A teacher's perspective on developing a dashboard for intelligent tutoring systems to support students in class

J. A. Blankestijn

University of Groningen

Supervisor: Prof. Dr. V. Aleven & Prof. Dr. N. A. Taatgen

Date: 03-12-2021

Abstract

Intelligent tutoring systems (ITSs) improve student learning outcomes by providing step-specific feedback and adaptively assigning problems based on skill mastery. Instead of students' needs, this research is among the first to explore teachers' needs for displaying complex ITS data. We aimed to develop a dashboard displaying real-time student data from ITSs, to support teachers' decisions for helping K-12 students in different classroom contexts (i.e. remote or in-class) using a five-stage design process. Firstly, examining related research, which suggests teachers need to see what students are doing, get to them quickly, and correct struggles timely. Secondly, we storyboarded ideas with teachers about displaying problems, communicating and classroom contexts. In-class, teachers wanted two novel ways of displaying student work: a problem replay tool *and* interactive annotated snapshots. Thirdly, during prototyping teachers favored using snapshots and replay to review problems with students over investigating by themselves. Fourthly, we tested our design by simulating student data in real-time. Teachers suggested replay is versatile, but want to give feedback in the dashboard. They would primarily use replay to investigate students and analyze errors with students. Finally, we make recommendations for in-class and remote teaching and present a set up for pilot testing the tool in-class.

1. Introduction

There is a shift towards online learning, as is evident from the increasingly growing literature on online teaching and learning (Martin, Sun, & Westine, 2020), and massive open online courses (MOOCs; Joksimović et al., 2018). Due to an increasing use of educational technologies, the field of learning analytics emerged (Schwendimann et al., 2017). Furthermore, due to COVID-19, teaching activities have become (partly) remote. Additionally, other reasons like sickness and snow days could necessitate remote learning. All-in-all more student coursework is performed through (online) technologies. This necessitates research on how to best use learning analytics to help students and teachers.

Intelligent Tutoring Systems (ITSs) are beneficial for student learning. ITSs are adaptive educational technology, which can be used for complex problem-solving practice, like algebra. An ITS can support students by adapting instruction with step-by-step feedback based on continuously updated models of students' current state of ability (Holstein, McLaren, & Aleven, 2017). Hereby, ITSs enable personalized learning for students which helps because different students learn at different paces. That is, some students need longer while others do not need to repeat the same item as often (Molenaar, Horves, & Baker, 2019). Recent meta-analyses show ITSs increase learning outcomes compared to other interactive multimedia systems (Hillmayr et al., 2020; Du Boulay, 2016; Kulik & Fletcher, 2016).

So, research has investigated the student's perspective and found learning benefits, however, ITSs cannot and should not replace teachers. Namely, teacher-student interaction improves learning outcomes (Lee, 2020). Nevertheless, during remote learning, teachers have fewer chances for personal interactions with students. Therefore, tutoring software should support teacher decision making and facilitate teacher-student interactions. A common remote student complaint is one-sided learning interactions, which indicates that the quality of interaction depends on the technology (Shim & Lee, 2020).

When designing learning tools, teachers' needs often have not been considered (Sedrakyan et al., 2016). Research has been more student-focused than teacher-focused, especially in the area of ITSs. Notably, literature on teachers' needs for an in-class tool to support the use of ITSs to help them help students is lacking. Contrastingly, researchers acknowledge the need for teacher inclusion in designing orchestration tools (Mavrikis et al., 2019; Holstein et al., 2017). Teachers should be able to use learning analytics tools to effectively gain insight in student learning to facilitate fruitful interactions. Nevertheless, learning analytics tools have not received the empirical back-up they deserve to help teachers regulate student learning (Sedrakyan et al., 2020). Teachers indicate that learning analytics reports prompted them to interact with certain students in a productive manner (van Leeuwen, 2019). In short, teachers need a tool that helps them support and interact with students.

This leads to the question whether data summaries from ITSs can facilitate teachers to help students better. If so, how should classroom data as learning analytics be shown in order to support teachers' awareness, reflection, and decision making?

1.1 Theoretical framework

Our study mainly focuses on mathematical tutors, like Lynnette, an algebra tutor created using the Cognitive Tutor Authoring Tools (CTAT) from Carnegie Mellon University (Aleven, McLaren, Sewall, & Koedinger, 2009). It was specifically designed for 7th and 8th grade students to learn equation-solving (Aleven, Xhakaj, & Mclaren, 2017). For this it uses model-tracing for individualized problem selection (Koedinger & Corbett, 2006) and Bayesian Knowledge Tracing (Corbett & Anderson, 1995) to keep track of the student's current mastery state of specific algebra skills.

Improved learning outcomes were shown with Lynette, compared to a popular gamified system for teaching algebra (Long & Aleven, 2017). Furthermore, Holstein, McLaren, and Aleven (2019) developed Lumilo, an augmented reality system which allowed

teachers to view analytics about student problem solving in real-time, through glasses. Specifically, teachers could see performance indicators above students' heads, like a "?" indicating unproductive struggling.

Other studies indicated that showing performance measures that indicated studying students are struggling and at risk of academic failure improved learning outcomes (Sutherland, 2016; Macarini et al., 2019; Holstein McLaren, & Aleven, 2018). Sutherland's (2016) results indicated that learning took place after a period of struggling followed by persisting through that period by performing multiple attempts on similarly complex problems. Herodotou et al (2019) found that learning outcomes improved while using predictive learning analytics, i.e. predicting failure on the next assignment with demographics, course-specific designs and teacher features (e.g. workload and average use of dashboard).

Different student status indicators will likely have an effect on how teachers provide feedback and on which students to zoom in. More concretely, it is expected that teachers give more process-oriented feedback (cognitive step-by-step problem feedback) when someone is unproductively struggling than when someone is abusing the system. In the latter case teachers' feedback is more behaviorally oriented, that is, feedback on classroom conduct (Sedrakyan, Malmberg, Verbert, Järvelä, & Kirschner, 2020). In addition, time spent viewing an ITS dashboard may lead to more differential feedback. In particular, Molenaar & Knoop-van Campen (2019) found that teachers who consulted a dashboard more often gave more different types of feedback. Lumilo involved other status indicators as well, namely, abusing the system, productive versus unproductive struggling, and being idle (Holstein et al., 2019). Therefore, it would be interesting to see what the effect of these status detectors on feedback is.

One difficulty presented in the literature is that teachers want to have the tools to orchestrate students, before they want to use the new technology in a classroom setting. The

visualizations that people use for these dashboards should be targeted specifically to teachers in order for the technology to be adopted (Mavrikis et al., 2019). This indicates that teachers should accept the software in order for it to be the most useful (Sedrakyan et al., 2020; Herodotou et al., 2019). Two factors from Davis' (1989) Technology Acceptance Model (TAM) suggest ease-of-use and perceived usefulness as the critical influencers of accepting a new technology (for a meta-analysis on TAM, see Scherer, Siddiq, & Tondeur, 2019). Holstein et al. (2017) confirmed in interviews that decreasing adoption rate of Lynette was not due to perceived usefulness, but difficulties of use. Therefore, this study will use a participatory design methodology from the start to focus on teachers' needs to increase ease-of-use and perceived usefulness, contrasting to involving teachers only in later stages of the design process (Prieto-Alvarez, Martinez-Maldonado, & Anderson, 2017).

1.2 Research Question

The research question of interest is: How to design a (classroom orchestration) dashboard which displays real-time learning trends, performance measures and examples from intelligent tutoring systems to support teacher decision making in facilitating, detecting and reacting to (struggling) K-12 students?

The first part of learning trend and performance measures refers to statistics derived from individual students who are working with the Lynnette tutoring software. Problem related statistics include - but are not limited to - number of (sets of) problems solved, time spent, and number of actions (correct, incorrect or using hints). Furthermore, one can display mastery skill bars that reflect how well students have mastered problem set specific subskills. Finally, there are detectors describing the student's working state (Holstein et al., 2019) for abusing the system, idling, (critical) struggling, and doing well. Each of these variables could be represented on an individual level (and compared to the class) as well as a class level.

Another way to display a student's task performance is replay of the problem-solving process. Teachers who worked with Lynnette and its current dashboard have been interviewed and a common theme from affinity diagramming (Martin & Hanington, 2012) that all six teachers described was the lack of being able to "look over a student's shoulder" during remote classes (Lawrence et al., 2021). Looking over a student's shoulder could be facilitated by replaying the problem-solving process during problems where students were struggling. Potentially, replay might be too labor intensive to be workable for teacher use, or judged as not useful. Therefore, we also explore whether an interactive annotated "snapshot" representation of a problem might work better.

1.3 The five-stage design process

The design process was divided into five stages: discovering needs, defining and redefining needs, developing and testing prototypes, pilot simulation studies, and delivering a solution. These stages were produced by combining interface design recommendations (Sharp, 2019; Design Council, 2019) and the LATUX workflow. The latter is specifically aimed at designing learning analytics tools for instructors and students (Martinez-Maldonado et al., 2015). During each stage, designers can opt to re-iterate a previous step based on feedback or after solving errors (Design Council, 2019; Martinez-Maldonado et al., 2015).

The first stage, discovering needs, consisted of exploring research about teacher needs for student analytics. Second, defining and redefining needs was achieved by storyboard speed dating with teachers (Davidoff et al., 2007). Resulting quotes were clustered and analyzed using affinity diagramming (Martin & Hanington, 2012). Third, teachers were interviewed and interacted with high-fidelity prototypes, which were iteratively changed. Fourth, pilot simulation studies during which classroom data was simulated (similar to Replay Enactments; Holstein et al., 2019). Fifth, we describe a plan to deliver the design solution using pilot studies that test the system in the target environment: a classroom.

Stage 1) Understanding the problem (Discover): Finding teacher needs, tasks and goals.

The reader should consider this introduction as the first stage of the overall design process. Development of a teacher interface should be strongly supported by research on teacher needs for student analytics (Martinez-Maldonado et al., 2015). Instead of assuming teachers' needs, tasks and goals, one should spend time with teachers to discover them (Design Council, 2019; Martinez-Maldonado et al., 2015). Since Lawrence et al. (2021) interviewed teachers who used the same ITS, their conclusions were used as a basis. Furthermore, research on classroom orchestration involving ITSs has informed the initial understanding of the problem. Specifically, teachers indicated they wanted to see what students were doing, to have the ability to get to students quickly who need help, and to correct struggles in a timely manner (Lawrence, 2021).

2. Stage 2) Redefine the challenge based on discovery (Define): Speed dating with storyboards.

The second stage of the design process is problem definition to redefine teacher needs, tasks and goals (Martinez-Maldonado et al., 2015; Design Council, 2019). Here, speed dating, - that is, showing multiple storyboards in quick succession - was employed to validate user needs found in the first stage (Davidoff et al., 2007; Holstein et al., 2019). To set requirements for the system, speed dating session data in the form of quotes was thematically clustered using affinity diagramming (Martin & Hanington, 2012).

In order to explore multiple ideas in quick succession, we performed storyboard speed dating with participants (Davidoff, Lee, Dey, & Zimmerman, 2007). Sometimes contextual factors that play a critical role in whether the software will be adopted are only discovered until the software is deployed. To ensure insight in the most influential contextual and social factors people should be exposed to many variants of interventions (Davidoff et al., 2007). Ideas to explore in these variants emerged from user needs discovered in previous studies.

This resulted in speed dating's main aim: *Find out how to best display students who are working in the system, so that teachers can help them optimally*. To achieve this goal, three research questions were asked. First *What are teachers' needs for a display of student work so that they can help students optimally?* Second *What is the influence of classroom context on mode of communication and displaying student work?* Third *What extra ideas and improvements would teachers prefer to see in the deep dive problem review page of students?*

2.1 Methods

Participants and design.

In this study one male and seven female K-12 math teachers were recruited. Six teachers were teaching in the United States, one in Taiwan and one in Croatia. On average teachers had 15.8 years of teaching experience ($SD_{\text{experience}}$ = 11.3, *range:* 5-40). One teacher

taught 6th grade, two 7th grade, one 8th grade, five 9th, 10th and 11th grade, and three 12th grade. Before this study all teachers taught their classes remotely for some period of time. Only one of the teachers had taught a class in a hybrid fashion, that is, some students working from home with others in class. All but one teacher were expecting to teach in person classes after the summer holidays. The exception was a remote math tutor, not a high school teacher. For each study, teachers were compensated for their participation with 30$ per hour. Sessions took between 1 and 1,5 hours. The study was approved by the Institutional Review Board (IRB) in Pittsburgh (The U.S.) and the local ethics committee from Groningen (The Netherlands). For each study teachers were recruited from a list of high-school math teachers who had expressed interest in participating in educational technology studies.

Materials.

During each stage the Zoom platform was used to record and meet with participants. Storyboards were created using Figma (see Figure 1) and slides were used to present these as well as an introduction to the system. Meeting recordings were transcribed automatically, using otter.ai, then validated manually. Next, we used Miro (Miro, 2021) to create affinity diagrams.

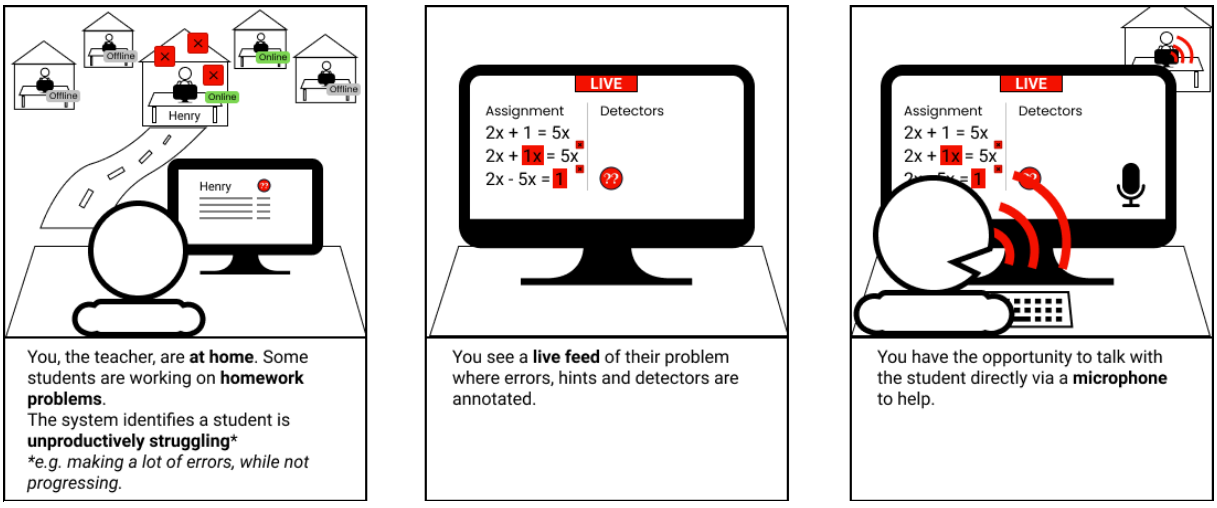


Figure 1. An example of a storyboard scenario. The left panel indicates the classroom context, e.g. remote teaching. The middle panel illustrates the display context, e.g. a live view of the student's screen. The right panel displays a communication method, e.g. audio communication.

Table 1

Matrix used to create storyboards.

			Type of problem		
		Communication type ↓	Unproductive struggle	Abusing the system	Being idle
Display context	Replay (step-wise)	Chat	$\mathbf{X}_{\text{Hybrid}}^{T}$		
		Microphone		$\mathbf{X}_{\text{Hybrid}}^{T}$	
		Draw & chat			$\mathbf{X}_{\text{In-class}}^{T}$
	Snapshots	Chat		$\mathbf{X}_{\text{In-class}}^{T}$	
		Microphone			$\mathbf{X}_{\text{Remote}}^{D}$
		Draw & chat	$*\mathbf{X}_{\text{In-class}}^{T}$		
	Live	Chat			$**\mathbf{X}_{\text{Hybrid}}^{D}$
		Microphone	$\mathbf{X}_{\text{Remote}}^{D}$		
		Draw & chat		$\mathbf{X}_{\text{Remote}}^{D}$	

Note. Crosses indicate storyboard context used. Three dimensions are displayed in the table. The fourth, classroom context, is noted behind the crosses.
^{T} Indicates that teachers were shown holding a tablet. ^{D} Teacher were shown behind a desktop PC.
* Tablet was shown vertically (portrait mode) instead of horizontally (landscape mode)
** Chat was switched out for *walk up to and speak with the student*

Our methodology for creating many contextual variants was inspired by Davidoff et al. (2007)'s matrix for systematically exploring ideas. Four dimensions were used to set up a matrix of different contexts (see Table 1). Firstly, during the study most teachers were teaching remotely, an unprecedented and unexplored scenario. Therefore, classroom context was included as a dimension (*in-class*, *hybrid* or *remote* teaching). Secondly, there were three main ideas on how to display student problems to teachers, the display context: *replay*, *snapshots* and a *live* interface of the student work. Thirdly, the type of teacher communication was included: *chat*, *microphone* or *draw & chat* balloons. Later a final idea was added in a single storyboard, the possibility for the teacher to *type in* the student interface. This storyboard is left out of the matrix for brevity. It used struggle, live, in-class, type in the student interface. Lastly, since the system uses thoroughly tested notifications about student

states to show to teachers (Holstein et al., 2019), these were used as one of the dimensions (*unproductive struggle*, *system abuse*, or being *idle*).

10 storyboards were created. This number was chosen to balance time constraints with teachers. That is, including many different variants while having enough time to attend to each. Every component (e.g. a live view) was included in a storyboard three times[1]. Finally, a storyboard displaying all variations was created and shown at the end of sessions to discuss (see Figure 2).

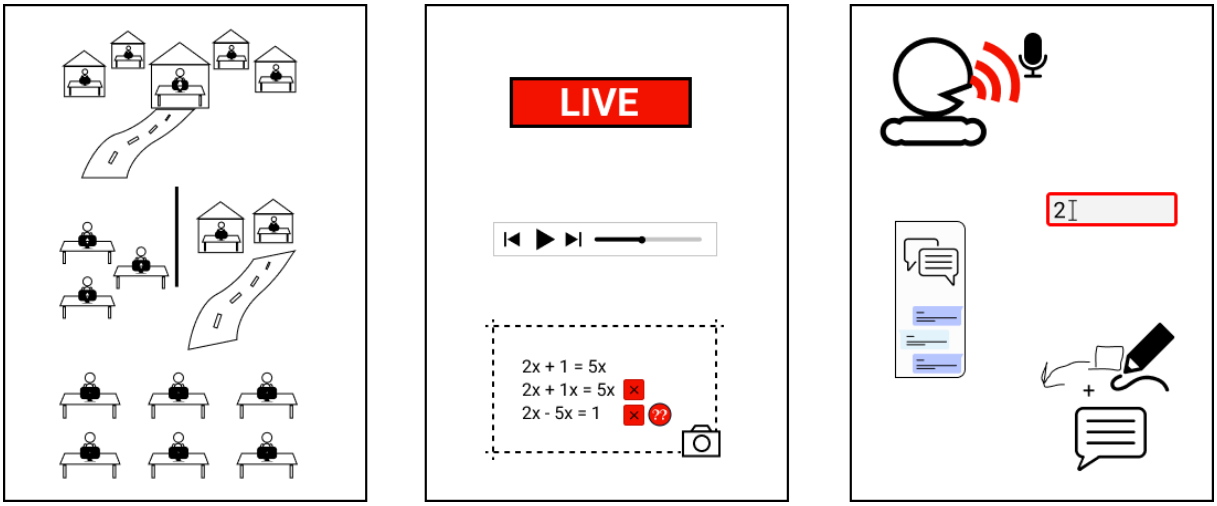


Figure 2. All individual variations of the storyboards. This storyboard is a combination of all storyboard variations. It was shown to teachers at the end as an overview.

Procedure.

First, we asked about the teacher's classroom context (last and next year), what grades and courses they teach, and how long they have been a teacher. Finally, we inquired about their previous experiences with (smart or adaptive) educational software?

Second, we explained the goal of the study, what teachers mentioned as needs before, and explained how the system currently worked. Two different tutors were shown during the explanation, a fraction addition and an algebraic equations tutor. We explained that CMU's

[1] Typing in the student interface was included in only one storyboard, since this idea came later.

math tutors are intelligent because they keep track of how well the students are performing on specific sub-tasks of algebraic math solving problems, and provide hints and feedback.

Third, we explained the notifications. There are three different indicators: idle (doing nothing, >5min), struggle (multiple errors at different problems, >2 problems; or a low mastery of a single sub-skill, >4 min), hint abuse (rapid guessing or using multiple hints, >2min).

Next, we show the different dimensions the storyboards vary on (see Figure 2), that is, classroom context (leftmost panel), how to display student work (middle panel) and mode of communication (rightmost panel). We told teachers that for every storyboard there are three central questions that we are interested in. First, whether they would use the technology shown in the storyboard inside their own classroom. Second, what they like or dislike about this scenario. Third, how would they improve the technology to be more suitable for their classroom. We encouraged teachers to think aloud as much as possible while looking at the storyboards. As previously mentioned, ease-of-use and perceived usefulness should be tested at an early stage of the design process (Prieto-Alvarez et al., 2017). Perceived usefulness is tested by asking whether they would use the tool in their own classroom. The questions about likes and dislikes, and improvements focus more on ease-of-use.

Affinity diagramming analysis.

The results were analyzed using affinity diagramming. Two researchers analyzed and interpreted the data. We discussed how to interpret quotes and the formal analysis plan. First, we checked whether the automatically transcribed transcription actually corresponds to what was said, and filled in the gaps. Then, below each storyboard, the corresponding transcribed data was added in a Miro board, with a color per storyboard. Next, the reactions were split into *atomic sentences,* i.e. sentences that are about single things (Martin & Hanington, 2012) and teacher codes (T1-T8) were added to trace quotes back to teachers. Two researchers

discussed how to split up teacher utterances into meaningful quotes about a singular topic, to make sure the quotes were somewhat inter-subjectively determined to be *atomic*. Subsequently, labels were added to each quote to indicate whether it was a negative or positive opinion, as well as which mode of display and communication they referred to. Then the three goals were laid out side by side and each quote was categorized into clusters. Two researchers discussed whether a couple of clusters seemed to belong under the same hierarchical cluster (e.g. “Being able to monitor students in real-time”). In that case a parent cluster category was created.

2.2 Results

Below the views of the eight teachers on benefits and downsides of display and communication modes are discussed. Using affinity diagramming an overview of how many teachers said particular things was created. Behind each argument an indication is given of how many of the n = 8 teachers mentioned the argument in the form of *n = x teachers*.

Monitor thought processes.

Being able to monitor student’s thought processes was mentioned almost unanimously as a benefit for each display mode: live (n = 7), replay (n = 7), and snapshot (n = 8). Contrastingly, a couple of teachers also noted that while teaching, time might be lacking to use a live view (n = 1) or replay (n = 2). For example, a teacher said, “I will be walking around. So, I would not have time to like check live, I would just go [up to the student]”

Synchronous versus asynchronous display modes. All three display modes provide ways to monitor students. A live real-time view allows timely feedback (n = 1), while replay and snapshot are inherently more asynchronous, thus would be less timely. Displaying a problem live allows teachers to view the students’ solution process in real time and see exactly where and on what they are currently stuck (n = 6). However, one teacher also

mentioned timely feedback in the step-based replay scenario (n = 1). Next, a live mode would allow teachers to direct students and guide them (n = 4). Which would not be possible with replay and snapshot, but this holds only for a remote teaching scenario. In class, teachers can take their laptop or tablet to students or let students come to their desktop PC and review it with them.

Information gain difference. The amount of information one would gain from each display mode differs. In particular, live displays would not allow teachers to view a student's problem history, i.e. what a student did before (n = 1). Replay would allow a history check (n = 2). Even though teachers did not mention snapshots specifically, snapshots would also allow viewing history. Most teachers (n = 6) felt that replay would allow the most thorough investigation of a student's problem. Specifically, because steps are important in math (n = 5), steps allow teachers to see where specific errors are (n = 2) and whether there are knowledge gaps (n = 1). One teacher said "having that ability […] to go through that, can be really helpful. Because I can see that […] maybe it is the same point that they are struggling on, or there are multiple, and address it from there."

Also, replay would give teachers examples of student problems (n = 2). First, this makes it easier to gauge what types of problems the students find difficult. As one teacher mentioned "[the] idea of seeing my past work or correcting anything based on past work might also apply to the teacher and student side here." Similarly, snapshots would allow one to see the student's process and not only their solutions (n = 3), which teachers mentioned as a flaw of other education software. Even though replay might allow the most thorough investigation of a single problem, snapshot's main benefit came from being able to determine patterns over problems sets (n = 5). Secondly, it could be checked quickly over entire problem sets whether hints usage was legitimate (n = 3). For example, abusing hints in the first

problem might be justified if a teacher can see that on a later similar problem the student now does that step correctly.

In direct comparisons, almost half of teachers mentioned they thought snapshots were quicker, and allowed easier pattern identification, but some said replay contained more relevant information. However, live displays allow giving immediate feedback, while replay and snapshot allow checking the student's history.

Idle students' thought processes do not need monitoring. The type of problem the student had can interact with the usages of a potential system. Namely, for the live (n = 1) and replay (n = 2), teachers noted that for an idle student it would not be helpful for them to see what the student did. Namely, "[idle behavior] is rarely somebody that is just going back through the steps and analyzing […] Idle is usually: I am bored, I am tired or something else got my attention." For draw & chat this point was specifically mentioned as well (n = 2).

Time saving.

Time is sparse during a 40–50-minute class. Perhaps unsurprisingly, most teachers indicated saving time - by being able to handle things more quickly - as beneficial in the context of live (n = 4), replay (n = 3) and snapshot (n = 2). Snapshots were assumed to provide the quickest at a glance overview (n = 2). Moreover, teachers noted it as a benefit of chat (n = 5), draw & chat (n = 3), and microphone (n = 1). For a live view (n = 4) and draw & chat (n = 2), the main argument of time gains was said to come from having the tools integrated in a single system. With a live view students would not have to share their screen (n = 3), thus all the tools necessary for helping would be in a single system (n = 2), given that a mode of communication is provided. However, it is noteworthy that for both chat (n = 2) and draw & chat (n = 1) some teachers mentioned that other options are available to chat (e.g. via the Video Conferencing tools) and annotate (e.g. screenshotting and using a standard

drawing tool. For example, someone said they "usually write, using Apple Pencil, on [their] iPad using Goodnotes, screenshot it and send it through Google Chat."

Privacy.

Interestingly, a particular topic that arose was privacy. Namely, students would not have to explain to the teacher what was wrong if the teacher could see it on screen. For replay one teacher noted it as beneficial, while for the snapshot three teachers noted that they liked that students would not need to explain, because "sometimes kids can't verbalize that". In the communication modes this point came up as well. Specifically, for type-in two teachers said it helped by not having to embarrass the students in front of the class. Moreover, with chat (n = 3) two teachers noted it would not embarrass students and two noted it would not interrupt the student or the class. Similarly, talking over a microphone remotely would not interrupt the class (n = 2). However, one teacher noted that in hybrid classrooms this privacy effect would be voided.

Participation increases.

Teachers noted that microphone (n = 7) and chat (n = 4) might allow them to increase participation in class. As one teacher explained, given the opportunity to talk to students, teachers can more easily use authority. This effect is immediate, as when students then feel observed they feel urged to work. Also, teachers can give wake-up calls to idling students (n = 6). Interestingly, for chat (n = 1) and by using a live view (n = 1) teachers noted that being monitored motivates the students to do their work. However, the main benefit for microphone (n = 6) and chat (n = 2) was to use it as a wake-up call for students who are idle or abusing the system. Although a teacher noted it would be easier to be authoritative using a microphone.

Preferred mode of communication. Teachers had differing opinions on what communication would help the most to get a class to participate. Namely, n = 3 teachers

mentioned that in this day and age students prefer to use chat. In contrast, n = 3 teachers mentioned that compared to microphone communication students would be more inclined to be non-responsive or find excuses to not have responded via chat. What is more, n = 2 teachers mentioned that microphone communication would have the issue of non-responsiveness as well.

A more coherent negative opinion from teachers about digital communication is that using mathematical terminology and vocabulary to communicate is difficult for students. This was especially expressed for chatting (n = 4). Furthermore, half of the teachers noted that audio chats would be preferred. Two found it more personal and another two mentioned it would allow shy students to talk more easily.

Way of helping.

For remote communication via microphone all teachers concluded that it was the most similar to one-on-one real-life contact. Therefore, “normal interactions” would be possible (n = 8). Teachers said it would be easier to talk about personal things and emotions (n = 2; e.g. to “make sure are they feeling okay? Not sick, not overtired, etc.”). Also, they could then coach a student through or redo a problem together. In short, verbal communication is well-liked in a remote scenario. For the other methods like draw & chat two teachers specifically mentioned that they preferred walking up to students to chatting. However, chat and draw & chat might still be beneficial in class according to teachers.

Helping via chat. First, sending chats around the class would allow for multitask helping (n = 4). For example, because “while you write one, you can go to another, write another one student. And when you work from one to another, you need more time.” Also, “just being able to quick[ly] shoot messages out while they're working and not interrupting them” was considered helpful. Moreover, a teacher might be able to quickly nudge someone in a specific direction (n = 1).

Second, chats would allow problem-specific feedback to be attached to problems (n = 4). This is similar to checking homework manually. This is useful "if kids want to go back and look at feedback. You can use it as assessments or practice, they can make modifications or changes." Additionally, it allows permanent feedback to be attached to problems (n = 2), which helps the student if the teacher "gave [the student] some pointers, or feedback [and the student] want[s] to go back and look at that." This was suggested as beneficial for draw & chat as well (n = 2). Similarly, audio chats, which were wanted by half of the teachers, would allow permanent feedback, thus students would be able to listen to feedback later (n = 1). Furthermore, two teachers mentioned this would be a first in terms of software they have worked with.

With regards to draw & chat, more than half of the teachers mentioned adding step-specific feedback and highlights (n = 5). This benefit is part of the overarching benefit of multifunctionality that draw & chat would entail, which all teachers mentioned. Similar to the point in asynchronous versus synchronous display modes, immediate feedback was named in draw & chat context by half the teachers. Also, teachers indicated draw & chat would be more dynamic than chat (n = 3). Finally, it is flexible (n = 1) and would for example allow small stimulators to be added (n = 1).

Caveats of digital communication methods. Draw & chat was highly regarded as multi-functional by all teachers during a remote scenario. However, half of them indicated that, in class, pen and paper would be preferred. Moreover, some worried about the accuracy of drawing on a tablet or using a mouse (n = 3). For microphone communication almost half of the teachers specifically indicated it would be redundant in class (n = 3), which was evident from its only benefit. That is, having almost normal interactions with students. Moreover, almost half of teachers mentioned they were happy with current options for microphone communication in video conferencing tools and other options that are available (n = 3).

Additionally, someone noted that chatting might reduce the amount of student thinking, because “They couldn't talk about and get [the] insight themselves, if I'm doing the thinking for them, or giving them a suggestion to go look at something”.

Similar to this argument about chat’s effect on student thinking, there were few positive comments about typing in the student interface. More than half of teachers mentioned this would take away student’s autonomy (n = 5). Similar to chat, but more consistently, teachers concluded that doing the work yourself makes for better learning (n = 4). One teacher noted that it might even create distrust between teacher and student because “if they [students] fail, they want to know that they caused they fail and that it is not because I intervene.” The only benefit that one teacher mentioned was the ability to nudge, it being a visual way of communicating things. For example, the teacher “would start very slowly typing it to see if they could take over”. All of these features are actually also manageable to implement in chat and draw & chat. One teacher also mentioned that this would be redundant in class (n = 1).

Preferred display combinations.

If there was time left after the other storyboards, teachers were asked specifically about their preferred communication and display combination in all of the classroom contexts. All-in-all six teachers responded to preferred combinations. All teachers preferred to be in class with their students.

In class combinations. Five teachers talked about their preferred combinations in class. In particular, four included draw & chat, three snapshots, three replay, one live and one chat. Three teachers indicated a preference for having both snapshots and replay plus a chat feature ($n_{draw\ \&\ chat} = 2$, $n_{chat} = 1$). One wanted draw & chat with replay. Another wanted draw & chat with live. Overall, draw & chat, replay and snapshots seem to be the preferred combination in class. Only a single teacher mentioned the hybrid scenario specifically in which they preferred to have replay, snapshots and draw & chat.

Remote combinations. Also, five teachers responded about the remote scenario. Similar to draw & chat's popularity in class, all five included draw & chat in their preferred combinations. Contrary to in class, four teachers included microphone and live. Two teachers found draw & chat with microphone best, two draw & chat with live feed and one preferred snapshot with draw & chat.

Ways to interact with students.

Two teachers wanted a feature in the software that could pause the student or the entire class. For the student it would be beneficial "to have a look at what they did" together. Furthermore, for the entire class it could be useful "if you notice […] half of your class has errors. You could pause everyone, do a few examples, or [show them common problems] and then work through some of those". If everyone is working on something different, then sharing a problem with the entire class (e.g. "by throwing it on the whiteboard") would be useful to do it together or divide people into groups and let them work on the same problem (n = 2).

What to send to students? As mentioned in the results for chat communication, half of the teachers would like to add audio chat messages rather than text-based chats. Furthermore, specific improvements apart from this were, sending a picture or video recording (n = 1), prefilling responses to chat (n = 1), and recording a proof of helping (n = 1) so that teachers can later show parents or use this in their own grading.

What data to show to teachers?

First and foremost, mentioned were common areas of struggle (n = 3). That is, teachers would like to see where common problems lie within the class, in order to timely intervene. Not only would common areas of struggle be beneficial for the entire class but also the individual student (n = 1). Next, the number of minutes spent per skill would help identify whether someone is actually having a misunderstanding or gap in knowledge for a particular

skill (n = 1). Besides this, the number of hints used over the entire problem set might help a teacher identify whether someone was actually abusing the system or using hints in a few problems only (n = 1). Finally, regarding the snapshot one teacher suggested hovering over hint indicators to make the hints that the student used pop up.

Notifications. Regarding the notifications, two teachers wanted these to pop-up on screen anywhere they went. Specifically, teachers mentioned that when looking at a particular student's problem, they would want to see another student's notification pop-up.

Student-side improvements.

Even though this study does not focus on the student's perspective, some ideas from this teacher study could extend to the student-side. For example, without much effort replay could be used on the student side as well. Two teachers said it would also be beneficial for students to review problems and previous mistakes. Finally, one teacher suggested not using red for errors. Red is associated with bad things the teacher said.

2.3 Discussion

Overall, draw & chat, replay and snapshots seem to be the preferred combination in class. Also, all queried teachers included draw & chat as a preference for remote teaching. However, most teachers included microphone and live for remote as well. For the purpose of this thesis, the focus will be on the in-class scenario. That is, the final design is created for in class use.

How to display working students?

Being able to monitor student's thought processes was considered beneficial for each display mode. This corroborates Lawrence et al.'s (2021) findings. However, time to use a live view or replay while teaching might be lacking. This will be investigated further during this thesis.

Replay and snapshots allow teachers to check the student's history. Almost half of teachers mentioned they thought snapshots were quicker, and allowed easier pattern identification, but replay contained more relevant information. However, live displays allow giving immediate feedback, view and direct the students in real time, but this was considered beneficial only for a remote teaching scenario. Notably, in Lumilo - the augmented reality glasses for the same tutoring software - teachers found a live display helpful (Holstein et al., 2019). Thus, on a screen teacher might consider it less relevant. Since live views have been tested before with Lumilo and teachers indicated they were not essential, they will not be implemented for the current thesis. Storyboards leave much to imagination regarding the dynamics of these tools. In an actual classroom environment, teacher opinions might be different. Thus, future research should investigate a live interface's merits in a classroom setting.

Saving time was considered as beneficial in all display and communication modes. Snapshots were assumed to provide the quickest at a glance overview. For a live view and draw & chat time gains came from having the tools integrated in a single system. However, as teachers indicated, these are replaceable by existing options.

One caveat with replay is that teachers might have felt like the replay would be time sensitive. Specifically, they might have assumed to see similar intervals to how long the students took to finish each step. However, it could take a long time for teachers to actually display this. Therefore, this should be tested with a functioning prototype.

It seemed like there were overlapping benefits for replay and snapshot. Specifically, showing student's thought processes was something that teachers found as the most relevant benefit. However, there seem to be specific benefits to both that make them less mutually exclusive. Snapshots are able to show patterns over multiple problems, while replay would not allow that as quickly. In contrast, teachers assumed step by step information would allow

them to more thoroughly investigate the problems, as well as being able to do problem reviews together with students.

Preferred way of communicating with students

Teachers worried about students' privacy. Using these communication teachers would not embarrass students in front of the class nor interrupt others, given the teacher can see what the student did (e.g. via snapshot or replay). However, in hybrid and in class scenarios this privacy effect would be voided.

Microphones are like real life, but other methods are available. Most teachers noted that these microphone and chat might allow them to increase participation in class. Interestingly, teachers noted that being monitored motivates the students to do their work. However, the main benefit for microphone and chat was to use it as a wake-up call for students who are idle or abusing the system. To speed up chatting prefilled responses could be used. However, both chat and microphone communication can lead to non-responsiveness. Moreover, communicating math via digital means can be difficult. Furthermore, almost half of the teachers mentioned they were happy with current options for audio communication in video conferencing tools and other options that are available. Since in class, one-on-one contact is possible, we consider microphone communication to be irrelevant.

Using the student's interface takes away autonomy. Most teachers agreed that typing in the student interface takes away the student's autonomy. Doing the work yourself makes for better learning (n = 4). The same holds for chatting, which might reduce the amount of student thinking. However, type-in might also create distrust between teacher and student. It gives a teacher the ability to nudge, but this can be achieved with draw & chat as well or by walking up to a student in class.

Chats allow quick permanent problem specific feedback. In class, half of the teachers would prefer pen and paper over draw & chat. Moreover, nearly half of the teachers worried about the accuracy of drawing on a tablet or using a mouse. However, chat and draw & chat might still prove beneficial in class. In general, sending chats around the class would allow for multitask helping by quickly nudging someone in a specific direction. Furthermore, most teachers agree that chats, audio chats, and draw & chat would allow permanent problem-specific feedback to be attached to problems. However, for draw & chat in particular, more than half of the teachers mentioned adding permanent as well as immediate step-specific feedback and highlights. Moreover, teachers indicated draw & chat would be more dynamic than chat. In short, draw & chat balloons seem to be beneficial in class and during remote teaching

What to implement?

Two teachers wanted a feature in the software that could pause the student or the entire class. While pausing the entire class, sharing a problem with the entire class would help. Desmos, an often-mentioned example by teachers, has the feature of pausing all or some of the students as well, which teachers noted to be useful. Furthermore, Nagashima et al. (2021) got similar feedback from teachers about this feature.

Half of the teachers would like to add audio chat messages (i.e. attaching audio recordings to problems) rather than text-based chats. This is part of a bigger concern teachers had, namely, recording a proof of helping. Finally, teachers indicated sending a picture or video recording would be interesting.

Since teachers mentioned common areas of struggle and this idea (of displaying common areas of struggle) was also implemented in Lumilo (Holstein et al., 2019), these are included in the current system. Next, the number of minutes spent per skill and the number of

hints used over the entire problem set are included. Finally, in snapshots hovering over hint indicators make the hints that the student used pop up.

Some teachers mentioned they wanted notifications to pop-up. However, they are not yet familiar with how often these will pop up in class. Thus, the number of notifications that come in at a time needs to be tested in order to not overload teachers.

Limitations.

There are different ideas that could also have been treated with the storyboards. In particular we were wondering what teachers' opinions were regarding retrieving a problem involving similar skills as the one a student (or multiple students) is struggling with. Furthermore, the option of giving that problem, or a problem, to all students was an idea that was interesting to explore. Therefore, these things will be investigated later.

A benefit as well as a limitation is that storyboards can be interpreted differently than a researcher intended. Some things, like how the class overview and notifications would look like were intentionally left vague. In the snapshots, replay and live view, errors, hints and detectors were said to be annotated. To create more distinction between the concepts this could have varied per storyboard. However, we considered it a benefit that all this data was similar, so that the overarching display concept differences were highlighted more.

We presented the storyboards to all teachers in a fixed order, because it made sense to introduce teachers in an organized way to the different ideas. Moreover, there are so many interactions with all the different storyboards that order-effects should be negligible. However, it might be better to remove order influences.

Conclusion and building a prototype.

This thesis focuses on building a teacher dashboard for in class use. As indicated in the previous paragraphs, there are different needs for in-class versus remote teaching. In class,

teachers wanted snapshots *and* replay, thus we built those for the next stage: high-fidelity prototyping.

No communication method. The most requested communication method was draw & chat. Since draw & chat is complex to implement, half of teachers preferred walking up to the students, and communication methods require building on the student-side, it will not be implemented in the prototype.

3. Developing an initial prototype

From Lawrence et al. (2021) three main teacher needs arose: to see what students are doing, give the ability to get to students quickly who need help, more generally, helping students, and correct struggles. Based on storyboard speed dating, the front-end implementation we created consisted of three main parts. First a *class overview*, which has a student table and a notifications panel (see Figure 3). Second, for each student, a *deep dive* screen containing information about areas of struggle, problem sets (and their corresponding problems), and a *snapshot* for individual problems (see Figure 4). Third, a *replay* problem review tool, which can play each action a student performed in a problem using video-player-like controls (see Figure 5).

3.1 Class overview

As the storyboards indicated a table view of all students would be created, which displays all the students progress and their notifications.

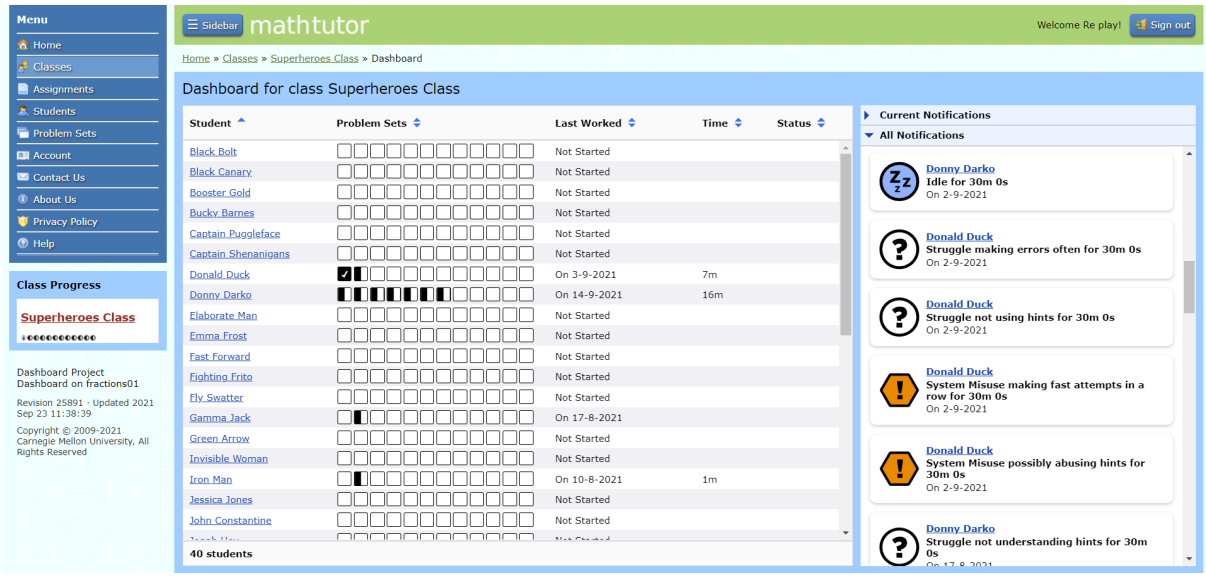


Figure 3. The class overview. It contains a table view of all students on the left. The problem sets column displays progress on the problem sets, with half filled blocks indicating that someone started the set and fully filled meaning they finished it. On the right notifications are shown. There are notifications for system misuse (orange '!'), idle (blue 'zzz') and struggle (white '?').

The left side of the class overview contains five columns: student name, problem set progress, last worked, total time of working, and current notification status. In TutorShop, the server for Carnegie Mellon University's tutors, teachers can add students to so-called *assignments.* An assignment consists of one or more *problem sets.* These problem sets contain *problems*.

Teachers want to see what students are doing. Therefore, an indication needs to be present whether students are working (i.e. the last worked column) and what they are working on (i.e. problem sets column). The choice was made to indicate progress on problem sets using blocks. Each block could be in one of three states. First, no-fill, indicating the student has not started working on problems in the problem set. Second, half-filled, indicating the student has started working on problems in the set. Third, filled with a checkmark, meaning that the student finished all problems in the set.

Teachers also want to get to students quickly who need help and correct struggles timely. This is where student name links, the status column, and notifications come in. Clicking a student name or a link inside a notification takes the teacher to the deep dive screen. The status column displays the last notification state known about the student by the system. As previously mentioned, there are three types of notifications. Notifications have been based on extensive prior research with teachers (Holstein et al., 2019). They work with detector files which run while a student is working in the tutoring software. Each (trans)action a student performs is tracked and labeled (e.g. *is hint step*). The struggle and system misuse detectors analyze a window of transactions. When certain conditions are met (e.g. 3 out of the last 10 transactions were hints) the detector fires and a notification is sent to the dashboard for the teacher to see. For more information on notifications see Appendix C, the CTAT Detector Library (2018) or Aleven, McLaren, Roll, & Koedinger (2006).

During storyboarding one teacher indicated that notifications should pop up everywhere. Instead, the choice was made to first implement the notifications as a panel in the class overview because we worried that notifications might come in too frequently and create too much distraction for a teacher when viewing the deep dive or replay screen. Furthermore, during the next stage, high-fidelity prototyping, notifications are static. This means that no new notifications will come, since there are no students at work. Thus, during prototyping teachers can only view the notification history. Therefore, it makes sense to leave pop up notifications out for now.

3.2 Deep dive screen

Then there is the deep dive screen. It consists of two panels, the left showing areas of struggle and a problem overview, the right showing a snapshot of the currently selected problem.

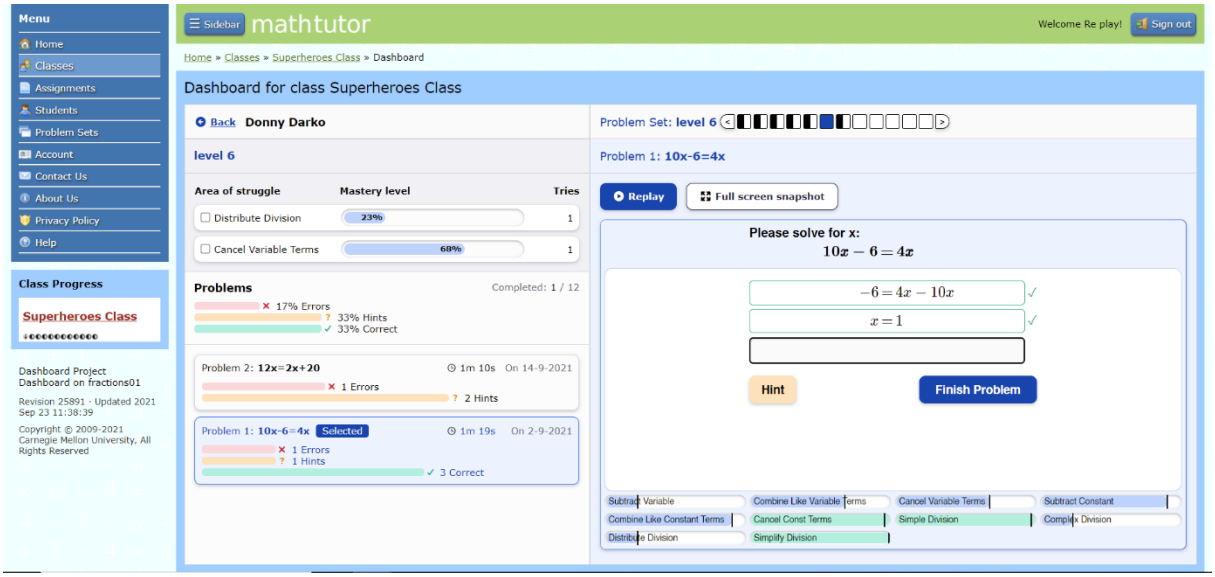


Figure 4. The deep dive individual student screen. On the left, areas of struggle are shown. That is, skills that a student has the most difficulty with. Below that is an overview of the problems, with error, hint and correct steps represented in graphical bars. On the top right an overview of the problem sets is given. Below that the selected problem in the current problem set is displayed. In this version the snapshot of the problem only contains the final correct steps, so errors were not displayed.

Areas of struggle.

There are multiple reasons for implementing areas of struggle. Teachers indicated a need for seeing areas of struggle for students during storyboarding. Moreover, correcting struggles, a need identified in Lawrence et al. (2021), requires the teacher to know where struggles lie. In addition, Holstein et al. (2019) identified this need as well. They implemented an algorithm for determining which skills students struggle on. This algorithm was implemented in our dashboard as well. It determines areas of struggle were as follows.

First, filter out all skills that are used in a problem set, but have not been attempted by the student. Next, sort all skills in ascending order, according to skill mastery level, i.e. p_{know}, a measure in Bayesian knowledge tracing indicating the probability a student has mastered a skill (Corbett & Anderson, 1995). Then, extract the three lowest skills from this. Afterwards, sort these skills by attempt number, that is, the number of steps inside each of the problems a student did that had an effect on the skill level. This results in the order in which the areas of struggle are shown in Figure 4. On the right side of each skill, attempts per skill are displayed. During storyboarding teachers requested minutes per skill, but since the algorithm uses attempts, it seems clearer to display attempt numbers.

Problem overview.

Each problem set contains multiple problems. Teachers want to get to students quickly who are struggling and see what they are doing (Lawrence et al., 2021). So there has to be some way to get to these problems. This is why the problem overview panel is shown on the bottom left of Figure 4. During storyboarding teachers requested, number of hints used over the entire problem set to be able to judge system misuse. Moreover, judging where difficult problems lie might be difficult without error frequency information. Therefore, we chose to present the user problem set percentages of hints, errors and correct actions and hint, error and correct system counts on a problem-by-problem basis.

Snapshot.

In Figure 4 (right-side) the snapshot is shown. Both snapshots and replay are able to show the specific student steps history, which was of the highest importance for teachers as the storyboard sessions indicated. Snapshots were judged to be the quickest way of displaying and analyzing student information by teachers during storyboarding. This is important, since time is limited for teachers. Also, teachers expected to be able to easily find patterns over multiple problems. During this stage of the project the snapshot was implemented as an overview of all the correct steps a student performed in a problem. As shown in the storyboards, the snapshot will be annotated with hint, correct, and error actions. As requested by a teacher, hovering over hint annotations will display the hints.

3.3 Replay problem review

The replay tool shows all the transactions a student performed on a step-by-step basis (Figure 5). This step-by-step solution process display was judged to be the most important benefit of replay by teachers during storyboarding. It is implemented as a draggable bar containing buttons and a slider. It is inspired by media player controls, because these interfaces are familiar for most people. This makes the replay tool intuitive to use. As shown on the slider, each correct (green '✓'), error (red 'x') and hint (orange '?') step in the student's solution process is annotated. One can drag the slider around, click on the annotation on the slider or use the buttons to show specific steps the student performed.

The interested reader is referred to Appendix C for an overview of the system's architecture and to Appendix D for ideas that teachers came up with but that were not implemented in the dashboard.

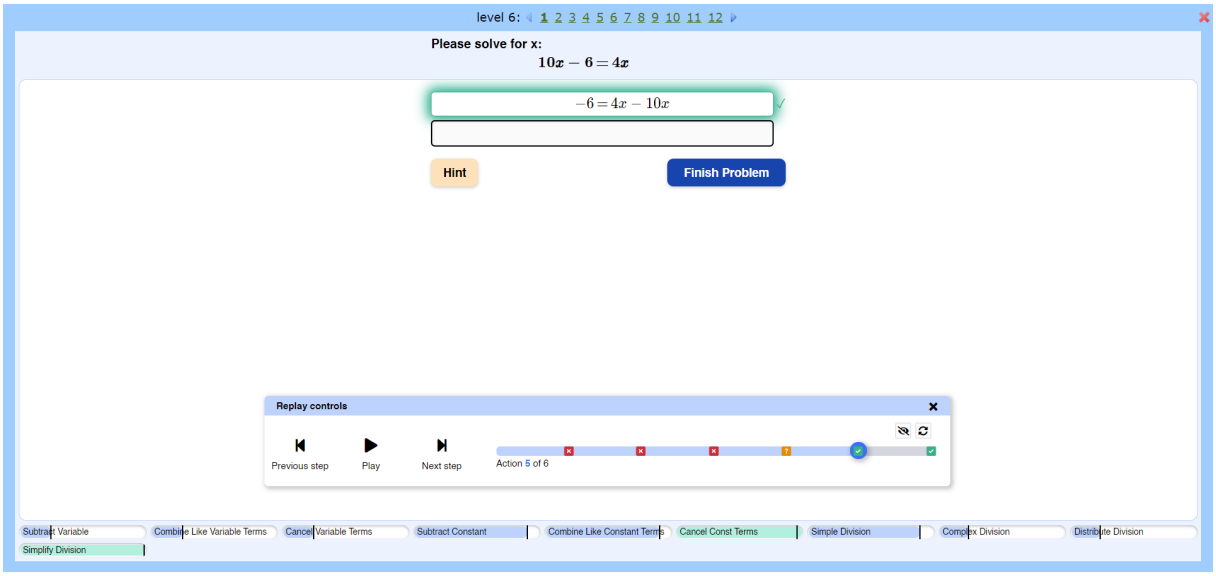


Figure 5. The replay controls. Each error, hint and correct step in the problem is highlighted (with a corresponding red, orange and green shadow) when it is replayed. The controls are shown in the bottom above the skill bars. There are previous, play and next buttons, which take the user to the previous step, play the problem (with a 1 sec interval between steps) or take them to the next step that a student performed while they were working in the tutor.

4. Stage 3) Rapid high-fidelity prototyping (Develop).

In class, teachers wanted to see what students are doing with snapshots *and* replay, so we implemented both for high-fidelity prototyping. Both tools allow teachers to see students' problem-solving history. Snapshots can be used for quick analyses and identifying patterns over problems, while replay gives a more detailed rendering of the student's solution process.

4.1 Introduction

It is important to use a participatory design philosophy to find how well requirements have been met, find improvements and start testing usability. During this stage it is important to collaborate and co-create with teachers (Holstein et al., 2019; Lawrence et al., 2021; Sharp, 2019; Martinez-Maldonado et al., 2015). In addition, researchers should communicate visually by showing ideas to develop a shared understanding with teachers (Martinez-Maldonado et al., 2015). We communicated visually with teachers by letting them interact with a high-fidelity prototype of our system.

The fidelity of a prototype refers to the associated level of realism of the prototype. In earlier stages of a user-centered design process, one can opt for lower fidelity prototypes (e.g. on paper) that investigate information processing and cognitive needs. In later stages it is more common to inquire about physical needs using higher fidelity prototypes, e.g. interactive computer-based prototypes (Hall, 2001). In design research low-fidelity prototyping often takes place during this stage, because they are easy to create, thus faster to test multiple ideas with. However, to test replay, it makes sense to create a system that can use real data to test with. If one wants to test ideas with this amount of complexity and tool-specific as well as environmental interactions, one should attempt to test concepts with higher fidelity prototypes, to increase the likelihood that ideas are actually usable.

The first goal of this stage was to the ease-of-use of the prototype. Secondly, we wanted to know how well the system could be integrated in class. Thirdly, we aimed to find out what information teachers require. We used a think-aloud protocol while assigning teachers tasks in the system. Each prototyping iteration the system's design was improved based on teacher feedback.

4.2. Methods

Participants and design.

In this study four female K-12 math teachers and one male math director were recruited (M_{age} = 52, SD_{age} = 12.3). Three teachers were teaching in the United States, and one in Taiwan. On average teachers had 18.75 years of teaching experience ($SD_{experience}$ = 15.95, *range:* 6-40). Currently, one teacher taught 6th grade, one 7th-12th grade, one 9th grade, and one 9th-11th grade.

Procedure.

The introductory setup was similar to the one used in storyboarding. That is, we explained the system and detectors. Next, we asked the teacher to do a student problem on their own to familiarize themselves with the system. Teachers were encouraged to use hints and deliberately make errors, to help them understand all the mechanics that students would go through. After doing a couple of problems teachers were asked to log into the system on a teacher account. Then, they performed 18 tasks, 4 tasks in the class overview, 6 tasks in the deep dive screen and 8 tasks in the replay problem review controls. Finally, they were asked to report any potential problems they noticed, if they were to use it in class.

The tasks were written down in a table, as shown in Appendix B. Questions were developed to guide the teacher through the system and tasks. During each task three metrics testing affordances of the dashboard were monitored and subjectively evaluated: whether it

was clear what action to perform, clear the action was available, and easy to comprehend whether it was right or wrong. Teachers were informed about the metrics and encouraged to think aloud (about them). During some of the tasks, teachers were asked a couple of questions related to the goals.

Usability analysis.

Metrics were scored by the researcher on a five-point Likert scale (1: very unclear – 2: relatively unclear – 3: not clear / not unclear – 4: relatively clear – 5: very clear). Moreover, for each task potential problems were noted. This created a table with rows specifying the tasks, e.g. "go to the previous step the student took in the problem". There are three columns dedicated to each of the clarity questions, and one for notes on potential problems (see Appendix B). After the prototyping sessions the researcher rewatched the recordings and rated all three columns had been rated, for each task and each user. When a task was not performed by a user, no rating was given. Next, an average was taken of each cell in the table. Then, for each row, the three column averages were averaged to get a final task clarity rating. Each rating fell into one of three classes, low (clarity < 4), medium ($4 \leq$ clarity < 4.5), high ($4.5 \leq$ clarity). Tasks with lower task clarity ratings are more unclear, thus, less usable, than other tasks and indicate where the system can be improved.

Design changes between sessions.

In between the sessions, we made the following changes to the dashboard, informed by the observations during the session. Ts refer to teachers and the associated numbers to the order in which they participated. Since five teachers participated, this section described T9 to T13.

Figure 6. Changes made to error, hint and correct bars.

Between T9 and T10. On the dashboard, notifications were made to navigate users to the student deep dive on click. On the deep dive, in progress indicators were added and words explaining the bars were added (hints, errors and correct; see Figure 6). Namely, T9 clicked on the problem sets as well as the notifications to go to the student, but that did not work then. Moreover, the words "action summary" were added to explain the summary. Lastly, the snapshot panel was made to display the same information as the replay. Indicators for errors, hints, and correct steps are added in the top right corner of the corresponding text field that the student typed in (see Figure 7). They were placed next to each other in a row if, on a single text field, the student tried multiple inputs or hints. Hovering over the indicators or clicking shows the action that the student typed or which hint they saw.

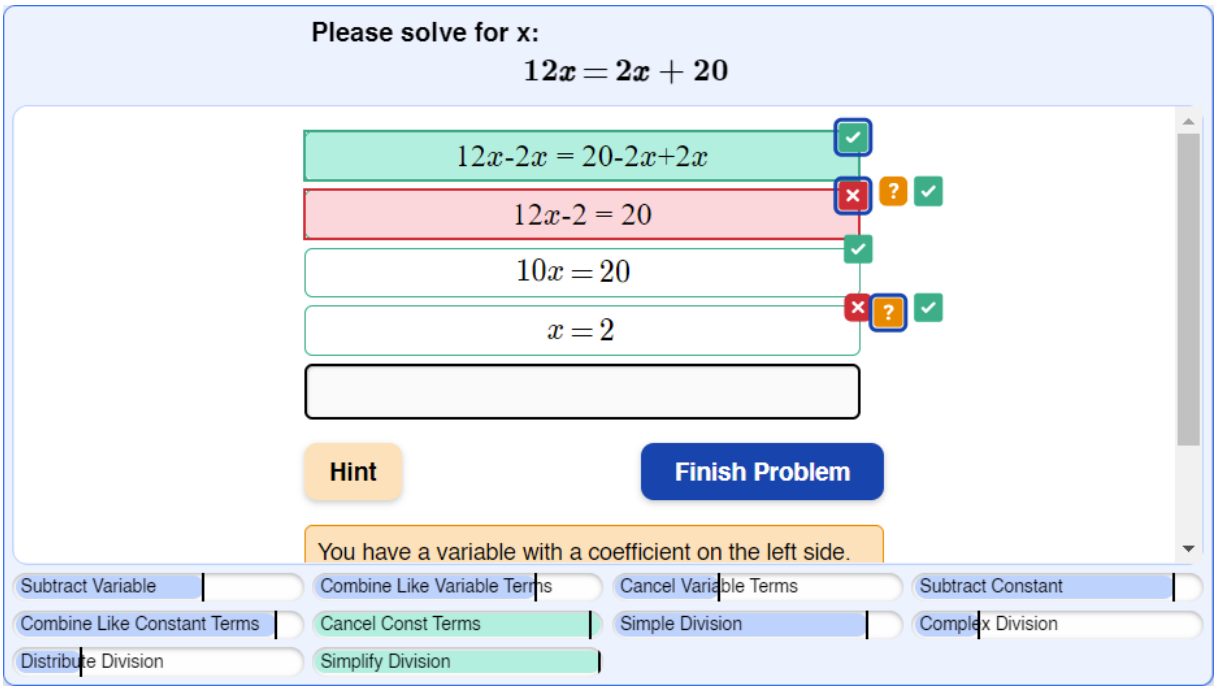


Figure 7. The snapshot. Each error ('x'), hint ('?'), and correct ('✓') step in the problem is indicated with a small square. When hovering over or clicking the box the action the student took is displayed in its corresponding location with a lighter red, orange or green background.

Between T10 and T11. Notifications were made to include problem and problem set names (see Figure 8). Furthermore, on the deep dive problem overview, problems display a small notification detector icon. Finally in the replay, skills are coupled to steps and light up

when a step involves a skill. Namely, both teachers before T11 mentioned that they would want to know what skill a student is working on and whether they can determine that multiple students have difficulty on that skill. Also, one teacher mentioned they would like error information that elicited the notification. T9 stated: “They got it wrong for some reason. What is the reason? What did you do wrong that got you there?” Showing errors is a feature that is implemented in Lumilo (Holstein et al., 2019). However, since the current system is aimed to be used with multiple tutors that have different formats, it would be difficult to find a way to condense this information down into a notification.

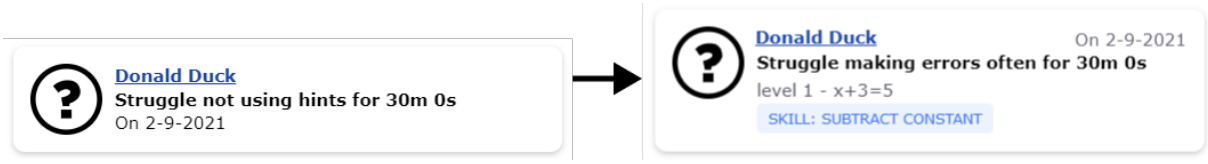


Figure 8. Changes made to the notifications during this stage. The level denotes the problem set, while x + 3 = 5 denotes the problem. Furthermore, a box containing the skill that the student was struggling with is included.

Between T11 and T12. Both T9 and T10 indicated that it was not immediately clear that half filled blocks indicated that someone got started. They interpreted it as being halfway through the problem. Therefore, after T11, problem sets do not display half blocks, but percentual progress (see Figure 9). Moreover, skills are now visible in the notifications (see Figure 8). After the proportional progress block change, teachers noticed that the filled-in portion of the blocks indicated that progress on “lessons” was being made. Thus, they were able to interpret them correctly.

Figure 9. Percentual progress indicated in problem set boxes.

Between T12 and T13. In the notifications panel, we added tabs so the users can switch between displaying all notifications (i.e. the entire notification history) versus only the current notifications (i.e. notifications from today; see Figure 10). Filter (on notification type

or skills) and sorting (alphabetically or recency) functionalities were added. Finally clicking on a problem set or notification leads to the specific problem set or problem that a notification went off on.

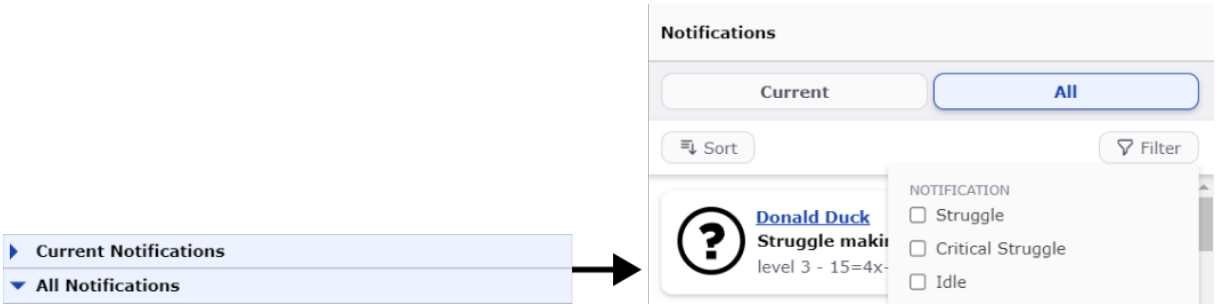


Figure 10. Notification panel design changes. A tab option for current and all notifications was added. Moreover, sorting and filtering features were created.

4.3 Results

Clarity ratings.

All-in-all, eleven tasks' ratings fell in the high clarity range (4.5 ≤ clarity), four in medium clarity (4 ≤ clarity < 4.5) and three in low clarity (clarity < 4).

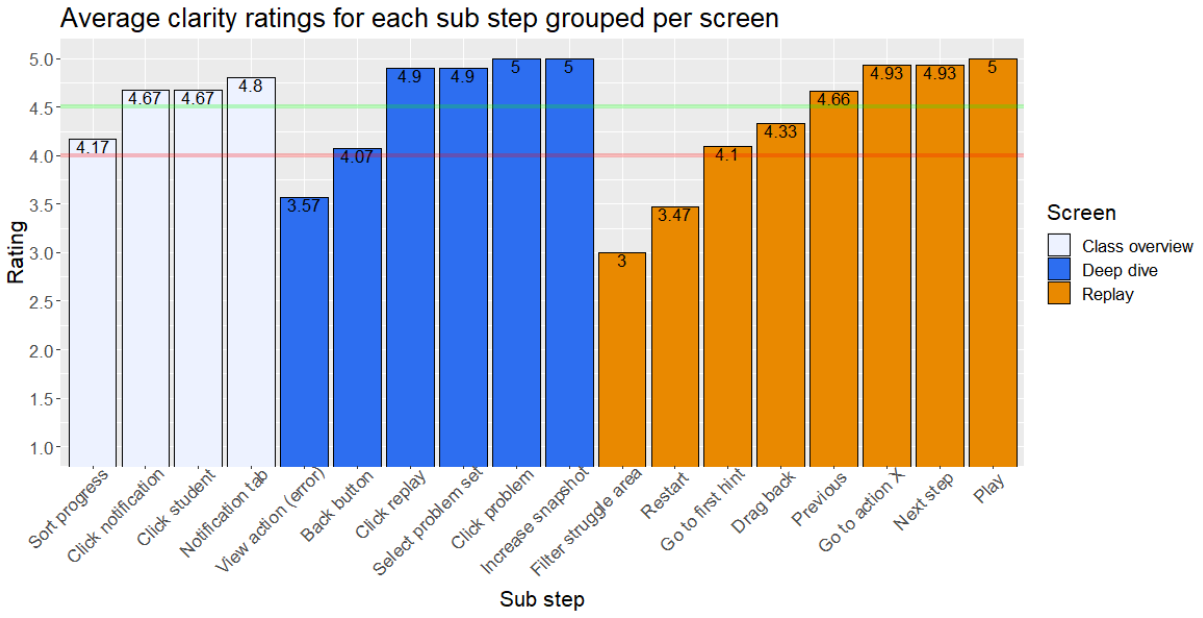


Figure 11. All clarity ratings grouped per screen. 5 indicates high clarity.

In total four tasks were asked in the class overview, six tasks in the deep dive and eight tasks in the replay screen. As seen in Figure 11, most unclear steps came from the replay screen (3 medium, 2 low). Clarity ratings were higher in the deep dive (1 medium, 1 low) and class overview (1 low).

In the replay screen playing the problem, clicking next and previous and going to the fourth action the student performed by clicking on the slider in the problem, were clear. In contrast, dragging the slider was unclear. *drag the slider backwards* (M = 4.33, n = 4) and *drag the slider forwards* (M = 4.23, n = 4) both received medium ratings. *go to the first step that contains a hint* received medium as well (M = 4.1, n = 4). Finally, *restart* received a low clarity rating: M = 3.47, n = 4.

Tasks that were clear in the deep dive were, selecting a problem from the problem list, going to a different problem set, increasing the size of the snapshot and clicking on the replay button. However, *go back to the class overview page* received a medium rating: M = 4.07, n = 5. Furthermore, *filter problems based on area of struggle (e.g. cancel variable terms)* (M = 3, n = 2) and *check out one of the attempts (e.g. click an error)* (M = 3.57, n = 3) got a low clarity rating.

In the class overview (i.e. the overview table and notification panel) most tasks were clear. In particular, switching between all and current notifications, clicking on a student and on a notification. However, the *Sort the progress* (M = 4.17, n = 4) task received a medium rating.

4.4 Discussion

Clarity ratings.

Based on the clarity analysis, most usability issues lie in the replay screen. Therefore, attention should primarily be focused on replay. The big buttons on the replay bar were clear.

Moreover, clicking on the slider to go to a step was evident as well. However, three points were unclear.

Replay. First, the dragging functionality was moderately unclear. People are not forced to use dragging if they do not want to. Notably, due to technical limitations, dragging backwards causes a lag in the system, which sometimes confuses teachers. We asked two teachers specifically about this and they responded that they liked clicking over dragging and found that more intuitive to do. However, to make the functionality clearer the dragging button and action indicators are increased in size when hovering over them.

Second, going to a hint step was not always immediately clear. Some teachers indicated they were not sure what the indicators meant until clicking around. To make this clear from the beginning a legend could be added. However, one has to make sure that this legend does not overlap with the problem components itself and contain a close button. The screen should display an info icon when it is shrunken.

Third, the restart button was very small, which made it less noticeable for teachers. Therefore, we decided to make the restart button appear in place of the play button when replay ends. This adaption helped make this button more noticeable. One has to consider whether sacrificing space for a big restart button is worth the space that it might take up from the display of the student's work in the tutor. Therefore, we decided to move it to the top of the bar instead of increasing its size, where its white color contrasts with blue, making it stand out more.

Deep dive. Navigating problems and problem sets was clear. However, some teachers mentioned that they did not immediately comprehend that the black boxes indicated problem sets. This could potentially be remedied by adding the problem set boxes at the top of the problem list where they are closer to the problems, which they manipulate. However, when a teacher creates an assignment with many problem sets, it would cause wrapping of the

problem set boxes, thus covering multiple lines on screen. Moreover, a different placement of the previous and next problem set buttons needs to be implemented in that case. By looking at different screen sizes and coming up with responsive solutions this can be fixed. Another point relatively unclear in navigating was: navigating back to the dashboard page. Therefore, it was decided to make this into a button instead of an underlined word.

The buttons above the snapshot were clear. That is, clicking the full screen and replay button received a high clarity rating. However, checking out one of the attempts by hovering over the indicators in the snapshot was less clear. Some instructions should be added as to how the indicators of errors and hints work. An option would be to add a legend. However, that takes up space. The best option might be to show a legend and short sentence explaining hovering the first time a teacher logs in. While this is displayed an error indicator could be highlighted to make the teacher aware where they need to hover or click. Moreover, an info icon could be added if a teacher forgets afterwards what they should do.

Finally, filtering problems based on areas of struggle was unclear. The main unclarity here was that nothing seems to happen for a teacher when they press the button[2]. This could be remedied by adding small chips, such as in Figure 12, explaining the filters just above the problems. Moreover, animating problems out, similar to the animation on notifications, might emphasize attention towards what changed.

Figure 12. a. A filter chip (left) and b. an option for sorting the table (right). When the user filters for a skill, chips like these could be an indication of which filters are active.

Class overview. In the class overview most tasks were clear. In particular, switching between all and current notifications, clicking on a student and on a notification. However, the

[2] When all problems contain the skill in question, nothing changes when someone filters for that skill. Namely, no problems are filtered out by that skill.

sorting task received a medium rating. The choice of remediation here is to make the sorting arrows encapsulated in a small button, to make them stand out more as buttons (see Figure 12).

Improving notifications.

Most teachers indicated that notifications were clear and useful (n = 4). However, three teachers indicated the need for sorting and/or filtering of the notifications. Moreover, collapsing the notifications of one student together would make it easier to separate different students' concerns (n = 1).

Sorting and filtering. The majority (n = 3) of teachers mentioned they would like some way to sort, filter and group students. Alphabetical sorting came up, since that teacher had their class arranged alphabetically and alphabetically sorting would make it easier to group students. Moreover, grading them would be simpler. Specifically, a teacher mentioned that they would make similar marks to the notifications in their gradebook, which is alphabetical.

Teachers discussed three other ways of sorting and filtering: by problem (i.e. progress), notification type, and skills. Two teachers deemed skills and notification types to be the most important, since they allow teachers "to look for the question that most people struggle the most on." Using a combination of sorting for progress and filtering for struggle, a teacher might be able to find out where common struggle points lie, since all struggle notifications on a similar problem would be lumped together. The same would be true for common skill problems when combining skills and progress.

Grouping into common areas of concern (e.g. two students with similar struggles) and gamification were concepts that teachers discussed. Grouping into common areas would help the teacher make decisions on what to let students do when they grasp the concept. As one

teacher put it “If most do well, not to dwell on class problems but letting them do enrichment.” TutorShop already contains a feature that implements adaptive individualized problem paths so students that do well on certain skills get less problems for that skill (Holstein et al., 2017).

Positivity. Several teachers suggested a positive feedback button, using more friendly vocabulary for notifications, and a doing well notification. First, teachers wanted to see whether students are doing well (n = 2), since each student deserves (equal) attention. This is covered by the existing doing well detector. The doing well detector should send a notification once a student has performed eight out of their last ten steps correctly. However, the doing well detector does not work due to a bug in the code (see Appendix F). Furthermore, one teacher mentioned they would like to reward these students. Therefore, a reward button could be created, which prompts the student with a message, e.g. “You are doing great, keep going!”. Next, another teacher suggested notifications should be made to be more positive, because negative notifications only give a negative view of how the class is doing. This teacher was concerned that showing the dashboard to the students would be met with backlash from students as well as parents. However, as with the previous storyboard study, teachers (n = 2) indicated that when students are monitored and afraid of getting caught, they are more likely to work hard.

How would teachers use the tool and integrate it in class?

Teachers looked at problem sets and how far students got there (n = 2). Moreover, scrolling through the list and seeing how many have started in the last worked column would help teachers (n = 2). What is more, one teacher preferred scrolling over sorting. In addition, a glanceable visual representation (i.e. the action summaries) of how students did over all problems in the problem set was viewed as helpful by teachers (see Figure 6). In particular, they liked the colors and used it as a quick overview.

In general teachers felt that they would use the tool during practice time when everything is explained. Moreover, one teacher emphasized that the quality of questions given to students was more important than exploring the features of the software (e.g. mastery tracking and teacher notifications). Thus, much care should be taken to instruct teachers what kind of tutors there are, since there are many (see Aleven & McLaren, 2021). Moreover, it would be an interesting idea to give teachers an example exercise for each skill that a problem set tests. That way they can adapt instruction based on that. As suggested by a teacher, for each skill we could create an instructional section explaining what the skill does. This also leads to the suggestion of having an example exercise being queried per skill.

Snapshot versus replay. Similar to the storyboarding, teachers suggested they would use both snapshots *and* replay. One teacher said: "I think I will use both: For smart students I would use snapshot more and explain to the student. You can just use the mouse and hover. For average students it is better to show step by step. Their thinking is not so holistic. So, it is better to explain it per step." However, one teacher explicitly mentioned that they would likely only use the snapshot, since it takes less time and is connected to the overview. On the whole the deep dive feature with the replay might be hard to use for teachers, since they do not have the time to get this granular, glancing at student problems during class. As one teacher put it: "Teachers are always balancing a lot in their head. Time, lessons, communicating accurate steps to name a few." We asked teachers about their preferences for the speed at which a problem is replayed. One second per step is too fast according to practically all teachers. Teachers indicated that control over the speed was neither necessary nor preferred. A two second increment will be chosen for the replay. However, as an additional option, the speed could be changed with a small unnoticeable button.

Error analysis. Four teachers out of five mentioned that they would use replay for error analysis with students and sometimes snapshots as well. Three mentioned they would

use it mainly on an individual level, and one specifically on a class level. As teachers noted, sometimes they assign problems to students that contain an error and let the students find the error as a form of learning. Of note is that privacy should be protected for students, so their names should be blurred. Moreover, teachers suggest it would be a good conversation starter to get into student thought process - a recurring concept from both previous stages - by asking them e.g. “what were you thinking here”. Furthermore, “sometimes you don’t know which step came first when looking at a written assignment and you don’t know which step the student did first.”

Using notifications and the tool. A number of teachers stated that, after receiving a notification about a particular student, they would likely walk up to that student. As one teacher said “Notifications should quickly be used to get back into the work again as fast as possible.” Similar to the previous stage, two teachers mentioned they would use the deep dive screen for struggling students, but would handle idle and system misusing students mostly in person. One teacher said they might use the replay tool mostly for struggling students to give them extra scaffolded practice material. The tool is able to scaffold a step by replaying a problem until a struggle step or by showing a hint. Interestingly, one teacher who indicated they would prioritize certain notifications in the storyboarding study, now indicated they would check off notifications based on recency. During storyboarding the teacher assumed notifications would appear side by side per category, instead of as an incoming list. This shows the importance of contextual information that is only gained through prototyping (Davidoff et al., 2007). In particular, showing a visual representation that can be interpreted in multiple ways allows teachers to think differently. Therefore, ambiguous visual representations are good for an ideation phase. Also, this highlights the importance of testing prototypes as well instead of moving from ideas to product without this step, since in that case we might have implemented a prioritization scheme which would not work well. Moreover, it

shows the need to try the notifications out in a simulation or classroom environment, where many more environmental variables play a role.

4.5 Final design

A number of things were changed based on input from teachers. Between stage 3 and 4. Since only a few changes were made after stage 4 the final design and decisions that led to it are outlined below.

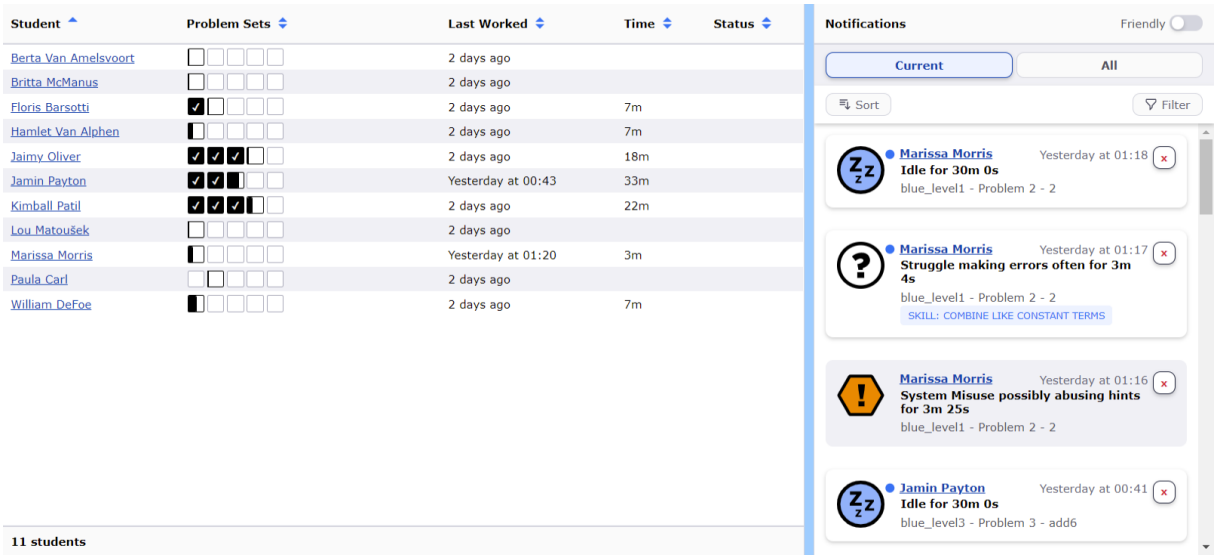


Figure 13. The final class overview. Left: the student table. Problem set blocks indicate proportional student progress. That is, a half-filled block means the student finished half of the problems. Right: Notifications panel. The notifications can be unseen (blue dot) or seen (grey background). Moreover, if a student notification is related to a skill it is displayed in the notification (e.g. second notification). The teacher has the ability to sort and filter notifications (top right). Finally, when they want the notification to show friendly icons (see Figure 14), they can click the switch on the top right.

Class overview. In the class overview, friendly notifications and animations for notifications when sorting, filtering and starting were added. Friendly notifications were added since one teacher specifically noted that displaying the “negative” indicators for students on a screen would never allow teachers to show the system to parents or students (see Figure 14).

Figure 14. Friendly notification images. Each image could come with its own name. Left: finicky fox, a replacement for system misuse. Left-middle: Bubble trouble, replacing critical struggle. Right-middle: Sleepy sloth, replacing idle. Right: Bothered bunny, replacing struggle.

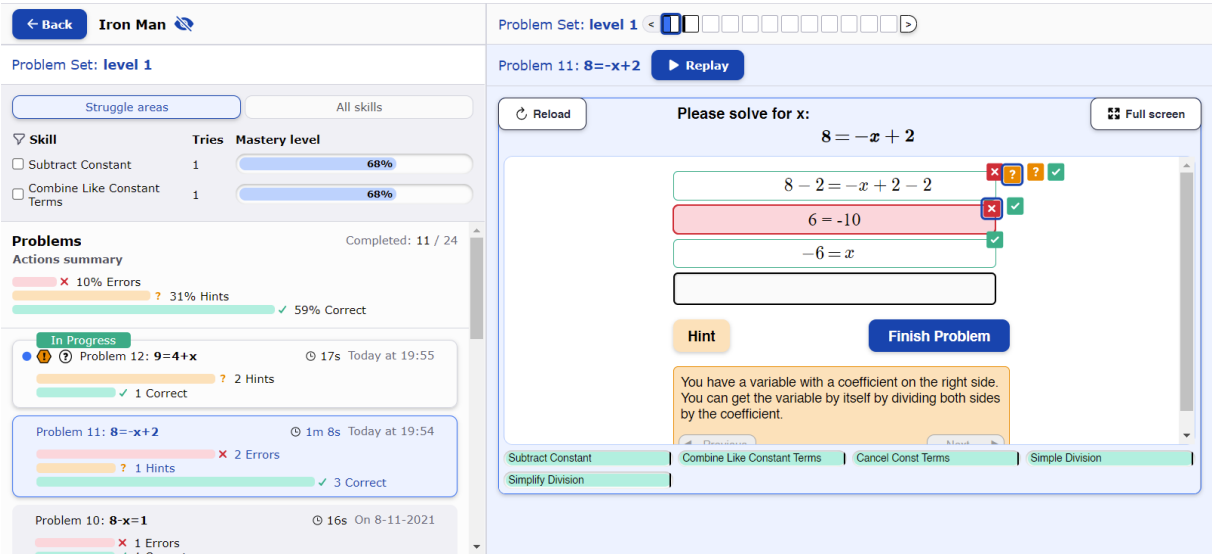


Figure 15. The final deep dive individual student screen. The back button (top-left) has been made more prominent, because it was unclear for teachers during high fidelity prototyping. Furthermore, a button has been introduced to anonymize student names (top-left), since some teachers wanted to show the deep dive to students for problem review. Next, problem action count bars' width is made proportional to the maximum number of actions a student has taken (bottom-left). This makes it easier to see which problem contains more relevant actions than others. It was unclear to teachers why the snapshot did not have the same number of actions as the problem cards indicate. To make this clearer, when new actions come in for a problem a blue dot appears to indicate the problem contains new actions. Furthermore, a reload button has been added inside the snapshot (top-right), to allow the teacher to easily refresh the snapshot. Finally, the selected problem set is highlighted with a blue outline to make it clearer that it is selected.

Deep dive. We implemented an "anonymize student name" button, placed "attempts" in the central *areas of struggle* column to make it easier to read, and made the back button clearer and more consistent. Furthermore, we moved "in progress" to the top left corner to not

obstruct reading, and made action bars relatively sized to the maximum number of actions of a single type over the entire problem set.

Replay. In the replay screen, the replay bar was shrunk to allow more space for the tutor. In addition, actions are displayed at the nob and hovering over the action indicators makes them grow. Also, hovering over the big next, previous and play buttons shows their text with animated icons. Finally, the restart button was moved to the top bar and the bar is now slightly transparent so when it overlays the tutor you can see through it.

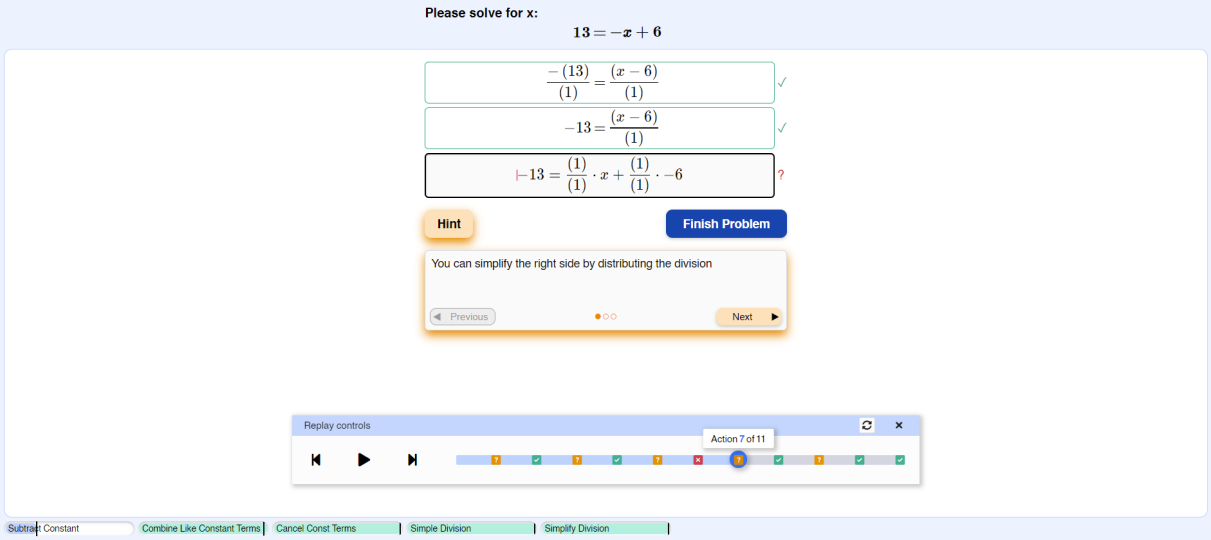


Figure 16. The final replay controls design. The main design changes in the replay controls include the action count that is displayed above the knob to more easily view which action the teacher is on. Furthermore, the restart button has been moved to the top bar. In this location it contrasts more with the background (white on blue), so it should be easier to find. Finally, the previous, play and next button do not display the words anymore. Only when hovering do they appear. The controls were deemed intuitive, thus we chose to remove them, since less redundant information is less distracting.

5. Stage 4) Deliver: pilot studies with Digital Enactments.

In the next stage, we performed, what we call, Digital Enactments with high-fidelity prototypes. We aimed to validate the value of our design solutions in a simulated real-life setting. Teachers re-enacted different scenarios that could happen in-class while using the system as similarly to "real-life use" as possible. Real student data was replayed for the user. Specifically, data from another study with the system, a Lynnette study using 6th, 7th, and 8th grade students (Nagashima et al., 2021). Digital Enactments were inspired by Replay Enactments (Holstein et al., 2019). However, we did not run the sessions in a classroom or computer lab, with one student's replay per seat. We performed our sessions online. Furthermore, we did not ask teachers to engage in role play - acting out what they would do and how they would help students. We asked them what they would do in different scenarios, but did not let them fully act out the point where they would have walked up to a student.

5.1 Research questions.

The data from previous stages was used to set up digital enactments with. Two research questions guided the study. The first is *What is the influence of notifications and student progress data on teachers' choice for snapshot, replay or interaction with students?* The second *How can we make the teacher dashboard better suited for in class use?*

To test the influence of the class overview (i.e. notifications and progress data) on teacher decisions, six different expectations were investigated. In short, idle, system misuse and doing well notifications, and quick progress were expected to mostly lead to quick in-person interactions. While struggle and slow progress were expected to lead to more thorough investigation using snapshot or replay.

Influence of notifications and progress data on teacher decisions

There were six different expectations to what features of the interface would lead to what interactions from teacher to students. These expectations are outlined in Table 2.

Table 2.

Hypotheses table: Influence of notifications and progress data on teacher decisions

	Type of interaction					
Elicitation type ↓	Do nothing	Interact short	Interact long	Problems	Snapshot	Replay
Struggle						
Slow progress						
System misuse						
Idle						
Fast progress						
Doing well						

Note. Each scenario will be graded on a scale of 4 (1: very unlikely, 2: somewhat unlikely, 3: somewhat likely, 4: very likely). The orange squares indicate the expected preferences of teachers.

The first three expectations are that idling, system misusing, and doing well notifications will mostly be handled via quick interactions with students. First off, we expected idle notifications to lead to quick interactions with students, but not to an in-depth view of snapshots or replay, given that during storyboarding teachers indicated that idle students need a "wake up call". Moreover, during high-fidelity prototyping, some teachers specifically mentioned they would not investigate idle students but call them out. In a similar vein, we anticipated system misuse notifications to mostly lead to quick interactions with students, but sometimes to the deep dive problem overview. During storyboarding some teachers expressed interest in viewing whether such notifications were legit. That is, they wanted to see what types of hints and on which exercises students were using hints. Additionally, we assumed *doing well* notifications and fast progress lead teachers to have positive quick interactions with students.

The last two expectations are that low progress and struggle notifications lead more often to detailed investigation. Specifically, we expected that low progress sometimes leads to investigating the problems overview and snapshot (i.e. the deep dive). The reason for this hypothesis is that during high fidelity prototyping teachers indicated that a student who is progressing slowly might need help. Furthermore, in both stage 2 and 3 teachers indicated they would like to see step by step details for students who have difficulties. Struggle notifications and areas of struggle are expected to lead to investigation and long interactions with the student including problem review using snapshot or replay.

How can we make the teacher dashboard better suited for in class use?

Three hypotheses were developed to answer this question. Our first hypothesis is that teachers want to have live views of student problems. In stage 2, during storyboarding, a live view option of the student interface was explored as potentially interesting for teachers. Even though teachers indicated they might not have time for live views and might not need them in class, seeing the interface in action might change their mind. Moreover, one teacher indicated she would want a live interface.

Second, we expect teachers to want an option to give (positive) feedback in the tool. Giving quick feedback came up in both storyboarding as well as during prototyping, and specifically positive feedback was requested. Furthermore, during storyboarding, teachers indicated a need for quick chats.

Third, we assume that teachers want to have the option to query a problem that involves similar skills to give to either the student or the class. This feature was developed using the results from storyboarding and prototyping. Teachers indicated a need for reviewing problems with students, showing a problem to the entire class and finding out what skills are involved in certain problems. Therefore, we assumed they want to have this feature in the dashboard as well.

5.2 Methods

Participants and design.

Three female K-12 math teachers participated (M_{age} = 53, SD_{age} = 14.4), all of whom had taken part in one or more of the previous studies. Two teachers were teaching in the United States, and one in Croatia. On average teachers had 18.67 years of teaching experience ($SD_{experience}$ = 18.58, *range:* 6-40). One teacher was teaching 8th-10th grade, one 7th-12th grade, and one 9th-11th grade. The United States 9th-11th grade teacher was interviewed and showed a particular interest in replay during most scenarios. However, due to time constraints her analysis is excluded from this thesis.

Materials.

During this study three things were needed: a tool to replay student data, data to replay, and a computer(s) that can handle the load of replaying multiple students at once. The tool to replay student data was created by the CTAT+Tutorshop staff to support this study. The tool works by reading in a dataset of transactions. We used DataShop (Koedinger et al., 2010), which records all the actions a student performs in the tutor. Since Nagashima et al. (2021) performed a study using the algebraic equation solving tutor Lynnette, their data was used to replay for teachers. The data used came from a 6th grade class of 11 students. It contained problems of increasing difficulty (Xhakaj et al., 2017). In particular, there were five different problem sets, each containing three or four problems (18 total). Overall, the students performed 2160 steps of which 359 were incorrect, 533 correct and 1232 were hints. Since data was gathered in short time increments, data was concatenated to have as much data per student as possible. Time intervals between transactions were never increased.

Procedure.

First, we start by explaining the goals of the study. The replay of all 11 students' data

was started at the beginning of the sessions with teachers. That is, finding out how teachers would make decisions given the current system and how to make it more suited for in class use. Then, we show the teacher the features of the dashboard and focus on how all the features of snapshot and replay work.

Next, we ask them two open questions: “What would you do with this tool, given that you were in class with a group of 11 students?” and “Is there a certain order in which you would handle the notifications?” After this introduction, the teacher is let to their own accord to interact with the dashboard. In between the teacher was ask specifically about how they handle each type of notification and why. We categorized each answer into the interaction categories shown in Table 2. For each category they were asked the likelihood that they would handle in the ways they describe from on a four-point Likert scale (1: very unlikely, 2: somewhat unlikely, 3: somewhat likely, 4: very likely). Furthermore, in each of the six scenarios (four detectors, two progress types) the teachers will be asked to go to the deep dive screen and look at it. Also, they were asked to investigate the snapshot as well as the replay. This makes sure that they know what information it has to offer and potentially brings insights into their reasons for (not) checking them.

Following this they were asked to fill out a questionnaire about how often they would use the features of the teacher interface (i.e. student progress table, notifications, deep dive screen, snapshot, and replay). For all features the frequency of expected use is asked: “Given students were working in the tutor during class, from the time that you are looking at the teacher software, how often would you look at [feature]” ((almost) never – rarely – sometimes – often – (almost) always) . We assumed it would be too difficult for teachers to predict how often they would use each feature during a classroom session. Therefore, a separate question was asked first “Over the entire time that students are working in the software, how much time do you think you will spend interacting with the teacher software?” (never – sometimes –

about half the time – most of the time – always). Finally, the question was asked whether they prefer using snapshot or replay (strongly prefer snapshot - slightly prefer snapshot - no preference - slightly prefer replay - strongly prefer replay). The second part of the questionnaire contained questions about whether teachers want to have a live view, give feedback in the tool, and retrieve a similar problem (strongly disagree - disagree - neither agree nor disagree - agree - strongly agree). Finally, two open questions were asked: "Would you use this system differently given you were teaching online? If so, how?" and "Are there certain features that you would like the system to have given you were teaching online?"

Analysis.

For the tool's influence on teacher decisions, each of the six display scenarios decisions will be averaged over teachers. Table 2 was used to analyze the agreement between the hypotheses and teacher responses. If there are disparities between teachers and hypotheses, the reasons teachers give will be explained. With regards to the questionnaire questions the Likert scale responses will be converted to numbers, e.g. (almost) never = 1. Then the responses will be averaged over teachers as well.

5.3 Results

All expectations are discussed below in a stepwise manner. First, we discuss the influence of each type of notification (i.e. struggle, idle, doing well and system misuse) and progress type (quick and slow) on teacher decisions results. Next, we discuss whether teachers would use the system for reviewing problems with students. After this, we discuss how we make the teacher dashboard even better suited for in class use.

Struggle

It was expected that struggle notifications would mostly lead to long interactions with students, investigating the problem overview, snapshots, and replay. These expectations were

confirmed. However, short interactions also took place. Furthermore, both teachers seem to have a preference for snapshots and walking up to the student for quick or long interactions.

In case of struggling, in the first two scenarios T14 went to the deep dive via the notification, inspected the snapshot and gave an analysis of what was wrong. T14 read the hints, looked at the errors and told me what the student's problem was. T14 indicated she would like to send a message: "So first I would probably start by only sending her a message. Then if she was really still struggling, I would probably go to her." The reason for this was that initially it is important to not make other students aware of someone's struggle. Saying it out loud could cause more problems. As quoted: "first go with messages and if they still have problems, then I will go to them."

T15's response was similar apart from the messaging, which was not a feature in the current software to begin with. T15 would click on the student and then check quickly in the snapshot what the problem was. Then based on the one she saw there, T15 mentioned giving a quick tip to the student was likely sufficient and that she would do that in person. T15 liked to investigate the snapshot to see what the student does well and is stuck at, e.g. the right side of the equation. If the last part of the problem was wrong then T15 would address it. In this comment it shone through that T15 wanted to correct errors as soon as possible (similar to Lawrence et al., 2021) since she wanted to correct the student "from the very beginning".

These initial responses were interesting, since both teachers automatically started investigating the snapshot immediately. Moreover, they were able to figure out the problems the students were having. However, when asked specifically to investigate the replay as a comparison, different results emerged.

T14 remarked that in the case of struggle she would also use replay, because it seemed easier to see what the students were struggling with. Similarly, we asked T15 whether she would want to use the replay in class or only after class (since she mentioned she would want

to use it after class as well). Her response was that - on the condition that replay would be accessible quickly (e.g. based on a notification) - she would look at the missteps and then walk over to either suggest something quickly or more lengthily.

Slow progress

We expected long interactions, investigation of problems and the snapshot for slowly progressing students. The first teacher stated she would mostly have long, in-person interactions. In contrast, the second teacher would do all the expected things, but mainly focus on replay.

T14 stated that if someone is slow and idle, she would walk up to the student. If the problem is small, it is a short talk, but most likely a longer conversation. Moreover, her expectation is that someone is struggling in this case. Since students can actually work on paper in class and some of them would rather do that, T14 indicated that it is useful to quickly check up on these students in person and see what they are doing.

T15's response to the first scenario was to check the snapshot for the problem the student was at. In this case the student had many hints and errors. Therefore, T15 wanted to see the replay. However, from the snapshot, it was also clear that everything was "just numbers". T15 mentioned "looking at the replay control really helps", and compared it to "asking: do you know now how you got $x + 2$?" Now the teacher knew what to ask and would go up to the student. If the student could quickly answer, then the struggle and time was helpful. "But if he makes the same mistakes, again, this is something he's struggling with." In general, T15 commented that replay at this speed worked well "because this is what I see students do on paper. I let them make the mistakes in front of me when I'm tutoring them" mainly to see students' thinking process.

System misuse

We expected system misuse to lead to short interactions, problem overview and snapshots. However, replay and long interaction were often preferred in this case, since system misuse was judged by teachers to be similar to struggling.

In T14's initial response she investigated the snapshot again. To T14, however, the hints are not that important and do not bother T14 much. Some students are only interested to know what the hint is. Therefore, T14 was less concerned about hint use, but more interested in mistakes. T14 seemed to prefer snapshot usage and did not go to replay. When directly asked, her response for system misusers was to immediately go to students to see what the problem is. This is the way she usually handles things in class, no matter the problem. Notably, in this case she would not send messages. For system misusers T14 said it was most important to talk to them so students know "that I can see what they're doing." While, for struggling students T14 thinks it's better to send a message, because then they know there's help if they want or need it, but for now they can go on with their business.

T15's response was notably different from T14. In the system misuse case T15 would want to see the replay. This is especially the case if it seems like a lot of errors have been made. In addition, T15 subtly mentioned here that she wanted to walk over while watching this. In the case where so many hints were being used the teacher said she wanted to do the entire problem again from the beginning. Moreover, there was an indication that the replay would be used as a problem review tool here.

Idle

For idle notifications the main expectation was confirmed, that is teachers would mostly have short in-person interactions with these students. Although, if the snapshot is quick to use, they would check that as well.

T14's initial reaction was to click on the notification and investigate the snapshot. Next, she wanted to message the student, indicating that they should go to work. Otherwise, she would go up to the student to tell them.

T15's response was more extensive. Initially she wanted to walk up to the students, because she thought: either they are having login problems or possibly they are struggling. However, it also seems like T15 would go to the snapshot to quickly see what is going on and tell her to try some hints (this was a student that made many errors and did not yet use hints). In general, she would walk up to the student. She said a snapshot might be useful, but only if it is quick to use and can easily be used to troubleshoot the problem. In this case the example is given of using the hint button, which is easy to spot due to the lack of hint indicators. T15 quickly scanned that there were many errors in the current problem and immediately suggested inspecting the hints.

Doing well and fast progress

Similarly, for students who are doing well and making fast progress, we hypothesized teachers would quickly interact with students, which was confirmed.

T14 mentioned that in the case of fast progress and doing well she would send a short message or praise them in-person, "[b]ecause [students] only like to be visible when they're doing good." T15 would just walk up to the student and do a quick interaction. Namely, she wants to give attention to everyone. T15 did look at the snapshot. She saw the student was fast but stuck on one problem. Sometimes if a student "is able, if he's doing well, and maybe he just wants to solve on his own." So sometimes the teacher might check the snapshot and not interact. Most often quick interactions seem to be the preferred method of handling these students.

Problem review

Teachers were asked whether they would use the replay or snapshot as a tool to review problems with students, since some teachers indicated they would during high-fidelity prototyping.

T14 would likely not use problem review with all students, since when students are on screen for the entire class, she thinks it is better to have more variety. In addition, due to the COVID pandemic and studying from home, T14 mentioned that these days, after the remote learning sessions, students indicate they prefer real-life one-on-one communication and old-fashioned explanations to on screen explanations. Moreover, presentations are less popular now as well. However, for some students "who like that you show such a thing on screen", T14 might use replay as a problem review tool.

In contrast, T15 was saying she would use it as a review tool, since technology can make math more fun and engaging. She said that schools are often not the first to get new technology, while students in different facets of their life use it all the time. To entice students for math at an early age is essential for enjoying it later in life. This type of technology helps for that, because "kids like learning online". As previously mentioned in the system misuse case, T15 might use replay for reviewing problems with students. She would prefer to have a tablet, however.

Teachers would frequently use all parts of the teachers software.

In general, the questions about whether teachers would use the teacher tool were answered positively. Both teachers indicated they would use the teacher software 'most of the time' or 'always'. One teacher indicated for each specific feature's (e.g. snapshot) *frequency of use* question that they would use it 'often' (4), while the other always indicated '(almost) always' (5). Notably there was no variation in this. There was a noticeable preference for

replay over snapshot, with one teacher strongly preferring and the other somewhat preferring replay.

Extra features.

Teachers were asked whether there were additional features, not part of the current tool, they wanted and whether they saw general improvements. The teachers were asked specifically whether they wanted a live view, would like to give feedback in the software, and whether they would want to retrieve similar problems as ones struggled on. All features were positively evaluated by teachers. Giving feedback and retrieving similar problems were 'strongly agreed' to by both teachers. As previously mentioned, giving feedback in the software already came up during the interview with one teacher. The least popular option was a live view. One teacher 'somewhat agreed' to wanting a live view, while the other 'neither agreed nor disagreed'.

There were other improvements that teachers noted as well. First, a grid overview with all students, where notifications would pop up. The way the roster is lined up as a table view might not be the clearest layout. A grid layout, similar to viewing a class from up top might work more easily as one teacher noted. Her second idea was notepads for students. She noted that not every detail might be visible in replay. Students often jot things down with pen and paper. If the tool would allow drawing or making notes, which could also be replayed, the teacher would be able to see everything the student does.

Using the tool in a remote classroom

Since previous parts of this thesis investigated remote classroom use, this topic was inquired about here as well. T14 specifically remarked that nothing had to be different and that teachers would learn to handle notifications faster. T15 remarked that at home it is also helpful to have this software for quarantined students, since nobody knows how long these quarantine sessions will hold. But when teachers have access to tools like this, she can still

see what students are doing, and students have help from the system. However, during remote learning she saw “how that feature will actually help teachers, because it is so hard to monitor what your students are doing”. Moreover, during remote learning T15 had a similar preference for sending messages to students to “individually tell them what to do or something to suggest to them. Instead of saying, out loud for everyone to know”

5.4 Discussion

All-in-all, teachers were positive about using the provided teacher tools. Our hypotheses predicted modest use but teachers were more inclined to use the software than expected. It was expected that struggle notifications would mostly lead to long interactions with students, investigating the problem overview, snapshots, and replay. This was confirmed. Surprisingly, system misuse and slow progress were met by similar responses by the teachers. In particular, system misuse was interpreted by teachers as struggling. Therefore, it is logical that they would reply similarly. Regarding idle, doing and fast progress, quick interactions were expected by us as well as teachers.

A live view is not so popular, while replay is preferred over snapshot

Similar to the conclusion that followed from the storyboards, a live view is not the most requested feature by teachers. One teacher mentioned that this feature should be on a different monitor, since it would be too distracting. Moreover, this teacher had concerns about Wi-Fi usage, because their school has limited bandwidth. However, she said “When I was able to see what you told me, like I had the dashboard, and then I couldn't get the notifications. And then I could look at the replays. Like, that's enough for me.”

Problem review and other use cases

One teacher was enthusiastic about using the tool to review problems with students, while the other would only use it for specific students that like technology. Apart from this use-case three other scenarios came up: using the tool for getting to know students, helping

students asynchronously (i.e. using the tool for homework), and for making parents understand grading.

The first, getting to know students, was explained as follows. In the first quarter of the year T15 does a lot of grading to see where the students are at. T15 was pointing out that this tool would be great for students that the teacher does not know yet. After a while teachers get to know students and what types of missteps they make. Given this tool the process of recognizing missteps would be sped up.

T15's second point is about students who are falling behind. These students would not like to be in class with students from a lower grade level. Moreover, she would not want others to know they are falling behind. Thus, having this tool where only the teacher and student can see what is happening would help. This falls in the realm of privacy concerns. In addition, it is relevant for the COVID fallout, given that students all over the world have been falling behind on their education (Azevedo et al., 2021).

The latter comes back to recording proof of helping, a benefit that teachers mentioned in both storyboarding and high-fidelity prototyping. According to T15, in parent-teacher conferences the tool with replay could be helpful, since some parents do not believe why their child was graded badly. Using this tool, you could show what things students have difficulty with. In short, the teacher would like to say: "this is where your child is struggling, and this is what we can do to work with them."

Replay with real data contains more contextual information

From studies using replay enactments and user enactments it was already clear that contextual information helps prototyping, especially when it concerns a dynamic environment (Holstein et al., 2019). The fidelity at which one wants to enact scenarios depends on the questions you are trying to answer, but should be realistic enough to evoke their own experiences (Odom et al., 2012). This is why Digital Enactments were used to simulate a real-

life classroom with actual student data. The results are interpreted more realistically, even when teachers are viewing replays from classes that are not their own. As T15 said: “This is like so much what I struggled with, with my seventh-grade algebra students. I'm like, this is like them in a nutshell. I'm like, hey, this looks like Marissa. So, I have seen quite a number of these kinds of missteps.” This statement indicates that T15’s students make similar mistakes as what is seen in this data. The steps the students skip is another thing that is often mentioned, while they are important. Even though teachers are positive and indicate they would use the tools often, one should consider that these teachers were interested enough to participate in this type of research. Moreover, in-class teachers might behave differently.

Limitations

Even though the results indicated that teachers would use all parts of the software, it is notable that all teachers gave the same frequency rating to each part. This might have been prevented by giving teachers the question first how often they would have in person interactions with students. This could have indicated to teachers that they needed to compare frequencies of use. Another method to force differentiation would be to let the teachers indicate a quantitative guess (in minutes) as to how much they would use each part of the software compared to in person interactions. For example, “given your students were working on the software for 50 minutes, how much time would you spend on replay, snapshot, in-person interactions, etc.?”

During digital enactments, three issues came to light. That is, which problem set was selected was unclear, it was unclear on what problem set the student was working, and what the working status meant. For T14 it was relatively unclear what the order of the snapshot indicators was. This was solved before the session with T15 in which this was not an issue anymore.

6. Final discussion

In the first stage of the design process, teachers' needs were defined. Teachers need to see what students are doing, get to them quickly, and correct struggles timely (Lawrence et al., 2021). In the second stage, storyboarding, results showed that, in-class, teachers want a replay tool and interactive annotated snapshots. This information was used and adapted to create a high-fidelity prototype for the third stage, during which, teachers often used snapshots, but preferred replay which can be used to analyze errors with students. In the fourth stage we simulated real-time student data in the tool. Here we concluded teachers were planning to use the replay tool more often than snapshot. In short, teachers want to see what students are doing, but are time constrained. Thus, in-class, teachers will likely use the dashboard to review problems with students.

The final step in the design process laid out in this thesis is delivering and evaluating the system. The question remains how well the teacher dashboard works in a real-life classroom context? After this section, a plan is laid out to do a pilot study in class to validate our (see section 7).

6.1 Replay: A versatile tool for teachers

The replay tool was the most preferred component of the dashboard by teachers, in part due to its multiple use cases. Using the replay tool, a teacher can look at a student's history to see their thinking process and quickly understand their struggles. However, replay was most popularly quoted by teachers to be useful for error analysis with students.

Problem review and error analysis.

In recent years, mathematics education has introduced more error analysis practice in curriculums, since it shows positive learning outcomes for students (McLaren, Adams, & Mayer, 2015). For example, Adams et al. (2014) used a web-based tutor to teach decimals to

students. Some students were required to perform error analysis, while others only solved problems. The students that did error analyses performed better on a posttest. Thus, using the replay tool for error analysis with students shows promise for positive student learning outcomes. Future research should investigate the added learning benefits for students when teachers perform error analyses with ITS replay tools. Since teachers pointed out that the replay tool could also be used to share an erroneous problem with the entire class, this might be the easiest way to test the effectiveness of using the tool for problem review.

Furthermore, snapshot-based error analysis and replay based error analysis could be compared using A/B testing (Sharp, 2019). Often, two designs are pitted against one another using A/B testing, a data-driven approach to measure which design choices are better (Kohavi & Longbotham, 2015; Martinez-Maldonado et al., 2015). Here statistics could be gathered to more conclusively answer which display mode works best.

Comparing snapshots, replay and live.

Snapshots and replay allow teachers to see student thinking. When working with pen and paper students can have unreadable handwriting or erase their erroneous work, thus erasing their thinking process. The main benefit of snapshots and replay compared to a live view is that the former two allow a history check.

Replay is preferred, but snapshots are useful. From storyboarding teachers concluded a replay contained more relevant information. However, snapshots are perceived as quicker to use. Moreover, snapshots allow easier pattern identification over multiple problems. Teachers' first instinct during digital enactments was to use snapshots to identify and investigate patterns over one or more problems. Nevertheless, teachers indicated a marked preference for replay over snapshots. They mentioned replay could be used for getting to know students more quickly and explaining to students, parents and others what the underlying problems are and how to help students further. Furthermore, replaying felt as a

more natural way to view a student's solution process. As one teacher put it: "I like the replay better. I feel like the replay shows me the student in action, and that's just how my brain works. I like to see things in motion."

Why prefer replay over a live view? Replay might fit better for teachers than a live view. In-class, during storyboarding as well as digital enactments, a live view was not preferred. In contrast, for Lumilo - the mixed reality glasses orchestration tool for teachers - a live view was popularly requested (Holstein et al., 2019). Lumilo also implements areas of struggle, notifications, and a class overview. Why is a live view sought-after for a mixed reality tool but not for a display-based tool? One explanation could be that the controls for a mixed reality tool are less well suited for things like replay. Of course, a teacher could be given a controller. Some mixed reality tools even use gestures for controlling the interface. However, people are less familiar with mixed-reality tools, than a pc, laptop or tablet. This comes with challenges for the interface designer.

With replay teachers can view problem solution history, while a live view only allows the teacher to watch the student in real-time. Viewing problems in real-time is bound to keep the teacher waiting for a point of struggle while teachers ask a student to solve a problem, similar to how they would in-class, as indicated during all stages of the thesis. Therefore, replaying a previous problem, or the current problem up until the point the student solved it, helps a teacher see all the relevant information at a glance and move quickly through steps that are less relevant to view. Furthermore, they can analyze multiple problems using replay and find points of struggle that persist over time. Also, replay allows teachers to review problems with students, which live does not. Replay does not exclude a live view. One could create a live view at the end of a replay. When going to the final step of the student problem solving process a live view could start, for example.

6.2 Recommendations based on teacher studies

For both in-class and remote teaching we recommend implementing a way to give feedback to students as well as the option to query a similar problem as one that is currently viewed. There were many other ideas that teachers came up with, but we could not explore them all (see Appendix D). Follow up research should consider which ideas to pursue and which not to. To inform the reader about some of the ideas that came up, an informal effort versus impact graph, based on ideas from teachers is shown in Figure 17. Effort suggests the amount of work it takes to implement the idea. Impact was judged based on what would help teachers the most in supporting students. For example, although 'student use of replay' (bottom left) is considered low impact for teachers, it could be helpful for students.

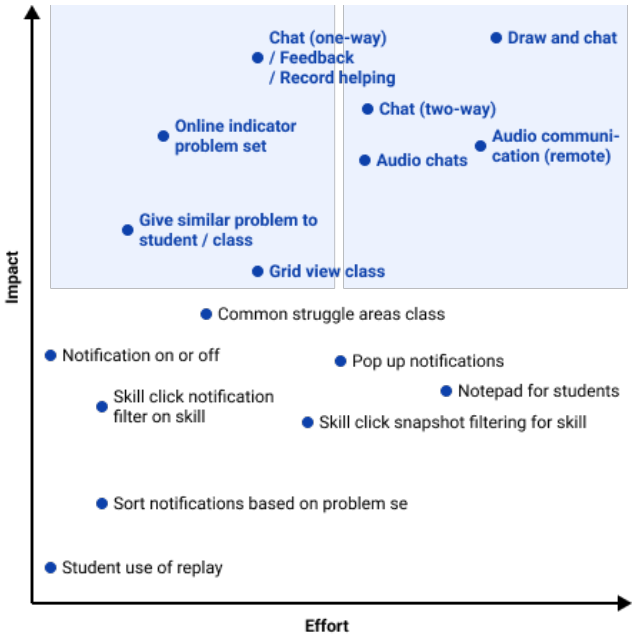


Figure 17. Effort versus impact chart. The chart shows ideas from teachers classified based on the amount of effort they take to implement (x-axis) and the amount of impact they would have on teachers supporting students (y-axis). The blue-shaded boxes could indicate thresholds for deciding whether to implement an idea.

Implement the option to query a similar problem.

We talked to teachers about querying a problem, similar to another problem that a student struggled with. We have worked on how this should be implemented. The envisioned

option is to have a list of all problems and a list of skills used in those problems. The skill the student struggled on is listed, and could be used to query a similar problem. It remains to be answered how similar a problem should be. Should all skills be the same? Should you ask for a slightly easier problem? Do you take into account how well other students did on the problem? These are some of the challenges regarding similarity. Other questions are whether skills should update when a teacher performs a queried problem with the student. A teacher helps the student, so arguably the skill level of the student is not tracked on its own but in combination with a teacher.

Add communication methods to provide feedback (in-class & remote).

For both in-class and remote teaching, teachers said drawing and adding chat balloons was the most appealing method of communication during storyboarding, since it is the most flexible and real-life-like. Chats allow for quick permanent problem-specific feedback and proof of helping a student. While microphone connections are not helpful in class, for remote use they might be helpful.

During storyboarding and digital enactments teachers indicated that they would like to give feedback to students. Our recommendation is to add a simple feedback option in the dashboard. After in class testing, one can evaluate whether drawing would have additional benefits.

Teachers' opinions varied regarding the need for microphone communication in the dashboard. Other options are available (e.g. Zoom), but according to teachers, verbal communication is necessary for remote teaching. Therefore, our recommendation is to first evaluate in class use of a simple text field and button-based feedback option. Then based on the indicated needs from teachers, audio feedback or two-way microphone communication could be added.

6.3 Limitations

Asynchronicity between current student state and teacher view.

It should be noted that there is some asynchronicity between what the student is doing right now, versus what the teacher can see in the dashboard. Neither the snapshot nor the replay updates when a new transaction from the student comes in. For snapshots updating was solved using a refresh button. The teacher is made aware that new steps come in through updating problem steps counts. Snapshot's refresh button should be highlighted when new steps arrive. For the current simulation study, not updating was not a problem. There were no students to attend to. But for in-class use the updating issue should be resolved. Replay should have an indicator that shows that new steps have arrived (e.g. a notification up top). Currently the mechanisms for both snapshot and replay work similarly. This mechanism necessitates that the entire page that displays the student problem should be refreshed. However, it would be better if a new transaction came in through the CTAT library whenever a new transaction was made by the student.

Digital enactments' simulation tool as live feed of simulated students.

Results indicate that teachers might not need a live feed. However, this hypothesis should be more thoroughly tested. Namely, during this research, teachers have not been exposed to a live view. One way to expose teachers to a live view would be to use the Digital Enactments tool to simulate students. This would show a teacher exactly what a live interface would look like. Instead of showing a teacher the snapshot, a live feed could be shown by querying the URL that the student is simulated on. This way teachers could reason more thoroughly about how that feels and whether they would prefer this in class.

Responsiveness

Designing for different device types is important. During high fidelity prototyping, one teacher explicitly mentioned they wanted to use an iPad to use the tool in class, since it is

more portable than a laptop. From all teachers in the storyboards three out of eight had a tablet, even though most of them did not use them on a regular basis. Four had laptops (and a desktop) and one a desktop only. One teacher even mentioned they would potentially use their phone. The developed tool does not exclude tablet use and has been tested on tablet screen sizes. However, if the tool is to be used in class with tablets, then the responsiveness should be more critically evaluated.

6.4 Future directions

Future research should prioritize a classroom validation study to support our results (see section 7). Also, research should investigate how remote use of this technology differs from in-class instruction. There are two other directions two pursue: continuing digital enactments with feedback implemented, and investigating the use of a dashboard tool for homework.

Follow up digital enactments with feedback

The types of feedback teachers will give can be further investigated by designing the dashboard in such a way that teachers can click buttons to add different types of feedback to a student's work. For example, four types of interactions can be preprogrammed: giving praise, giving a hint, telling a student to get to work, and asking a student whether they need help. The feedback that teachers give could then be compared to what the literature considers the most relevant feedback. Molenaar & Knoop-van Campen (2019) found that teachers who consulted a dashboard more often gave more different types of feedback. Also, Sedrakyan et al. (2020) showed that system misuse notifications led to more behaviorally oriented feedback (e.g. classroom conduct). In contrast, struggle led to more process-oriented feedback (i.e. cognitive step-by-step feedback). During digital enactments teachers indicated that system misuse was interpreted as struggle. If teachers therefore respond in a more process-oriented fashion, student learning outcomes might improve.

Asynchronously using the tools (for homework).

During storyboarding, high-fidelity prototyping and digital enactments, there were teachers that expressed interest in using tutors for homework. This thesis decided to focus on the in-class use case, since requirements for homework use are likely different from in-class use. However, teachers would also be able to view the dashboard after class. A clear difference would be that idle notifications would have less relevance when students use the tool for homework. Moreover, a live view becomes more redundant, since we should not expect teachers to view students working in the evenings on their homework (although they did indicate they would probably do that sometimes). Future research should investigate the requirements for using the dashboard for homework.

6.5 Conclusion

This thesis is among the first teacher-centered design research to use data from an ITS to support teachers helping students in class using a dashboard. Before this study there has been little empirical back-up to help teachers regulate student learning (Sedrakyan et al., 2020), especially in the realm of ITSs. Meanwhile ITSs (Hillmayr et al., 2020; Du Boulay, 2016; Kulik & Fletcher, 2016) and teacher-student interactions are shown to improve learning outcomes (Lee, 2020). Furthermore, there has been a shift in education worldwide towards *one-to-one* education, that is, one computer for each student (Islam & Grönlund, 2016). Given the improved learning outcomes, it seems likely that computers and ITS will become more prevalent in education. Therefore, teachers should have tools that help them improve teacher-student interactions.

We presented two novel ways of displaying student work to teachers: replay and snapshot. Interestingly, replay has multiple use cases. Primarily teachers expect to use it as a problem review tool with students. Also, it can be used for grading and explaining student data, getting to know students quicker, and for homework. Other than these novel methods, it

was also the first to make recommendations on how to use these types of tools for remote classroom instruction. In addition, we presented a unique framework to categorize teachers' decisions based on data from ITSs. Teachers indicated our tool would work well to support them in helping students working in ITSs. To validate these results a classroom validation study proposal is laid out in the next section.

7. Stage 5) Deliver: Classroom study.

The final stage of the LATUX workflow is validation in the classroom. In this stage, pilot studies should be held to validate the value of the design solutions in a real-life setting. This process was outside of the scope of this thesis. We finalized the set up and scheduled two classroom sessions for the software with a teacher. But the sessions were cancelled. Below the setup for an in-class session is explained.

7.1 Research questions

The following research questions are central to this part of the study: *How do teachers use the dashboard in a real-life classroom context?* and *How do replay and snapshots compare in terms of their use?* This leads to the hypotheses described in Table 3.

7.2 Methods

Participants.

To be determined.

Materials.

The finalized product was to be delivered to the classroom including two tutors requested by the teacher. These tutors are picked to align with current student goals, thus 6th grade math level. Any MathTutor from the MathTutor site (Aleven & McLaren, 2021) can be used. We were going to use a laptop aimed at the class, so that the researcher could view the class while the teacher was teaching. Each student in the class should have their own laptop or tablet. The teacher was going to view the dashboard on a desktop computer which was facing in her direction. Their laptop should broadcast a video feed of the class via Zoom, so the researcher can interact with them as well as observe. This feed should not be recorded, due to student informed consent constraints. Namely, getting informed consent from all students to

videotape them was not worth the effort and risk of non-participation, given that our study focuses on teachers.

Table 3

Hypotheses table in class

	Way of testing		
Hypothesis ↓	Tallying	Post session interview	Log data
Teachers use replay to review problems together with struggling students	X		
The dashboard often leads to interactions with students	X		
Teachers will use the deep dive (and then walk to students)	X		
Seeing multiple students with similar struggle helps make decisions for entire class		X	
Teachers would rather use the dashboard's replay and snapshot feature after class		X	
Teachers will use areas of struggle to determine what a student needs help with.		X	
Teachers will use replay only for students that are flagged as struggling or abusing the system.			X
Teachers will use snapshots to find patterns in student data			X
Areas of struggle will entice teachers to replay problems that make use of these areas for deeper investigation.			X
Teachers will find replay too time consuming to use in class.			X
Teachers will use replay for longer than snapshots			
Time spent in class overview / student overview / problem review modes will be divided 80% / 10% / 10%.			X

Note. There are three ways in which data will be recorded. First, tallying during in class observation. Second, asking questions after the class has ended. Finally, recording all button presses in log data.

The researcher should have two computers as well, one with a second monitor connected. The separated computer can display the class and could be used to interact with

and explain to the class. The monitor could display the teacher interface via Zoom and record it. Lastly, the connected laptop screen can display a tallier program (see Figure 18), a program created for in class sessions to tally teacher action. That is, whether the teacher interacts shortly (X) or lengthily (O) with a student, whether the interaction originates from something seen in the teacher software (Y) and when the interaction ends (L).

Figure 18. The tallier program. Each rectangle represents a student table. Students are automatically assumed to sit in pairs. The number of tables in class is adjustable. For each table four buttons are included (X: Short talk / pat on shoulder, O: Long talk / explanation, Y: Interaction started from dashboard, L: Teacher left). When one of the buttons is pressed it adds a timestamp on the table. The save button saves the object so it can be read out later and matched up to the timestamps in the teacher software.

Procedure.

A set up for the procedure is as follows. The researcher shortly explains the topics that the tutor covers to the students. Then explain how the software works and how students should login. Next, do an example problem in the tutor while explaining things that can be done inside the tutor.

While the class is working and the teacher is walking around, the number of interactions with each student is tallied with the tallier program (Figure 6). In addition, track whether or not the interaction stemmed from the tutor. Finally try to gauge whether an interaction stemmed from a notification or other information in the dashboard.

In session observation. During the session different variables are tallied. A short talk, like a pat on the shoulder or a single sentence is marked using an *X* on the classroom legend, while a longer explanation is marked with an *O*. If a teacher is talking to multiple students both students are marked. When a teacher leaves, mark the table with an *L*. Finally, the number of transitions from tool to student is noted. That is, if the teacher went from an individual's deep dive screen to the person or the other way around, it was marked with a *Y*. After the session, time spent watching the dashboard can be calculated by subtracting the time the teacher spent with a student from when the teacher left the student.

Post-session interview. Following the session, the teacher can be asked how everything went. First, ask some global questions: 1) How did it go? 2) What went well / less well? 3) How beneficial was the dashboard for you? 4) What do you think the students thought of this? 5) What information did you get from the system?

Thereafter, ask more specific questions about the replay versus snapshot: 6) How was it to use replay? 6a) What situations would you want to use replay? 6b) What situations would you like to use snapshots? 7) How did you use the system? 7a) What could be better to make it easier for you to help students with this?

Finally, ask questions about the events in class. Try to ask about specific examples, i.e. 8) Was there a time where it was very hectic, 9) Was there a moment that the dashboard came with new information for you that was especially helpful? 10) In what situation did notifications come in especially handy?

Post-session log data analysis. Different things were tracked in the system. Mainly button presses and timestamps. From these the hypotheses about students can be answered. We specifically tracked which notifications were investigated in which path. In addition, log how much time and how frequently snapshots and replay are used. Furthermore, mouse hovering over areas of struggle is logged. Finally, the time that each feature is used is logged.

Analysis.

To be determined.

References

Adams, D. M., McLaren, B. M., Durkin, K., Mayer, R. E., Rittle-Johnson, B., Isotani, S., van Velsen, M. (2014). Using erroneous examples to improve mathematics learning with a web-based tutoring system. *Comput. Hum. Behav. 36*, 401–411

Aleven, V., McLaren, B., Sewall, J., & Koedinger, K. (2009). A New Paradigm for Intelligent Tutoring Systems: Example-Tracing Tutors. *Int. J. Artif. Intell. Educ., 19*, 105-154.

Aleven, V., McLaren, B., Roll, I., & Koedinger, K. (2006). Toward Meta-cognitive Tutoring: A Model of Help Seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence in Education, 16*(2), 101–128.

Aleven, V., & McLaren, B. (2021, October 15). *Mathtutor: a free site where middle school students learn math*. MathTutor. https://mathtutor.web.cmu.edu/

Aleven, V., Xhakaj, F., Holstein, K., & McLaren, B. (2016). Developing a Teacher Dashboard For Use with Intelligent Tutoring Systems. *IWTA@EC-TEL*.

Azevedo, J. P., Hasan, A., Goldemberg, D., Geven, K., & Iqbal, S. A. (2021). Simulating the potential impacts of COVID-19 school closures on schooling and learning outcomes: A set of global estimates. *The World Bank Research Observer, 36*(1), 1-40.

Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction, 4*, 253-278.

Davidoff, S., Lee, M. K., Dey, A. K., & Zimmerman, J. (2007, September). Rapidly exploring application design through speed dating. *In J. Krumm, G.D. Abowd, A. Seneviratne, & Th. Strang (Eds.), UbiComp 2007, Proceedings of the 9th International Conference on Ubiquitous Computing (pp. 429-446). Berlin, Heidelberg: Springer*. Doi: 10.1007/978-3-540-74853-3_25

CTAT Detector Library. (2018) https://github.com/d19fe8/CTAT-detector-plugins/wiki/CTAT-Detector-Library

Design Council, U. K. (2019, September 10). What is the framework for innovation? Design Council's evolved double diamond. Retrieved February 07, 2021, from https://www.designcouncil.org.uk/news-opinion/what-framework-innovation-design-councils-evolved-double-diamond

Du Boulay, B. (2016). Recent Meta-reviews and Meta–analyses of AIED Systems. *Int J Artif Intell Educ, 26*, 536–537. https://doi.org/10.1007/s40593-015-0060-1

Hall, R. R. (2001) Prototyping for usability of new technology. *International Journal of Human-Computer Studies, 55*(4), 485-501. https://doi.org/10.1006/ijhc.2001.0478.

Herodotou, C., Hlosta, M., Boroowa, A., Rienties, B., Zdrahal, Z., & Mangafa, C. (2019). Empowering online teachers through predictive learning analytics. *British Journal of Educational Technology, 50*(6), 3064–3079. https://doi-org.proxy-ub.rug.nl/10.1111/bjet.12853

Hillmayr, D., Ziernwald, L., Reinhold, F., Hofer, S., & Reiss, K. (2020). The potential of digital tools to enhance mathematics and science learning in secondary schools: A context-specific meta-analysis. *Comput. Educ., 153*, 103897.

Holstein, K., McLaren, B. M., & Aleven, V. (2017). Intelligent tutors as teachers' aides: exploring teacher needs for real-time analytics in blended classrooms. *Proceedings of the Seventh International Learning Analytics & Knowledge Conference (LAK '17). Association for Computing Machinery, New York, NY, USA*, 257–266. DOI: https://doi.org/10.1145/3027385.3027451

Holstein, K., McLaren, B. M., & Aleven, V. (2018). Student learning benefits of a mixed-reality teacher awareness tool in AI-enhanced classrooms. In C. P. Rosé, R. Martínez-

Maldonado, H. U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, B. McLaren, & B. du Boulay (Eds.), *Proceedings, 19th International Conference on Artificial Intelligence in Education, AIED 2018* (Part 1, pp. 154–168). Cham, Switzerland: Springer. doi: 10.1007/978-3-319-93843-1_12

Holstein, K., McLaren, B. M., & Aleven, V. (2019). Co-Designing a Real-Time Classroom Orchestration Tool to Support Teacher–AI Complementarity. *Journal of Learning Analytics, 6*(2), 27–52. https://doi.org/10.18608/jla.2019.62.3

Islam, M. S., Grönlund, Å. (2016). An international literature review of 1:1 computing in schools. *J Educ Change 17*, 191–222. https://doi-org.proxy-ub.rug.nl/10.1007/s10833-016-9271-y

Joksimović, S., Poquet, O., Kovanović, V., Dowell, N., Mills, C., Gašević, D., Dawson, S., Graesser, A. C., & Brooks, C. (2018). How do we model learning at scale? A systematic review of research on MOOCs. *Review of Educational Research, 88*(1), 43–86. https://doi-org.proxy-ub.rug.nl/10.3102/0034654317740335

Koedinger, K. R., Baker, R. S. J. d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J. (2010). A Data Repository for the EDM community: The PSLC DataShop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (Eds.) Handbook of Educational Data Mining. Boca Raton, FL: CRC Press.

Kohavi, R., & Longbotham, R. (2015) Unexpected Results in Online Controlled Experiments. *SIGKDD Explorations, 12*(2), 31–35.

Kulik, J. A., & Fletcher, J. D. (2016). Effectiveness of Intelligent Tutoring Systems: A Meta-Analytic Review. *Review of Educational Research, 86*(1), 42–78. https://doi.org/10.3102/0034654315581420

Lawrence, L., Holstein, K., Berman, S. R., Fancsali, S., McLaren, B. M., Ritter, S., & Aleven, V. (2021). Teachers' orchestration needs during the shift to remote learning. *Technology-Enhanced Learning for a Free, Safe, and Sustainable World*. Springer International Publishing

Lee, J. E. (2020). Examining the effects of discussion strategies and learner interactions on performance in online introductory mathematics courses: An application of learning analytics [ProQuest Information & Learning]. In Dissertation Abstracts International Section A: Humanities and Social Sciences (Vol. 81, Issue 5–A).

Long, Y., & Aleven, V. (2017). Educational game and intelligent tutoring system: A classroom study and comparative design analysis. *ACM Transactions on Computer-Human Interaction, 24*(3), 1–27. https://doi.org/10.1145/3057889

Macarini, L. A., Lemos dos Santos, H., Cechinel, C., Ochoa, X., Rodés, V., Pérez Casas, A., Lucas, P. P., Maya, R., Alonso, G. E., & Díaz, P. (2019). Towards the implementation of a countrywide k-12 learning analytics initiative in uruguay. *Interactive Learning Environments*. https://doi-org.proxy-ub.rug.nl/10.1080/10494820.2019.1636082

Martin, B., & Hanington, B. M., (2012). *Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions (digital ed ed.)*. Rockport Publishers.

Martin, F., Sun, T., & Westine, C. D. (2020). A systematic review of research on online teaching and learning from 2009 to 2018. *Computers & Education, 159*. https://doi-org.proxy-ub.rug.nl/10.1016/j.compedu.2020.104009

Mavrikis, M., Geraniou, E., Gutierrez Santos, S., & Poulovassilis, A. (2019). Intelligent analysis and data visualisation for teacher assistance tools: The case of exploratory

learning. *British Journal of Educational Technology, 50(*6), 2920–2942. https://doi-org.proxy-ub.rug.nl/10.1111/bjet.12876

Miro. (2021) *The visual collaboration platform for every team: Miro*. https://miro.com/. (n.d.). Retrieved December 2, 2021, from https://miro.com/.

McLaren, B. M., Adams, D. M., Mayer, R. E. (2015). Delayed learning effects with erroneous examples: a study of learning decimals with a web-based tutor. *Int. J. Artif. Intell. Educ. 25*(4), 520–542

Molenaar, I., Horvers, A., Baker, R. S. (2019). What can moment-by-moment learning curves tell about students' self-regulated learning? *Learning and Instruction,* 101206. doi: 10.1016/j.learninstruc.2019.05.003

Molenaar, I., & Knoop-van Campen, C. A. N. (2019) How Teachers Make Dashboard Information Actionable. *IEEE Transactions on Learning Technologies*, *12* (3), 347-355, doi: 10.1109/TLT.2018.2851585.

Nagashima, T., Bartel, A. N., Yadav, G., Tseng, S., Vest, N. A., Silla, E. M., Alibali, M. W., & Aleven, V. (2021). Using anticipatory diagrammatic self-explanation to support learning and performance in early algebra. In *Proceedings of the Annual Meeting of the International Society of the Learning Sciences (ISLS2021),* Bochum, Germany.

Odom, W., Zimmerman, J., Davidoff, S., Forlizzi, J., Dey, A. K., & Lee, M. K. (2012). A fieldwork of the future with user enactments. *DIS '12: Proceedings of the Designing Interactive Systems Conference,* 338–347. https://doi.org/10.1145/2317956.2318008

Prieto-Alvarez, C. G., Martinez-Maldonado, R., & Anderson, T. (2017). Co-designing in learning analytics: Tools and techniques. In Lodge, J. C. H. & Corrin, L. (Eds.),

Learning analytics in the classroom: translating learning analytics research for teachers. London: Routledge.

Scherer, R., Siddiq, F., & Tondeur, J. (2019). The technology acceptance model (TAM): A meta-analytic structural equation modeling approach to explaining teachers' adoption of digital technology in education. *Computers & Education, 128*, 13–35. https://doi-org.proxy-ub.rug.nl/10.1016/j.compedu.2018.09.009

Schwendimann, B., Rodriguez-Triana, M. J., Vozniuk, A., Prieto, L., Shirvani Boroujeni, M., Holzer, A., Gillet, D., & Dillenbourg, P. (2017). Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Transactions on Learning Technologies, 10*(1), 30-41.

Sedrakyan, G., Malmberg, J., Verbert, K., Järvelä, S., & Kirschner, P. A. (2020). Linking learning behavior analytics and learning science concepts: Designing a learning analytics dashboard for feedback to support learning regulation. *Computers in Human Behavior, 107*. https://doi-org.proxy-ub.rug.nl/10.1016/j.chb.2018.05.004

Sharp, H. (2019). *Interaction design: Beyond human-computer interaction*. In (5th ed., p. 101-134). 10475 Crosspoint Boulevard: John Wiley Sons, Incorporated.

Shim, T. E., & Lee, S. Y. (2020). College students' experience of emergency remote teaching due to COVID-19. Children and Youth Services Review, 119. https://doi-org.proxy-ub.rug.nl/10.1016/j.childyouth.2020.105578

Sutherland, S. M. (2016). Constraint-referenced analytics of algebra learning in classroom network contexts [ProQuest Information & Learning]. In Dissertation Abstracts International Section A: Humanities and Social Sciences (Vol. 76, Issue 7–A(E)).

van Leeuwen, A. (2019). Teachers' perceptions of the usability of learning analytics reports in a flipped university course: when and how does information become actionable

knowledge?. *Education Tech Research Dev 67,* 1043–1064. https://doi.org/10.1007/s11423-018-09639-y

Xhakaj, F. Aleven, V., & Mclaren, B. (2017). Effects of a Teacher Dashboard for an Intelligent Tutoring System on Teacher Knowledge, Lesson Planning, Lessons and Student Learning. *315-329*. 10.1007/978-3-319-66610-5_23.

Appendix

Appendix A. Storyboards

Below the storyboards from the first user study are displayed in the order that they were displayed to teachers.

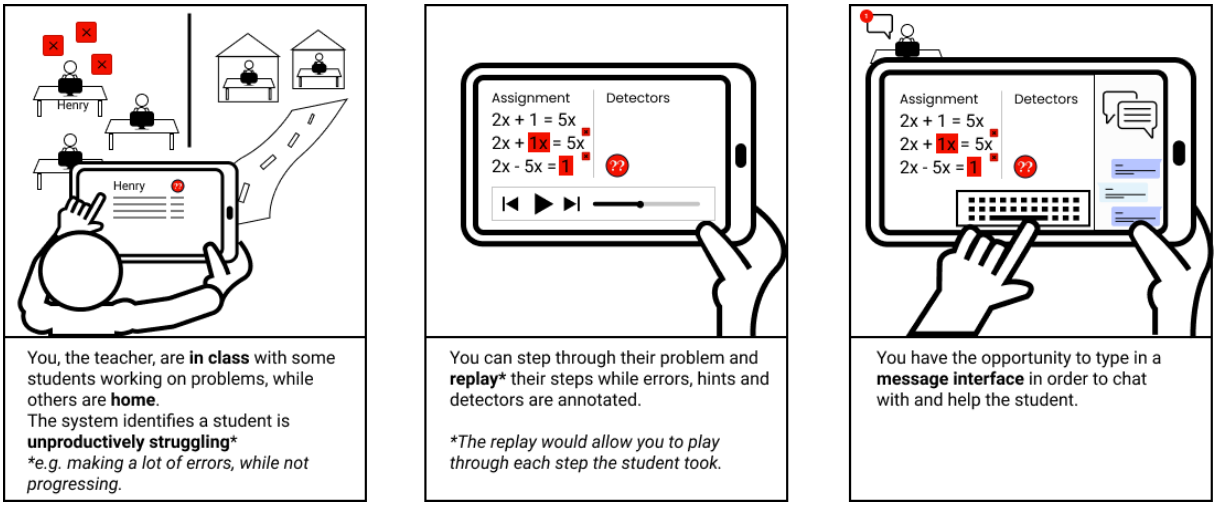


Figure A1. Hybrid, replay and chat.

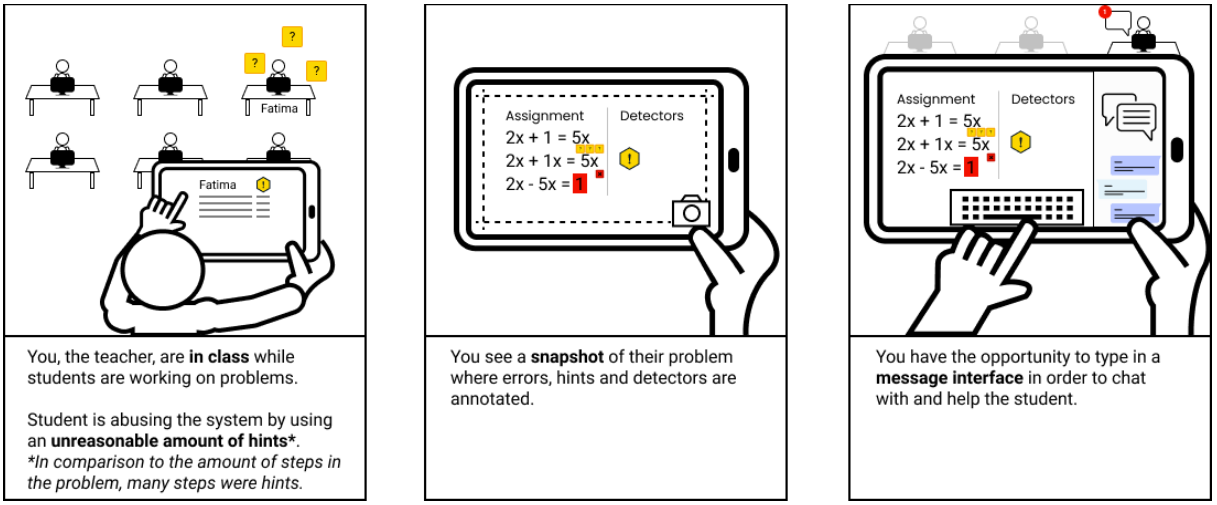


Figure A2. In-class, snapshot and chat.

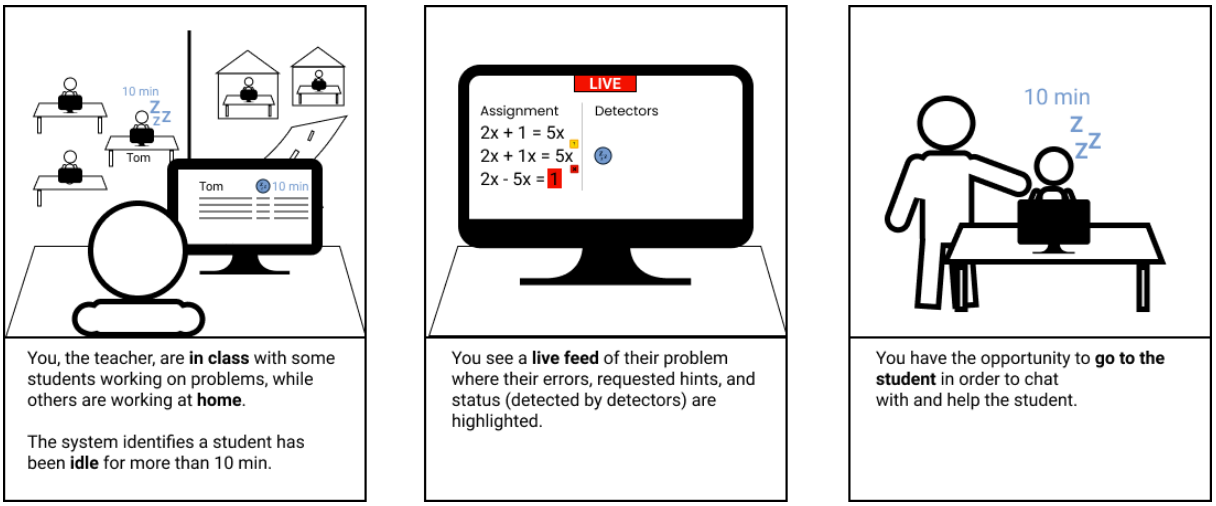


Figure A3. Hybrid, live and walk up to student.

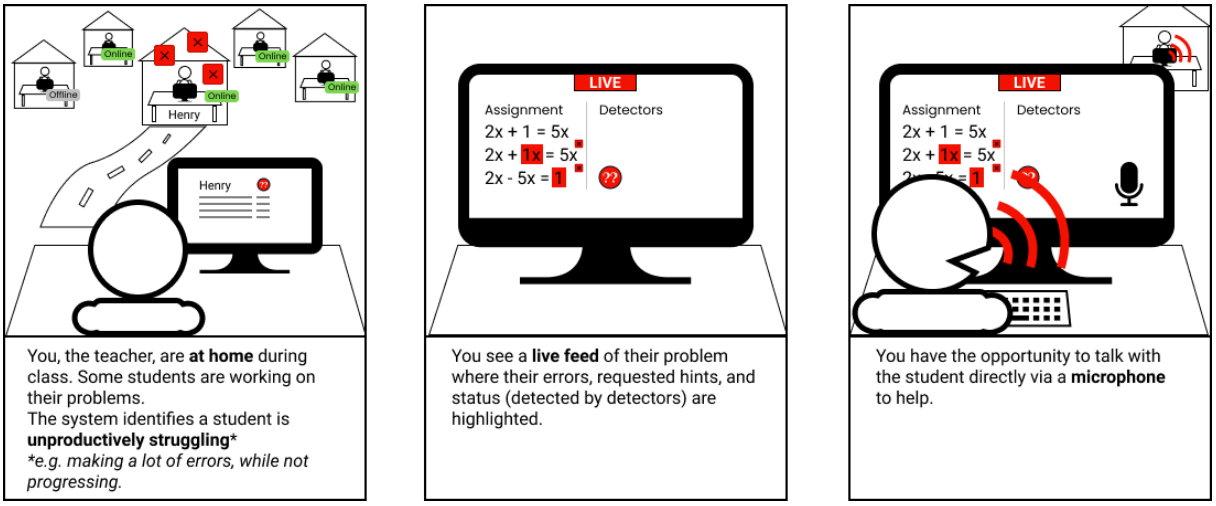


Figure A4. Remote, live and microphone.

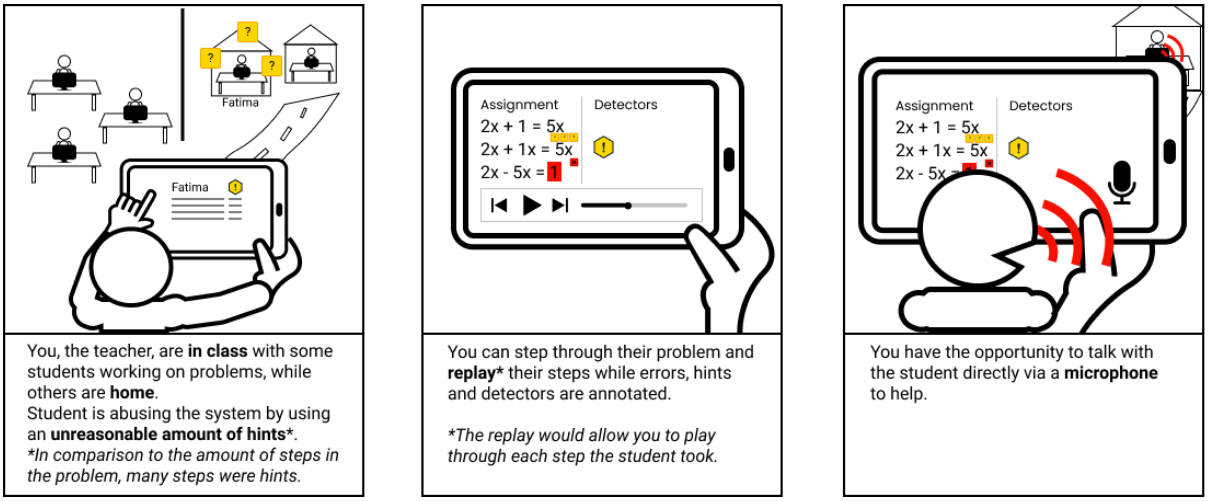
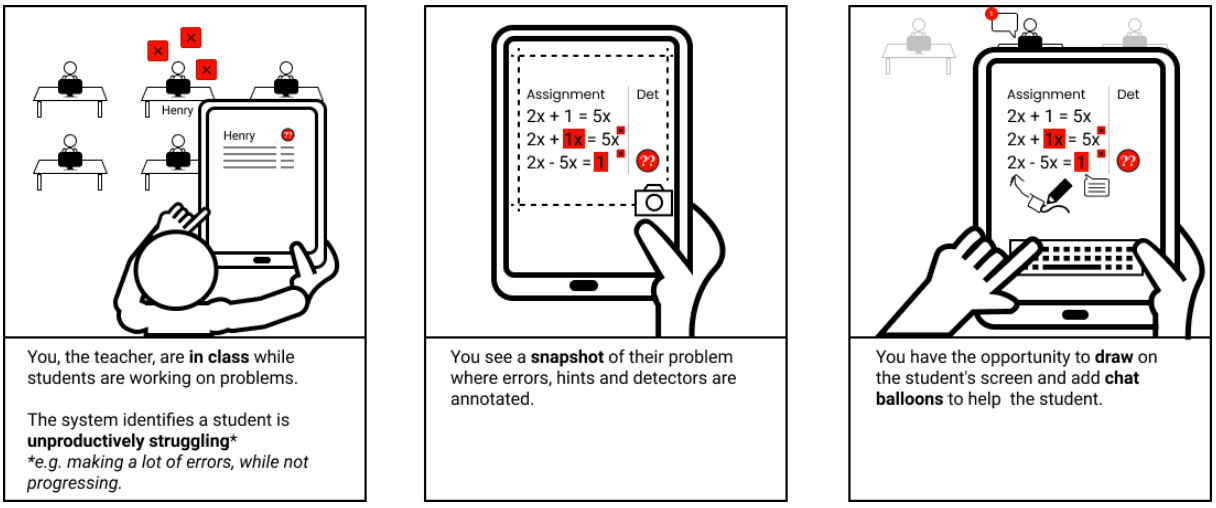


Figure A5. Hybrid, replay and microphone.



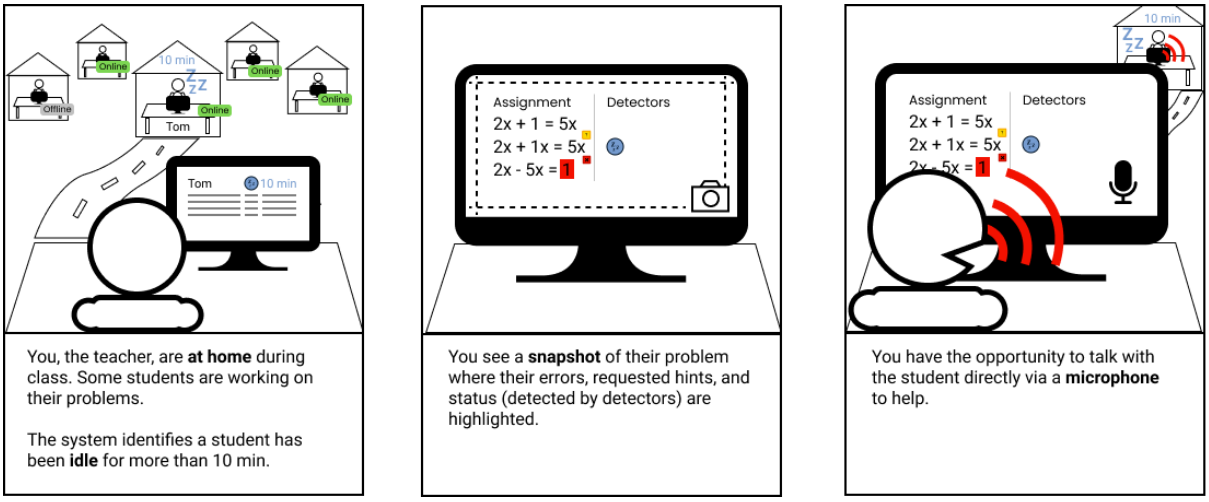
You, the teacher, are **in class** while students are working on problems.

The system identifies a student is **unproductively struggling***
**e.g. making a lot of errors, while not progressing.*

You see a **snapshot** of their problem where errors, hints and detectors are annotated.

You have the opportunity to **draw** on the student's screen and add **chat balloons** to help the student.

Figure A6. In-class, snapshot and draw & chat balloons.



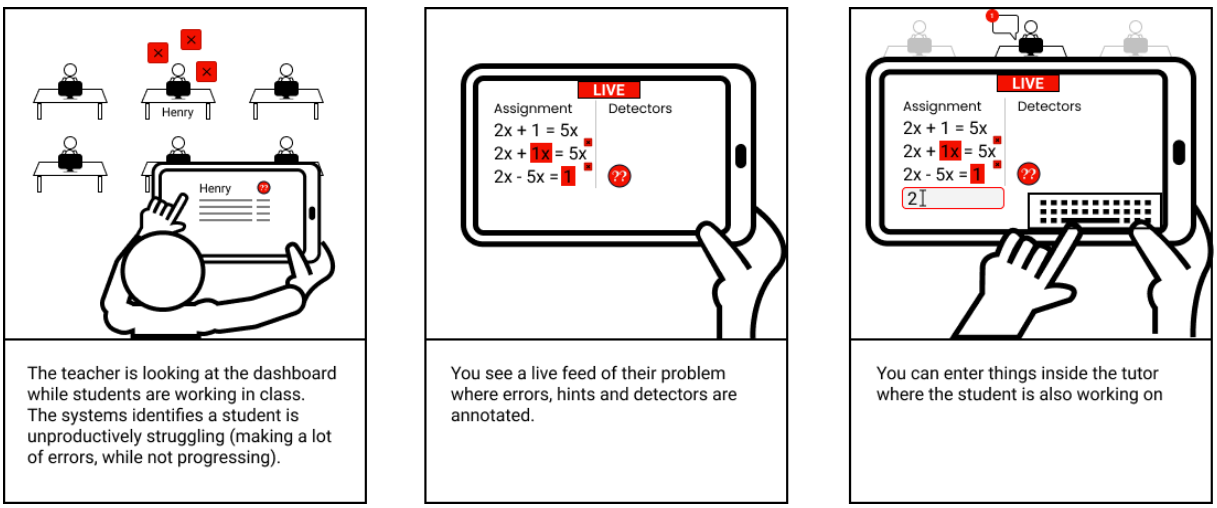
You, the teacher, are **at home** during class. Some students are working on their problems.

The system identifies a student has been **idle** for more than 10 min.

You see a **snapshot** of their problem where their errors, requested hints, and status (detected by detectors) are highlighted.

You have the opportunity to talk with the student directly via a **microphone** to help.

Figure A7. Remote, snapshot and microphone.



The teacher is looking at the dashboard while students are working in class. The systems identifies a student is unproductively struggling (making a lot of errors, while not progressing).

You see a live feed of their problem where errors, hints and detectors are annotated.

You can enter things inside the tutor where the student is also working on

Figure A8. In-class, live and typing in the student's interface.

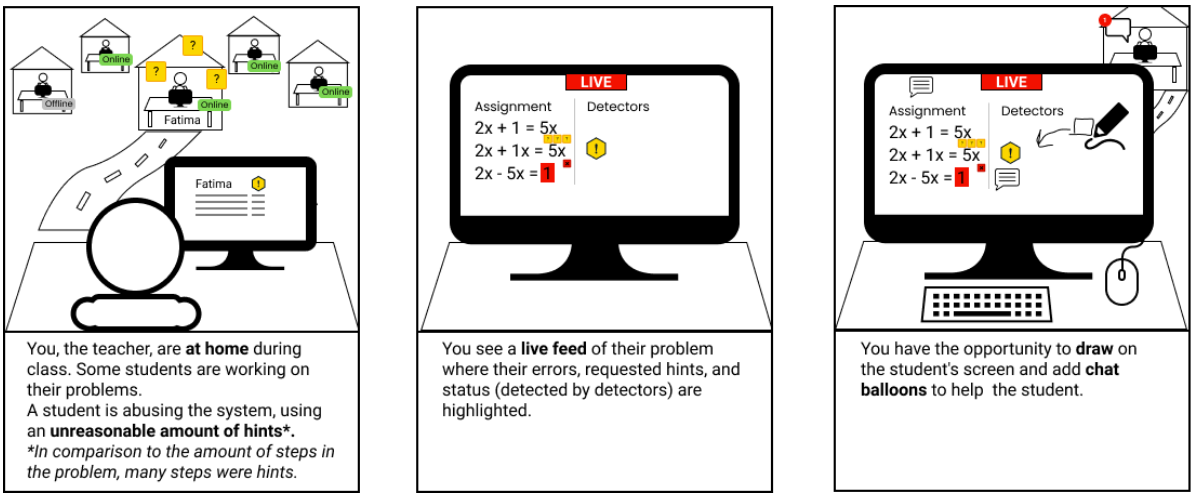


Figure A9. Remote, live and draw & chat balloons.

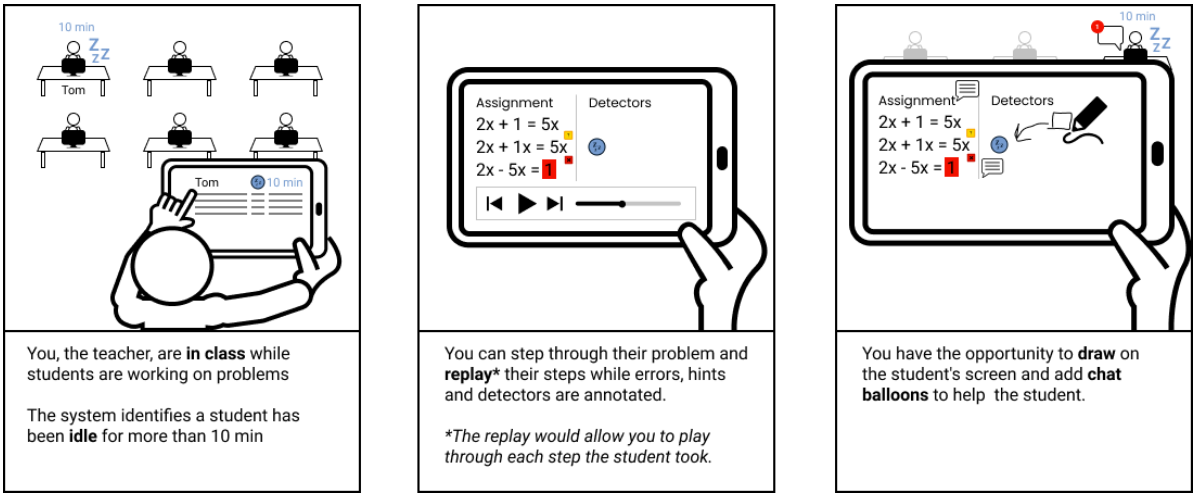


Figure A10. In-class, replay and draw & chat balloons.

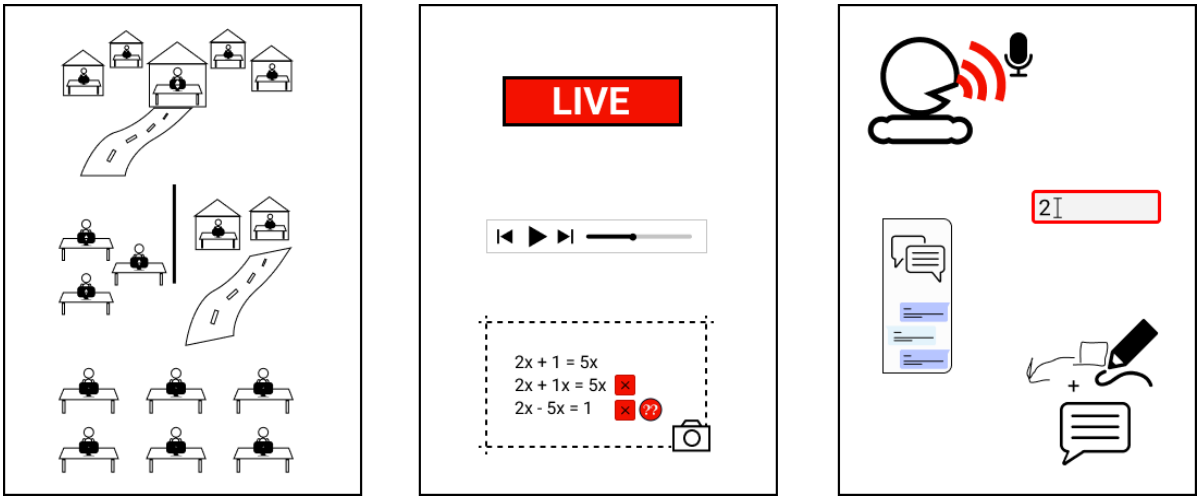


Figure A11. Do it yourself storyboard. Teachers were asked to create their optimal scenario given all the elements to choose from which were shown in the previous storyboards.

DASHBOARD (CLASS OVERVIEW)					
Task	**Sub-step**	**Clear what action to do?**	**Clear that the action is available?**	**Clear if action was right or wrong**	**Potential problems**
Question			**Answer**		
- *Given you were in class, how would you use this to check who has progressed the farthest?*					
Sort the progress					
Click on current/all notifications					
- **inspect notifications** - *In class, based on this information, how would you decide who needs your help first?* - *Which information is most/least important?* - *How could we improve notifications to make you more able to use it in class?*			-		
Go to Donny Darko in three different ways	Click on student name				
	Click on notification				
	Click on problem set (ask if they don't)				

Integration - *Do you see any way in which the dashboard can be improved to make it easier for you to use it in class?*	
Info - *What if you could see any kind of class-level information from students here, what info would you like to see here?*	

DEEP DIVE (PROBLEM REVIEW + SNAPSHOT OVERVIEW OF PROBLEM)					
Task	**Sub-step**	**Clear what action to do?**	**Clear that the action is available?**	**Clear if action was right or wrong**	**Potential problems**
Question			**Answer**		
Find out what the student needs help with	Filter problems based on area of struggle Cancel Variable Terms				
	Go to problem 2				
	Go to problem set level 4 (either via previous button or via click)				
	(optional) Increase the size of the snapshot				
	Click on replay for problem 1				
Usability - *Tell me about everything you see on screen and anything that's unclear.* - *What if you went to this student because you know they were struggling. How would you handle this?*					

- *How would you determine what the student is having problems with / what is difficult?* - *How would you handle an area of struggle?*	
Integration - *What scenarios would this be the most helpful (hint abuse, struggle and idle)? Why?* - *If you were in class would you use this, and how?*	-
Info - *What is good about the screen?* - *What would you like to improve on seeing the progression of the student? Why?* - *How would you improve the snapshot?* - *How much do notifications need to be linked?* - *Should there be some kind of coupling between skills and snapshot?*	

REPLAY (STEP-BY-STEP OVERVIEW OF PROBLEMS)					
Task	**Sub-step**	**Clear what action to do?**	**Clear that the action is available?**	**Clear if action was right or wrong**	**Potential problems**
Question			**Answer**		
Go to the fourth action of the problem					
Go to the next step					
Go to the previous step					
What did you think of the speed at which you were able to move back?					
Restart					
- What did you think of the speed? - Would you like to be able to control/set that yourself?					
Play the entire problem					
Go to the first step that contains a hint					
Drag the slider forwards					

Drag the slider backwards					
Drag the replay controls to a different place					
- Was it clear that you could move the controls? Why (not)? *- How can we improve that?*					
Usability - *What is your overall impression of this?*					
Info - *What kinds of things would you like to see in the replay?*					
Integration - *Would you see yourself using this in class?* - *If so, how would you use this inside your class when you're with students?* - *If no, why not?*					

Interactions in the classroom (transition from tool to student)

Question	Answer

- *Overall, would you use this in class?*
  - *If so, how?*
  - *If no, why not?*
- *What would you like to do with this information given that you were in class*
- *What do you think of the idea of clicking an AOS and putting that on the student screen?*
- *What part of this would you use with a student? How and why?*
	-

Appendix C: Architecture

As can be seen in figure C1 the architecture consists of multiple separated components.

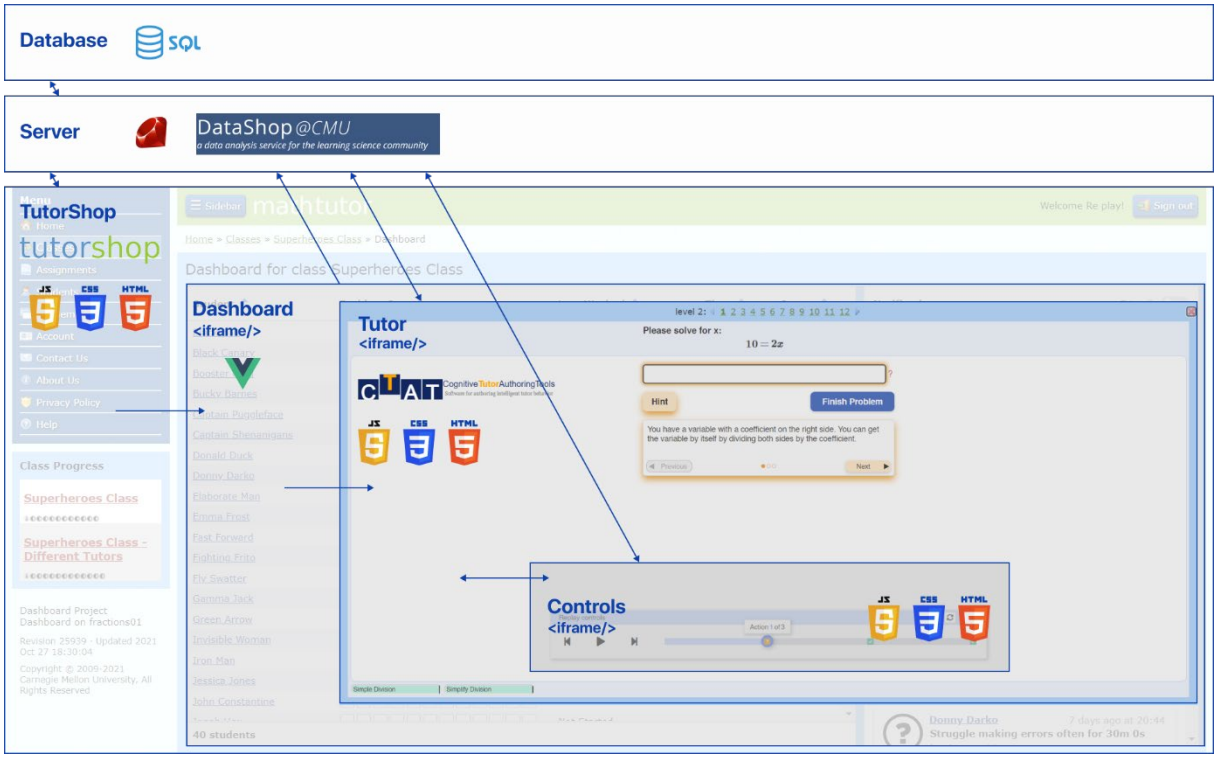


Figure C1. Architecture from the application. As can be seen multiple front-end layers are used. Each front-end layer is encapsulated in an iframe element.

Back-end

The CTAT/TutorShop architecture back-end consists of a Ruby server and an SQL database. The Vue dashboard code requests data analytics from TutorShop via the Ruby server. Afterwards only new data is updated via a direct connection between the dashboard and the server. Furthermore, it allows access to student's problem data and using Tutorshop it renders tutor problems from the class overview in an iframe (see Figure C1). When a tutor problem page is loaded DataShop log data is sent to the problem review part of the code. For more information on the data that the dashboard receives visit https://github.com/CMUCTAT/CTAT/wiki/

Front end

Vue dashboard.

ClassDashboard.vue This is the class overview page (see Figure C1). Its main components are the table (*ClassRoster.vue*) and the notification panel (*ClassNotifications.vue*).

StudentDetails.vue This is the student deep dive page. Its main components are the areas of struggle (top left; *StudentAreasOfStruggle.vue*), the problems list (bottom left; *StudentProblems.vue*) and the snapshot (right; *TutorInFrame.vue*).

Snapshot. The snapshot runs a URL from Tutorshop for a particular problem inside an iframe element. Tutorshop then loads a file that receives messages from the CTAT library (*ctat.min.js*). A user can request to view a specific snapshot or replay of a problem. When the user requests that problem in the deep dive, a tutor URL, created in the dashboard, is served from the back-end. Moreover, a (set of) file(s) containing the controls for snapshot replay are served with it.

snapshot.js Is the file that runs the code displaying the hoverable indicators on the snapshot panel. First, it receives a message from *ctat.min.js* called 'startRestoreSteps', which contains a summary of all the steps the student performed in the tutor. Then all steps are replayed by the tutor. During each replay the snapshot file intercepts student transaction messages ('restoreTx') from *ctat.min.js*. When it receives a transaction, it places an SVG element, the indicator, on the tutor interface (an html file). This element contains the logic, that is eventListeners, for interacting with the snapshot.

Replay. The structure described for snapshot works similarly for replay. However, instead of displaying the transactions in the tutor's html file, it has its own file contained in an iframe on top of the tutor.

replay.js The replay file receives the same start and transaction message from *ctat.min.js.* In addition, the replay file can communicate with the *ctat.min.js* file. It uses functions in *ctat.min.js* to control when transactions arrive and can restart ('reboot()') the tutor. Furthermore, it controls the dragging of the replay bar around the screen.

replay-bar.html This is the file that displays the replay bar.

replay-bar.css This is the stylesheet for the replay bar.

Tutors. There are a lot of tutors created via CTAT. A challenge was to get the tools to work for each tutor run in a student's browser. With snapshot and replay this worked well. However, there are still some unresolved issues with detectors (see Appendix F). Tutors use so-called brd files. These brd files are used to create the tree-like structure of the problem solution path and can be created using CTAT (Aleven et al., 2009).

Detectors. Each tutor can run detector code. These detectors are small software modules that analyze student analytics. Detectors execute next to tutors in a student browser. They analyze the transactions, recorded events in the tutor, from a student. After receiving a transaction then send messages containing information regarding their state using mail workers (*transaction-mailer-user.js & mail-worker.js;* for more information, see https://docs.google.com/document/d/1KILSGU2BFPYAjAR3NamteAWGacW99JWTdUgqQroFeQ4/edit?usp=sharing). There are a couple of issues with the detectors that need to be resolved (see Appendix F).

idle.js The idle detector.

struggle.js The struggle detector.

system_misuse.js The system misuse detector.

critical_struggle.js The critical struggle detector.

doing_well.js The doing well detectors.

Appendix D: Separate ideas from teachers

Below is a list of some interesting ideas worth considering that teachers came up with. Also, there is a list of our own ideas (inspired by teachers) here: https://docs.google.com/document/d/1IZV0Yqy_v9vsalWpkFjg117ZYWAMysUHBCEtptPVS6w/edit?usp=sharing

- T11 mentioned that they would want to know what skill a student is working on and whether they can determine that multiple students have difficulty on that skill.

Class overview

- Grid view for class instead of roster view
- Display online indicator (green dot in corner of blocks) to show teacher which problem set (in deep dive and class overview) a student is currently on.
- Displaying common areas of struggle for the entire class

Notifications

- Pop up notifications
- Clicking on skill in notifications would filter the notifications for that skill
- Sort notifications based on problem set (not that popularly requested per se but might be useful and relatively easy to implement)
- Display whether a notification is still incrementing in time or not.

Deep dive

- Chats (multitask helping) / audio chats / microphone for remote
- Recording proof of helping.
- Giving a similar problem to a student / the entire class
- Extra idea: [T11] If you're on something for too long it would be helpful that you're notified of that. (As a teacher)
- Problem set blocks names at the top of the problem list (keep in mind that they need to be responsive.

Snapshot

- Clicking on skill in snapshot would result in filtering problems for that skill

Replay

- Draw and chat

Student side

- Notepad for students
- Student side use of replay (and snapshot) [Similar software for students]

Student tutor

- Moreover, T15 indicated that sometimes hints can contain difficult to comprehend words for students, like 'constants'. However, she said it was good that these words were used since they should be learned.

Appendix E: TutorShop redesign argumentation

For the design files shown in the images, go to
https://www.figma.com/file/NjnFsyy5LOCGyowMFkW9uF/Deep-dive?node-id=54%3A0

Overall changes.

Color. Every color creates a visually separate element and element boundary. A clearly defined boundary makes users perceive areas as groups (Palmer, 1992). However, there is a limit to how many elements a user can perceive and keep in working memory. As a rule of thumb Miller's law states that 7+/- 2 items is the maximum number of items one can keep in working memory at a time (Cowan, 2010). To accommodate all users a good rule of thumb is about 4/5 elements, so that the screen is more easily perceived at a glance (Cowan, 2010).

Moreover, subjectively, the group found the site to look better. This creates an argument for the changes based on the aesthetic-usability effect: when a site looks better, it is perceived as more usable (Kurosu & Kashimura, 1995; Norman, 2004). As can be seen in Figure E1 the number of separable elements is close to 9.

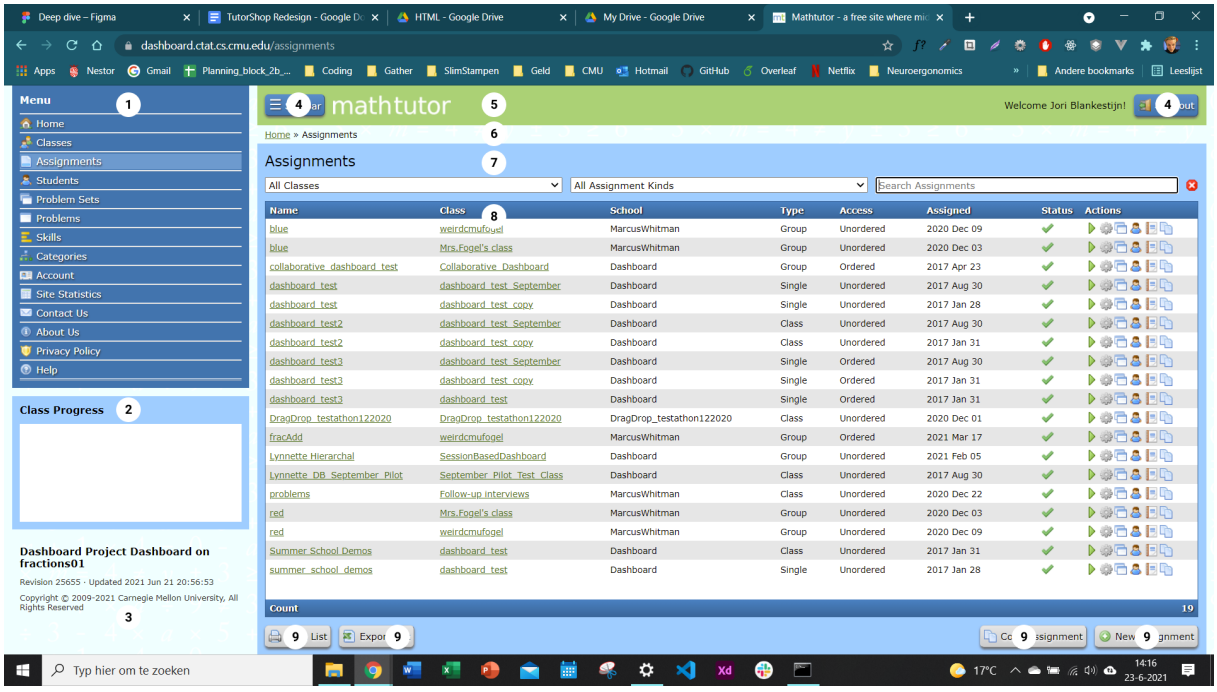


Figure E1. Current TutorShop interface. The numbers indicate the visually separate elements on screen.

By removing and replacing colors, the amount is reduced to 6 (see figure E2). Moreover, if you take elements 1, 4, and 5 together inside a visually separate group of navigational elements. Then it becomes exactly 4 element groups.

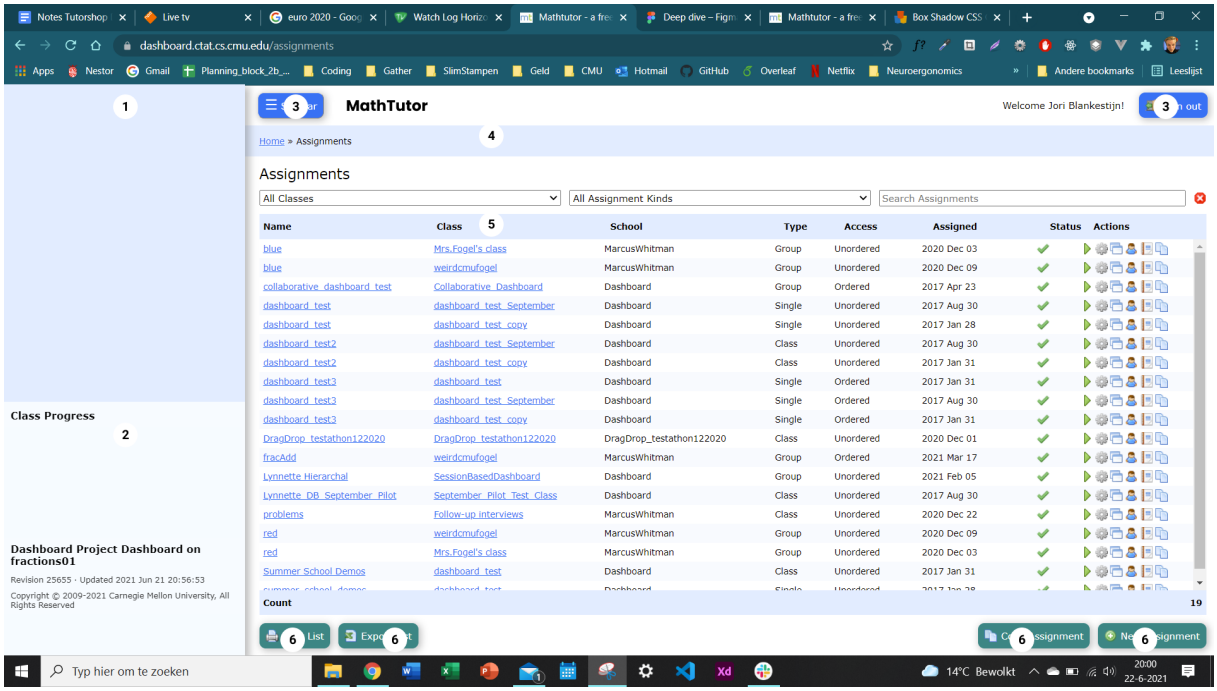


Figure E2. Proposed redesign. The numbers indicate the visually separate elements.

Move copy and new assignment, and print and export list above the table. According to the goal-gradient hypothesis, the closer the user is to the goal (proximally) the more likely it is to be approached by a user (Kivetz, Urminsky, & Zheng, 2006). A table is read from top to bottom, it is sometimes sorted by a user, and during search the result appears on top. Therefore, the buttons that manipulate items are often closer to the user's cursor when they are located on top, thus more likely to be approached. Furthermore, people read from top to bottom, which makes a user more likely to perceive the options.

Buttons

Remove text shadows. All text has been checked using the contrast checker, which confirmed every contrast was in compliance with at least an AA level from the Web Content

Accessibility Guidelines-2 (Contrast Checker, n.d.). Therefore, visually impaired people should be able to read the text conform to internationally established usability norms.

Remove gradients, outlines/borders, and change box shadow to be more subtle. In particular, items that are similar in form are grouped together (law of similarity; Kubovy & van den Berg, 2008). Thus, these "visual complexities" are redundant, since a user can tell that this is a button by color and shape alone. You should remove complexity as much as possible. Namely, Hick's law (Hick, 1952) states that the more complexity there is on screen the more time it will take to make a decision. There is a necessary amount of complexity, but other than that, one should remove redundancies (Tesler's law; Yablonski, 2020).

Add more padding to buttons. The surface area and distance are a function of the reaction time it takes to press a button (Fitts' law; Fitts, 1954). Therefore, to make the interface easier to use and quicker to use, button size should be increased.

Padding and margins

Add more padding to breadcrumbs, table header and menu items. By the same reasoning as for button padding, padding for the breadcrumbs, table header and menu items should be increased with the same padding (8px seems reasonable). First, this looks more visually appealing due to consistency and space. Second, it makes the interface easier to use (Fitts, 1954).

Remove redundant margins and add shadow to the menu. The red boxes in Figure E3 indicate redundant margins. Their use of space is redundant, since a shadow would indicate the separation as well and on the outside of the interface, they don't separate from

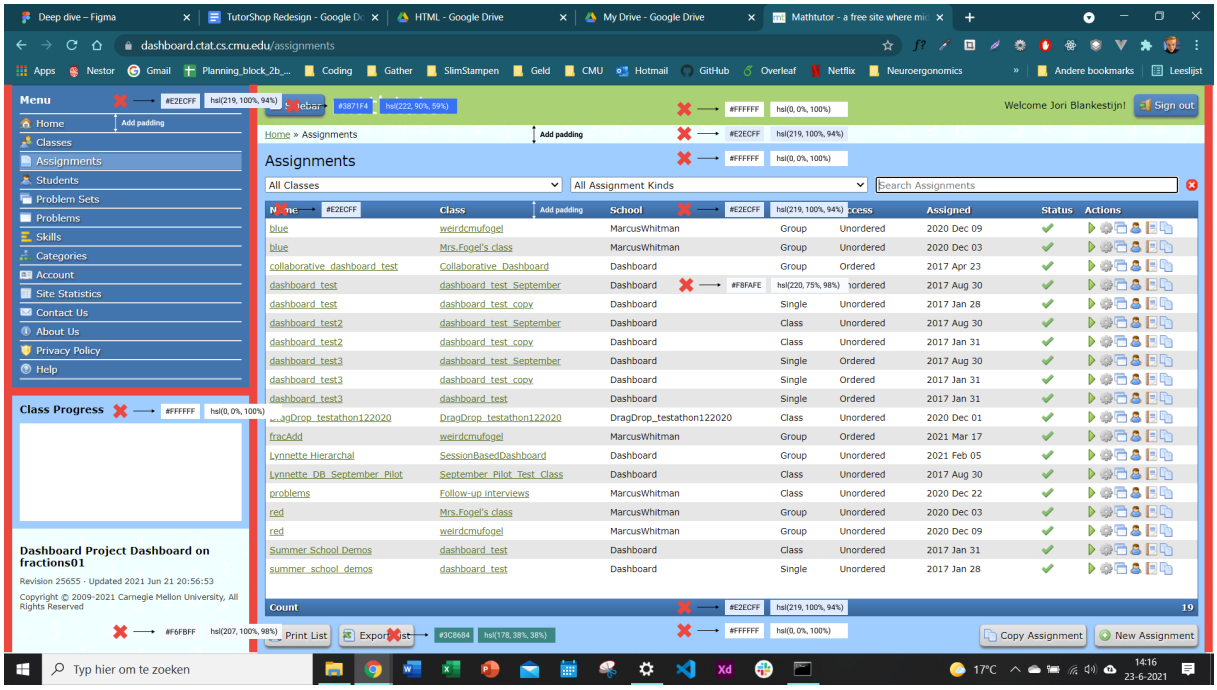


Figure E3. Proposed changes to old design.

Apart from the initial redesign changes that are discussed above, we also created a mock-up for a more involved redesign (see Figure E4).

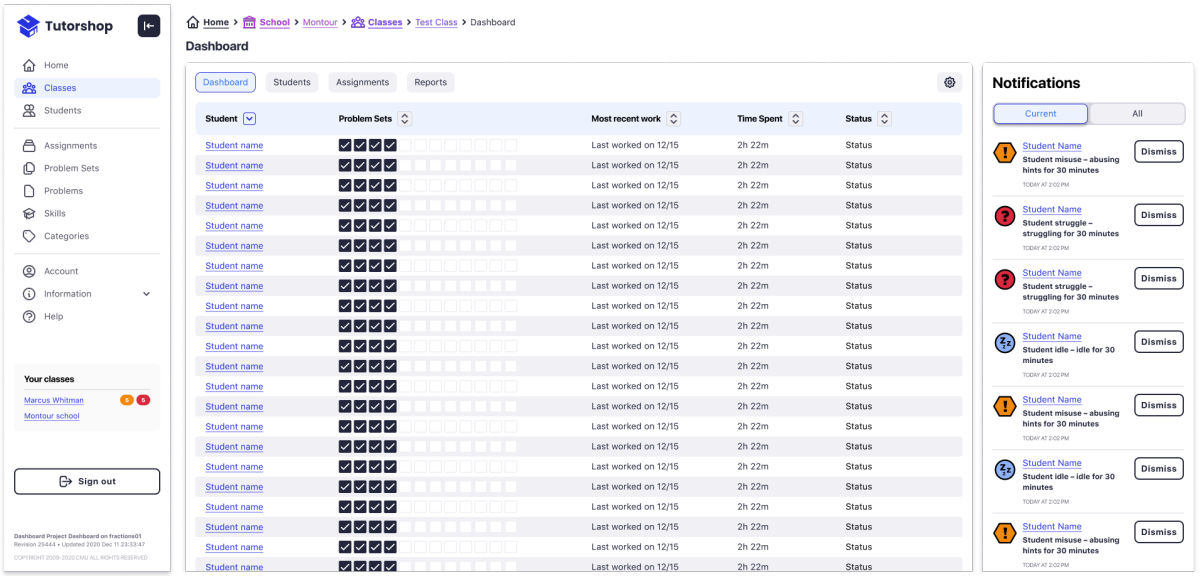


Figure E4. Full redesign of Tutorshop.

Furthermore, the logo on Tutorshop is currently plain text. To make Tutorshop recognizable as a brand, one should contemplate creating a logo for it. Therefore, we have created a logo concept for Tutorshop (see Figure E5).

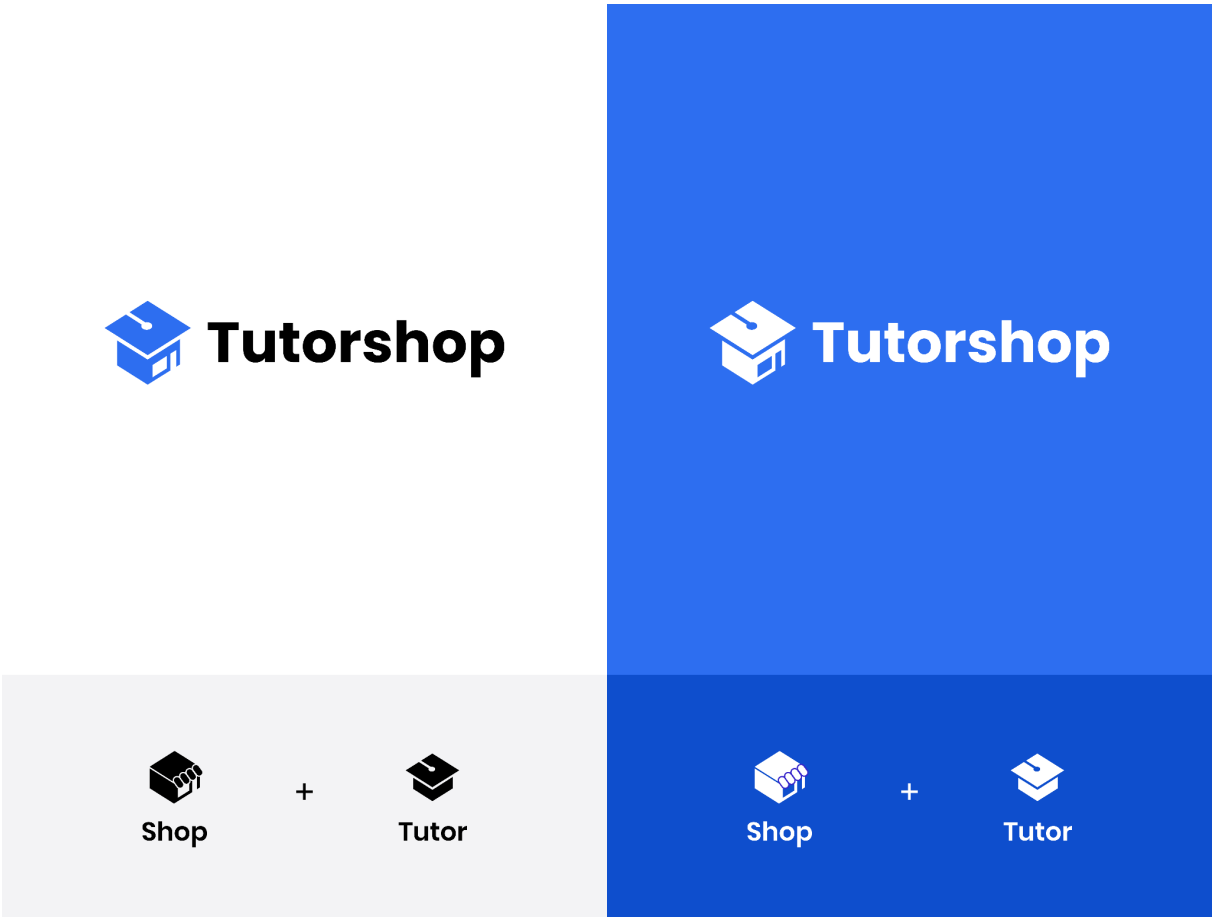


Figure E5. Proposed logo design.

References

Contrast Checker. (n.d.). Retrieved June 22, 2021, from https://webaim.org/resources/contrastchecker/

Cowan N. (2010). The Magical Mystery Four: How is Working Memory Capacity Limited, and Why?. *Current directions in psychological science, 19*(1), 51–57. https://doi.org/10.1177/0963721409359277

Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology, 47*(6), 381-91.

Hick, W.E. (1952). "On the rate of gain of information" (PDF). *Quarterly Journal of Experimental Psychology. 4* (4:1): 11–26. doi:10.1080/17470215208416600

Kivetz, R., Urminsky, O., & Zheng, Y. (2006). The Goal-Gradient Hypothesis Resurrected: Purchase Acceleration, Illusionary Goal Progress, and Customer Retention. *Journal of Marketing Research, 43*(1), 39–58. https://doi.org/10.1509/jmkr.43.1.39

Kubovy, M. & van den Berg, M. (2008). The whole is equal to the sum of its parts: A probabilistic model of grouping by proximity and similarity in regular patterns. *Psychological Review, 115*, 131-154.

Kurosu, M., & Kashimura, K. (1995). Apparent Usability vs. Inherent Usability. *Conference companion on Human factors in computing systems - CHI '95.*

Norman, D. A. (2004). Emotional Design: Why We Love (Or Hate) Everyday Things. New York: Basic Books.

Palmer, S. (1992). Common region: a new principle of perceptual grouping. *Cognitive Psychology, 24*, 436-447.

Yablonski, Jon (21 April 2020). "Chapter 9: Tesler's Law". Laws of UX: Using Psychology to Design Better Products & Services. p. 87. ISBN 9781492055280.

Color files

NEW

```
/* modifiers */
$step: 10%
$half-step: $step / 2

/* color variables */
$black: hsl(0deg, 0%, 0%)
$deepgray: hsl(212deg, 18%, 29%)
$darkgray: hsl(213deg, 13%, 42%)
$gray: hsl(213deg, 19%, 70%)
$lightgray: hsl(213deg, 49%, 86%)
$white: hsl(0deg, 0%, 100%)
$lightergray: hsl(207deg, 100%, 98%)
$purple:
$firebrick: hsl(0deg, 75%, 33%)
$red:  hsl(6deg, 61%, 54%)
$orange: hsl(41deg, 100%, 45%)
$olive:
$yellow: hsl(49deg, 80%, 73%)

$deepgreen:  hsl(179deg, 100%, 16%)
$darkgreen: hsl(178deg, 38%, 38%)
$green: hsl(180deg, 60%, 62%)
$lightgreen: hsl(181deg, 49%, 80%)

$deepblue:  hsl(221deg, 63%, 35%)
$darkblue: hsl(222deg, 90%, 59%)
$blue:  hsl(222deg, 91%, 92%)
$lightblue:  hsl(219deg, 100%, 94%)

/* special */
$dot: #faa400
$kiwi: #d0e8c8

/* lynnette */
$brightwhite: #eee
$outerspace: #202a58

/* theme colors for planets */
$planet_empty: #404c8d
$planet_fill: #5e6aa2
```

OLD

```
/* modifiers */
$step: 10%
$half-step: $step / 2

/* color variables */
$black: hsl(0deg, 0%, 0%)
$deepgray: hsl(0deg, 0%, 20%)
$darkgray: hsl(0deg, 0%, 40%)
$gray: hsl(0deg, 0%, 60%)
$lightgray: hsl(0deg, 0%, 80%)
$white: hsl(0deg, 0%, 100%)

$purple: hsl(270deg, 50%, 60%)
$firebrick: hsl(0deg, 60%, 50%)
$red: hsl(0deg, 100%, 60%)
$orange: hsl(48deg, 100%, 50%)
$olive: hsl(60deg, 50%, 60%)
$yellow: hsl(60deg, 100%, 50%)

$deepgreen: hsl(90deg, 100%, 20%)
$darkgreen: hsl(90deg, 50%, 40%)
$green: hsl(90deg, 50%, 60%)
$lightgreen: hsl(90deg, 100%, 80%)

$deepblue: hsl(210deg, 100%, 20%)
$darkblue: hsl(210deg, 50%, 40%)
$blue: hsl(210deg, 50%, 60%)
$lightblue: hsl(210deg, 100%, 80%)

/* special */
$dot: #faa400
$kiwi: #d0e8c8

/* lynnette */
$brightwhite: #eee
$outerspace: #202a58

/* theme colors for planets */
$planet_empty: #404c8d
$planet_fill: #5e6aa2
```

_colors.sass

```
/* modifiers */
$step: 10%
$half-step: $step / 2

/* color variables */

$black: hsl(0deg, 0%, 0%)
$deepgray: hsl(212deg, 18%, 29%)
$darkgray: hsl(213deg, 13%, 42%)
$gray: hsl(213deg, 19%, 70%)
$lightgray: hsl(213deg, 49%, 86%)
$white: hsl(0deg, 0%, 100%)
$lightergray: hsl(207deg, 100%, 98%)

// $purple:
$firebrick: hsl(0deg, 75%, 33%)
$red: hsl(6deg, 61%, 54%)
$orange: hsl(41deg, 100%, 45%)
// $olive:
$yellow: hsl(49deg, 80%, 73%)

$deepgreen: hsl(179deg, 100%, 16%)
$darkgreen: hsl(178deg, 38%, 38%)
$green: hsl(180deg, 60%, 62%)
$lightgreen: hsl(181deg, 49%, 80%)


$deepblue: hsl(221deg, 63%, 35%)
$darkblue: hsl(222deg, 90%, 59%)
$blue:  hsl(222deg, 91%, 92%)
$lightblue: hsl(219deg, 100%, 94%)

/* special */
$dot: #faa400
$kiwi: #d0e8c8

/* lynnette */
$brightwhite: #eee
$outerspace: #202a58

/* theme colors for planets */
$planet_empty: #404c8d
$planet_fill: #5e6aa2
```

Appendix F: Unresolved issues in the dashboard.

Improvements

Making the detector code work properly. Currently the detector code does not work as well as it could. Specifically, there is a small bug that sometimes makes notifications display negative time indications (e.g. student has been idle for -1h.).

Replay going backwards takes time. Similar to the refresh issue just described, when going back a step, the replay page needs to be refreshed. Then it replays all the transactions minus one to get to the previous step. It was implemented in this manner so the tutor - which interprets student actions - can do the exact same updates as were performed in the student interface. A solution for not refreshing when going backwards is updating the UI once using all the actions and recording the changes in the UI. Then going back-and-forth between actions could be handled in the front-end by the replay code. A caveat is that some tutors in the CTAT library contain tutor-specific JavaScript files, which makes it hard to create tutor-agnostic replay that does not have to refresh.

Bugs

Class overview.

- The time column in the class overview does not always show minutes and does not update while students are working. It seems like the updates only happen when a detector gets sent through TutorShop. It might also be that it uses studentAssignment instead of problemSummaries. Adding up the times in problemSummaries would likely result in more up to date total times.

Detectors & Notifications.

- The detectors can sometimes show negative numbers for the time the student was in the state. This does not seem to happen when students have stopped working, but when the state switches from one state to idle.
- The doing well detector does not seem to work. Each time the student goes to a new problem, the doing well detector's attempt window resets. Thus, only if during a problem a student can perform 8 steps (8 out of 10 last steps in the attempt window need to be correct) the doing well detector can go off. This should not be the case. It can be that there is another problem, but this seems the most pertinent.
- Detector timestamps are sent the moment the detector goes off. However, most detectors should actually have timestamps before the threshold that the detector goes off. The simplest example for this is the idle detector. When the threshold of two minutes is met, the detector should actually show a timestamp two minutes before the threshold was reached.

Deep dive screen.

- It seems like there is something not completely correct in the implementation of areas of struggle. Namely when a student has a lower skill level than the default, the skill does not always show in the areas of struggle panel of the deep dive. Therefore, it might not work correctly.
- The number of hints in the problems overview of the deep dive does not correspond to the number of hints in the snapshot. This is the case because the hint pagination requests are not sent to the dashboard.
- When clicking on a student name in the class overview, the dashboard takes you to the first problem in problems overview of the deep dive. However, after clicking out of that problem, the indicator (the blue dot) indicating that it has been seen by the teacher does not show that it has been seen.

- When clicking on a problem set where no problems have been performed by the student, the snapshot still loads a problem. It should not load a problem, since none have been performed.

Replay.

- Sometimes when replay starts during simulated students the replay bar does not show any indicators. The transaction information is sent, because the play button works and the steps can be replayed, but the bar does not indicate anything.
- It seems like the skill bars do not always light up when a step involves a skill. This mechanism worked but it might not work completely correctly anymore.
- Dragging the replay bar while it is playing makes the play button remain in the play state and not in the pause state, while dragging the bar actually pauses the tutor replay.