g., [10,11]) and tools for CSCL scripts (e. g., [12]) that allow users to model and implement learning scenarios. A general issue with the aforementioned tools is that they are limited to a special method (e. g., [12]) or allow teachers to specify flows of activities for specific learning scenarios using roles, groupings of students, and specific methods (e. g., [10,11]). Hence, these tools are not optimized or well suited for orchestrating task-based learning settings and/or agile teaching such as the method proposed in [4]. As mentioned before, the existing tool *AgileBoard4Teaching* (available as open source software on https://www.tel.ifi.lmu.de/software/agileboard4teaching/) only works completely offline and does not provide a comprehensive real-time overview of all groups for teachers.

There also is research on teacher dashboards that display real-time data of CSCL settings. This research is often conducted in the context of Learning Analytics with the goal of collecting and analyzing learners' data in order to understand and optimize learning and teaching [13]. Van Leeuwen et al. identified three types of support of CSCL teacher dashboards, namely mirroring, alerting, and advisory support [13]. The mirroring dashboards, which "just" display collected or aggregated information, have shown to enable teachers to more accurately interpret CSCL situations and allow them to provide more timely and effective support (e. g., [14,15,16]). Poh et al. conducted a literature review to identify general design principles for K–12 teacher-facing dashboards for in-person learning settings [17]. They found that dashboards typically support specific teacher tasks such as viewing data at student, group, and class levels, providing feedback and support, personalizing learning, monitoring/supporting collaboration among students, reflecting, and orchestrating activities. These tasks indicate that research on CSCL teacher dashboards has focused more on settings where teachers assign tasks to students or groups (e. g., [16]). However, these do not align with task-based learning settings, such as the agile teaching method, where students can choose the tasks they want to work on. Overall, Poh et al. derived general principles for the design of dashboards such as simplicity, glanceability, modularity, and informing to support sensemaking [17, p. 740].

Furthermore, there are (professional) tools for agile project management such as *Atlassian Trello* and *planio* (both provide a Kanban board) and more general tools such as the collaborative digital pinboard *Padlet*. These tools are, in general, not designed for the school context and are not pedagogical tools to support agile classroom orchestration as they offer many complex (professional) features, although, there are reports of successful use in school settings (e. g. [18]). A major limitation of such tools is that they are not compatible with European privacy laws (GDPR), particularly as the students are minors and the tools require user accounts. Furthermore, the tools are designed for (professional) projects and do not allow to clone a "board" (e. g., for multiple groups working independently on the same tasks, except for *planio*) easily, do not provide real-time overviews on all groups within a course, and do not offer educational features like requesting help/assessment from a teacher.

In summary, real-time overviews for teachers such as dashboards have proven to be helpful, but they seem to be very domain and use case specific in terms of what information is provided, how it is visualized, and what features are offered (cf. [13,19,16,15]). It is also important not to overload teachers with information (cf. [19,17,20]). The authors are not aware of any real-time dashboard approach for agile learning. Therefore, this paper addresses this research gap.

3 Designing the Dashboard

This section first presents the requirements. They are derived from related work, experience with the *AgileBuild4Teaching* tool, and feedback and discussion with teachers and researchers. Based on the requirements, the final design of the prototype and the dashboard is presented.

3.1 Functional and Non-Funtional Requirements

First, the functional requirements are presented: The teacher needs an authoring mode to prepare/design tasks including student visible details (e. g., mandatory/optional, review required, expected time frame) as well as the initial boards and rounds. Furthermore, the teacher must be able to efficiently instantiate a design for multiple groups and get a real-time overview of all groups with all relevant information. Also, communication features such as reviewing student work before tasks can be accepted in the "Done" column, setting a timer, and sending messages to groups are required. Students must be able to access and use their group boards without accounts. This includes setting a group name, moving tasks around, handing in work, requesting teacher assistance, (optionally) adding new tasks, and having a live overview of their group's progress.

As non-functional requirements, the tools needs to be usable by students and teachers. Furthermore, the tool needs to be usable without installation, e. g. in a browser, and robust enough to cope with unstable Internet connections (cf. [7]).

3.2 Towards the Networked Version

The original *AgileBoard4Teaching* tool has been implemented using JavaScript and the React framework (cf. [4]). Furthermore, the offline tool provides two modes: First, an authoring mode for teachers where they can prepare Kanban boards with tasks and rounds. The prepared boards can be saved to JSON files in which all details (such as optional images and further material) are included. Second, a student mode in which existing boards can be loaded from a file so students can work with their boards in their local browser, and save their current state to a file again. As the original version already employed web technologies, it could be extended easily.

To fulfill the requirements (cf. Sect. 3.1) in the networked prototype, a server-side back-end was needed, where all data is stored in a central place in a database, to which all stakeholders can connect to get the latest data. To protect the data, user accounts for teachers were introduced. The back-end has been implemented using Java Servlets. MariaDB was chosen as a relational database on the server side. For the developed prototype, the boards are saved as JSON strings in the database and are associated to an owning user account in order to stay compatible with the offline version. The prototype can be deployed as a Docker container.

Then, the user-facing front-end needed to be extended to be able to connect to the back-end to store and retrieve the data. For the online version, a login for teachers as well as "cloning" features for the group boards needed to be incorporated into the user interface. In the offline version, boards could easily be cloned by just copying the JSON files. Additionally, the existing authoring mode needed to be extended. It was designed with re-usability of the designs in mind for efficient lesson preparation. Teachers can create Kanban boards with tasks and rounds. Then, these boards can be instantiated for a specific class where groups can be created (cf. rows in Fig. 1). Each group is initialized with a copy of the (initial) Kanban board. Then, students can access their board with a randomly generated group specific join code and can set a group name. This way student can access their groups without the need for individual accounts. All join codes can be displayed at once to distribute them to the students easily/quickly. The teacher can also modify each group's Kanban board individually, collectively for the class, and can configure possible options for students such as allowing them to add their own tasks, how many tasks can be in the "In Progress" state at the same time, or whether work needs to be approved before tasks are accepted in the "Done" column or the next round can be entered.

When the website of the prototype is opened in a browser, there are two main options: First, students can enter their group code. Second, there is the login for teachers where they can access the authoring mode and the real-time dashboard.

The teacher dashboard as well as the board views for the students communicate using the WebSocket technology with the server. This allows the server to push notifications on changes in real-time to the clients which then retrieve the latest data using a HTTP(S) request. To cope with unstable Internet connectivity, the WebSocket automatically tries to reconnect to the server, and additionally, there is a regular automatic connectivity check as the current state of the WebSocket

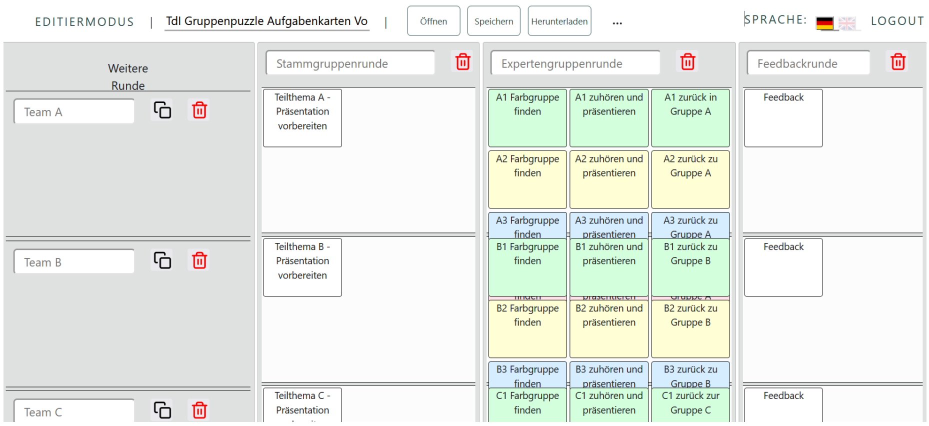

Fig. 1. Authoring view with three instances/groups and three rounds

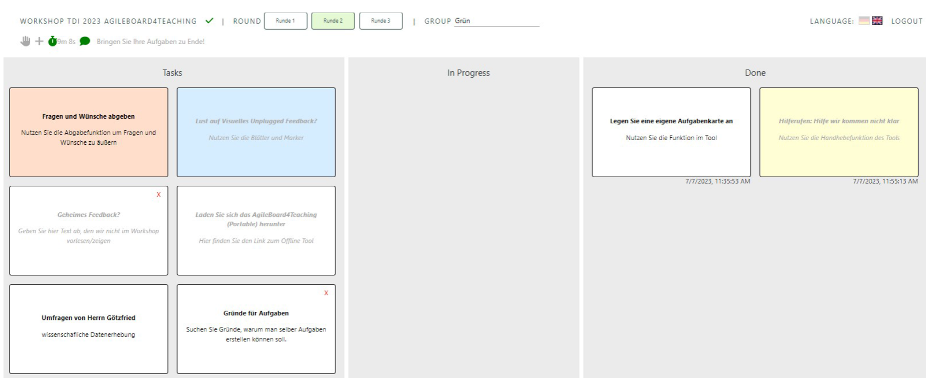

Fig. 2. Student view with the columns "ToDo", "In Progress", and "Done"

is not always properly reported by all browsers. The current connectivity state is displayed in the user interface (cf. green check mark in the header in Fig. 2).

All views follow the same pattern: At the top there is a header. It includes the title of the view (left hand side), buttons for interacting with the boards/groups and logout (right hand side). Below, the main content of each view is displayed.

The teacher dashboard is designed as follows (cf. Fig. 3): The header provides buttons for setting a timer which will also be visible to all students, sending a message to all groups, pausing/freezing the views of the students, and closing the current session for all students. Below, simplified overviews of all groups are displayed as boxes in a grid. For each box, the group name and their current round are displayed in the top. A double click opens the respective group's Kanban board. Below, there are three columns for each group. In the first column, there is an indicator whether students have raised their virtual hand (cf. Fig. 3, green hand in box in the top right). Furthermore, there are buttons to send a message

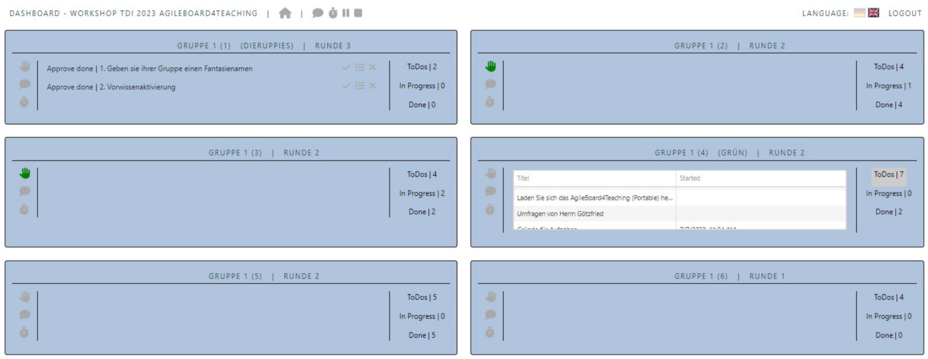

Fig. 3. Dashboard view with six instantiated groups

to a specific group and to set a timer for that group. The middle column, by default shows submitted work by students that needs to be manually approved by the teacher (e. g., box in the top left). The third and last column, provides an overview of the number of tasks in the respective columns of the Kanban board. When the teacher clicks on e. g. "ToDos", the middle column shows a detailed view of the tasks, including when students started working on a task (e. g., middle box on the right). After opening a group's Kanban board via double click, teachers can also edit tasks and move them to a different column.

In the student view, a Kanban board instance of a specific group is displayed (cf. Fig. 2). Students only interact with this view. In the header, students can switch to different defined rounds (the teacher may have to approve this) and can change their group name. In the second line of the header on the left hand side, students can rise their virtual hand and see the remaining time or a teacher message if these features were used by the teacher. In the middle, there is the Kanban board for this specific group displayed. This board is designed as specified in [4]. Students can interact with the tasks, i. e. they can start working on one by moving it to the middle column "In Progress" and assign it to group member(s). They can access additional task material as well as upload arbitrary files as results to a task and move it to the "Done" column. Depending on what the teacher has configured, they may need to approve this move from their dashboard.

4 Evaluation

The developed prototype was evaluated in two complementary case studies in Germany: First, it was evaluated in an authentic field study in a school. Second, a simulation study with experienced teachers and teacher trainers was conducted.

4.1 Methodology

Field Study In the first case study, a teacher-in-training used the agile teaching method for in-person teaching in an 8th grade IT course on computer-aided design

(CAD) for three weeks with two 45-minute lessons per week in a Realschule. The teacher first used the original, offline version at the beginning for one week and then switched to the extended, networked version with the real-time dashboard. This within-subjects design was chosen so that the teacher and the students can compare both tools. The tool was introduced to the teachers two weeks ahead of the study so that he could prepare all material, i. e., 5 rounds with an average of 5 tasks each. 18 students participated in the study, working in 9 groups of two on the same assignments. Tool usage was mandatory – no student refused to use it. Originally, one week was planned with each version of the tool, however, the project was not finished as planned in the second week and, thus, the study was extended by one week. Three of the eight lessons were observed by a researcher who took notes. The study was followed by a questionnaire (including a rating of the features, the System Usability Scale (SUS) [21], and open-ended questions) for the students, a 10-minute group discussion, and an interview with the teacher (including the SUS). In addition, log data from the use of the tool were analyzed.

Simulation Study The second case study was conducted in a two-hour workshop to present the agile teaching method to experienced computer science teachers and teacher trainers. The workshop was attended by 18 teachers on a voluntary basis and was itself organized in such a way that the developed prototype was used to orchestrate it. This allowed all participants to learn both the teaching method and the tool at the same time, and to get hands-on experience with a student board. Subsequently, the SUS questionnaire was filled out and a group interview was conducted to gain insight into the usability and estimated applicability in real classrooms.

4.2 Results

Field Study First, only the offline version was used. The students were able to use the tool without significant instruction. The prepared Kanban board data needed to be manually handed out and imported by all students as well as saved at the end of each lesson. This procedure took a significant amount of time.

After one week, the new, networked version was introduced. The initial opening of the groups' Kanban boards on the server using a group code caused trouble for two students, because not all letters were differentiable – this was easy to fix. No further issues or questions regarding the usability were raised and the tool showed to be intuitively usable by the students and teacher. As each student had his/her own computer, each group's Kanban board was open on two devices simultaneously. No synchronization problems could be detected. A drawback, however, was that the students had to import their old state of the offline tool themselves. Hence, the teacher did not use the dashboard much in the first lesson. The dashboard usage increased over the next lessons despite there was no tablet to carry around for the teacher available. Although, the dashboard was not used very often during the lessons. Particularly, the messaging and timer features were used to communicate the end of working phases and the start of the stand-up meetings at the end of the lessons.

The log data analysis shows that the tool was intensive used (≈ 365 server interactions per lesson) in class and also outside class. The teacher used the dashboard to review the student progress and to prepare the next lesson (at home). Interestingly, one student got sick and could not attend in person but joined his/her group from home to see their progress.

The questionnaire shows that the students significantly preferred the networked version (93 %), because of aspects such as increased productivity and no need to manually save the current state. Furthermore, the students stated that the structure helped them to document their work. The main features were rated by the students on a four point Likert scale (1 does not agree to 5 fully agree) on average as follows: *raise hand*: 3.9, *messaging*: 3.8, *timer*: 3.5, and *pause*: 3.5. The SUS score the students assigned to the networked version is 84 (“excellent”, cf. [21]). Notable is one student, who was interested in more information about finished tasks such as time tracking information. However, one student stated that the offline version gave him more room to interact with the teacher and one student reported a problem with vanishing student names of taken tasks on synchronization.

Overall, the teacher valued the live overview dashboard in the interview and highlighted being able to monitor multiple groups simultaneously. The dashboard was relevant for him and he liked the communication features as these reduced the need for discussion. During the lesson, there was not much time for approving assignments. He noted that he used the dashboard mainly after lessons to assess student progress and prepare for the next lesson. Two problems were reported with the creation of new tasks and drag’n’drop on a tablet. The teacher’s SUS score is 92 (“best imaginable”, cf. [21]). He would use the tool again if it were provided on a long-term basis. Finally, the teacher requested a slim variant with a focus on teacher’s action to prevent an overload with many groups, an extension for being able to add new rounds for all groups on the fly, and further views displaying a burn-down chart or aggregated data such as time taken or number of optional tasks picked.

Simulation Study The experienced teachers and teacher trainers in the workshop rated the tool as useful, although a bug occurred during the workshop which prevented some participants from entering the second round. They liked the live synchronization of the Kanban boards and the live overview of all the Kanban boards of all groups. The implemented features such as “raise hand” or “submit work for review” were seen as very helpful as a support for communication. Furthermore, the teachers underlined the efficiency and good usability of the tool, particularly they liked the authoring features to prepare Kanban boards and then being able to initialize/clone boards for new groups (which was possible even during a lesson). Overall, they expressed that they would like to use it in their own classes, but the fact that only a prototype is available diminished the excitement. The teachers requested a hosted and managed version. Finally, the teachers added that this tool now allows to conduct decentralized activities while still being able to closely supervise their students. Nine of the teachers answered the questionnaire. The overall SUS value is 69 ("good", cf. [21]) and the features are rated as follows: *raise hand*: 5, *messaging*: 4.3, *timer*: 4.4, and *pause*: 5. There were two optimization requests to auto save of text boxes and to make communication options more prominent.

5 Discussion

A challenge was to design the live dashboard. The goal was to provide as much information as possible without overloading the teacher and ensuring that all texts were still readable. Several different options were implemented and finally decided for the chosen design that provides an overview but also allows to view more detailed information on request. An optimization for many groups is needed.

In the field study, the students seemed to be very motivated to use the tool, but this may be caused by the novelty effect. The teacher valued the live overview dashboard, but did not check it frequently during the lesson (although no information was reported to be missing). In general, this was to some extend expected, as his main duty was to work with students in person. A similar result was reported in [8], but the factor of quantity vs. quality of the checked information regarding its value is still to be evaluated. However, it was unexpected that the teacher used the tool extensively to review the last lesson. Displaying the dashboard on a (separate) projector during class may increase its impact. Overall, the teacher found it useful for preparing agile teaching scenarios as well as for the review of student progress and their submissions to the tasks. Therefore, further research should investigate how the lesson review can be supported by providing more details about the learning progress (Learning Analytics, cf. Sect. 4.2).

In both studies, teachers expected a high initial effort to prepare the agile teaching session. This is, however, not caused by the used tool, but a characteristic of the method. Still, the teachers positively noticed the efficient duplicating of the boards and joining procedure and would use the teaching method as well as the tool to support it. In addition, the designs developed can be easily reused. However, a general limitation of prototypes was mentioned: getting them into production and ensuring long-term availability (cf. [22]).

A possible threat to validity is that the prototype was only evaluated in one course with one teacher-in-training. Nevertheless, both, the teacher and the students, provided valuable feedback. To complement this limitation, a second study was conducted with experienced teachers and teacher trainers to gain further insight. It should be noted that the participants in the simulation study seemed to be in a hurry at the end, so only half of the participants completed the questionnaire. Also, the low SUS score, compared to the other studies, may be explained by the bug that occurred during the workshop. Finally, the prototype was only evaluated in Germany and in an computer science context.

6 Conclusions & Future Work

In this paper, a real-time dashboard for task-based learning scenarios was proposed, implemented, and evaluated with a focus on supporting the agile teaching method proposed in [4]. The prototype provides authoring and cloning features for teachers, a real-time overview of all groups in a class, a privacy friendly join method for students, and communication features such as raising virtual hands, sending messages to students, setting timers, and reviewing student work.

The results of two case studies conducted indicate that the prototype with its implemented features and the dashboard are usable and helpful for preparing and implementing lessons. Although the real-time dashboard was not checked frequently in class, it was used extensively for reviewing lessons. The teacher in our study, experienced teachers and teacher trainers, liked the time-effective and privacy friendly group joining process, the authoring mode that allows for easy re-use and duplication of boards, the live overview, and would like to continue using the prototype. Students enjoyed working with the prototype, too. Still, the prototype needs to be further optimized before it can be used in production.

While the implemented prototype was originally developed for the subject of computer science, it provides a generic framework for novel teaching scenarios that are not limited to computer science and students working in groups at the same or different locations. The prototype allows to orchestrate homework assignments, to have different groups of students working on smaller, dependent parts of a bigger project, or to create competition like scenarios.

Future extensions could include learning analytics features such as tracking the time to finish tasks, average working time, deviation of expected time frames, or other measurements of performance or preference. These extensions could provide further insight to the teacher for self-reflection and (self-)assessment, but also to the students. However, transparency and privacy need to be considered as the main users are children.

Acknowledgement The authors would like to thank the teachers and students for their valuable input.

References

1. Zimmerman, B.J.: Attaining Self-Regulation, pp. 13–39. Elsevier (2000). DOI 10.1016/b978-012109890-2/50031-7
2. Bönsch, M.: Die differenziertheit der lernprozesse. Praxis Schule **21**(1), 8–12 (2011)
3. Hattie, J., Timperley, H.: The power of feedback. Review of Educational Research **77**(1), 81–112 (2007). DOI 10.3102/003465430298487
4. Strickroth, S., Kreidenweis, M., Wurm, Z.: Learning from agile methods: Using a kanban board for classroom orchestration. In: Proceedings 25th International Conference on Interactive Collaborative Learning, pp. 68–79 (2022). DOI 10.1007/978-3-031-26876-2_7
5. Strickroth, S.: PLATON: Developing a graphical lesson planning system for prospective teachers. Education Sciences **9**(4) (2019). DOI 10.3390/educsci9040254

6. Strickroth, S.: Unterstützungsmöglichkeiten für die computerbasierte Planung von Unterricht – ein graphischer, zeitbasierter Ansatz mit automatischem Feedback. PhD Thesis, Humboldt-Universität zu Berlin (2016). DOI 10.18452/17613
7. Elfreich, H., Strickroth, S.: Bedarfs- und Anforderungsanalyse für ein mobiles, unterrichtsbegleitendes Unterstützungssystem für angehende Lehrpersonen. In: Proc. Mensch und Computer (MuC), pp. 210–214. ACM (2021). DOI 10.1145/3473856.3474025
8. Elfreich, H., Strickroth, S.: munter: Ein mobiles unterrichtsbegleitendes Unterstützungssystem für angehende Lehrpersonen. In: Proc. DELFI, pp. 337–342 (2021). URL https://dl.gi.de/handle/20.500.12116/37032
9. Campbell, C., Cameron, L.: Using learning activity management systems (lams) with pre-service secondary teachers: An authentic task. In: Proc. ascilite (2009)
10. Derntl, M., Neumann, S., Oberhuemer, P.: Opportunities and challenges of formal instructional modeling for web-based learning. In: Proc. ISCL, pp. 253–262. Springer, Berlin Heidelberg (2011). DOI 10.1007/978-3-642-25813-8_27
11. Katsamani, M., Retalis, S.: Orchestrating learning activities using the cadmos learning design tool. RLT **21**(0) (2013). DOI 10.3402/rlt.v21i0.18051
12. Villasclaras-Fernández, E., Hernández-Leo, D., Asensio-Pérez, J.I., Dimitriadis, Y.: Web collage: An implementation of support for assessment design in CSCL macro-scripts. Computers & Education **67**, 79 – 97 (2013)
13. van Leeuwen, A., Rummel, N., van Gog, T.: What information should cscl teacher dashboards provide to help teachers interpret cscl situations? Int. J. CSCL **14**(3), 261–289 (2019). DOI 10.1007/s11412-019-09299-x
14. Tissenbaum, M., Slotta, J.: Supporting classroom orchestration with real-time feedback: A role for teacher dashboards and real-time agents. Int. J. CSCL **14**(3), 325–351 (2019). DOI 10.1007/s11412-019-09306-1
15. Molenaar, I., Knoop-van Campen, C.A.N.: How teachers make dashboard information actionable. IEEE TLT **12**(3), 347–355 (2019). DOI 10.1109/tlt.2018.2851585
16. Amarasinghe, I., Hernandez-Leo, D., Michos, K., Vujovic, M.: An actionable orchestration dashboard to enhance collaboration in the classroom. IEEE TLT **13**(4), 662–675 (2020). DOI 10.1109/tlt.2020.3028597
17. Poh, A., Castro, F.E.V., Arroyo, I.: Design principles for teacher dashboards to support in-class learning. In: Proc. ICLS (2023). DOI 10.22318/icls2023.197114
18. Garcia-Diaz, P., Garcia-Gomez, J., Redoli-Granados, J., de la Mata-Moya, D.: Case study: the use of trello for collaborative work in laboratory practice on engineering subjects. In: Proc. OT4ME. IEEE (2021). DOI 10.1109/ot4me53559.2021.9638838
19. Li, Y., Zhang, M., Su, Y., Bao, H., Xing, S.: Examining teachers' behavior patterns in and perceptions of using teacher dashboards for facilitating guidance in cscl. ETR&D **70**(3), 1035–1058 (2022). DOI 10.1007/s11423-022-10102-2
20. Van Leeuwen, A.: Learning analytics to support teachers during synchronous cscl: balancing between overview and overload. Journal of Learning Analytics **2**(2), 138–162 (2015). DOI 10.18608/jla.2015.22.11
21. Bangor, A., Kortum, P., Miller, J.: Determining what individual SUS scores mean: Adding an adjective rating scale. Journal of Usability Studies **4**(3), 114–123 (2009)
22. Bußler, D., Lucke, U., Strickroth, S., Weihmann, L.: Managing the transition of educational technology from a research project to productive use. In: Proceedings of the Software Engineering 2021 Satellite Events, vol. 2814. CEUR-WS.org (2021)