Visualizing Student Problem Solving In Real Time

Yvonne Chen
evechen@uw.edu

Eleanor O'Rourke
eorourke@cs.washington.edu

ABSTRACT

Research in education highlights the importance of exposing student data to teachers, who can use this information to provide students with timely and accurate assistance. As digital technologies become more commonly used in classrooms, we have an opportunity to collect and communicate student data to teachers in real-time. However, little research has explored how to effectively visualize student data for teachers during class. In this work, we partner with the non-profit company Enlearn to design real-time visualizations of student data collected through their tablet-based software. We characterize the domain problem, design appropriate data abstractions, and present two new visual encodings to support teacher tasks. Informal feedback suggests that our visualizations would be valuable for teachers, and we aim to formally study our designs with classroom teachers in the future.

INTRODUCTION

Historically it has been impossible for K-12 teachers to record and summarize student performance as students are learning in class. Instead, teachers have typically depended on time after class to grade assignments and get a sense of student progress and misunderstandings. While this post-hoc analysis of student learning helps teachers adapt instructional content across lessons, there is strong evidence that feedback is most effective when it is given as a new concept is being introduced [7, 16]. Prior research in education suggests that teachers could benefit greatly from having access to student data in real-time, allowing them to assist struggling students more effectively during class [4, 11, 12].

As digital technologies become more common in classrooms, teachers can collect and analyze rich student data during class time. One early technology for collecting student data during lecture-based classes is the "clicker", which is used to assess student responses to multiple-choice questions during class [6, 12]. In smaller classes, tablet-based software is becoming common. As these types of technologies become ubiquitous, the challenge morphs to one of data accessibility rather than data availability. Little research has explored how student data should be visualized for teachers in real-time to most effectively support in-class analysis tasks.

Figure 1. Our concept visualization of real-time student problem-solving data. Students are grouped by the mathematical concept that they are currently working on, and we only display information for the current concept. This view is designed to help teachers rapidly identify struggling students so that they can provide individual assistance.

In this work, we explore the design of visualizations for communicating student progress to teachers in real-time. To conduct this research, we partnered with Enlearn, a non-profit company that creates tablet-based adaptive problem-solving software for K-12 classrooms. In Enlearn classrooms, students work on math practice problems on individual tablets during class, while the teacher monitors progress on her own tablet and assists individual struggling students. Enlearn has studied two different methods of visualizing student data for teachers, but found that their designs were not appropriate for teachers' in-class needs. These findings highlight the need for further research in student data visualization design.

We first characterize the student data visualization problem space, and develop a set of design guidelines for real-time visualizations. These guidelines focus on the importance of providing actionable information for teachers and presenting data at both the student and concept level. We then define the operations needed to transform the data into a format that addresses teacher needs, specifically *sorting* and *computing* average performance. Finally, we present two new visual encodings for student data: one that displays information for the concepts students are working on in the present moment, and another that displays aggregate performance across multiple concepts. While we have not yet formally evaluated our visualization designs with teachers, we present a discussion based around early feedback from Enlearn staff.

RELATED WORK

Recent research in educational technology explores methods of exposing student data to assist teachers. In this section, we review this prior work and also describe Enlearn's early visualization designs for communicating student problem-solving data to teachers in the classroom.

Exposing Student Data for Teachers

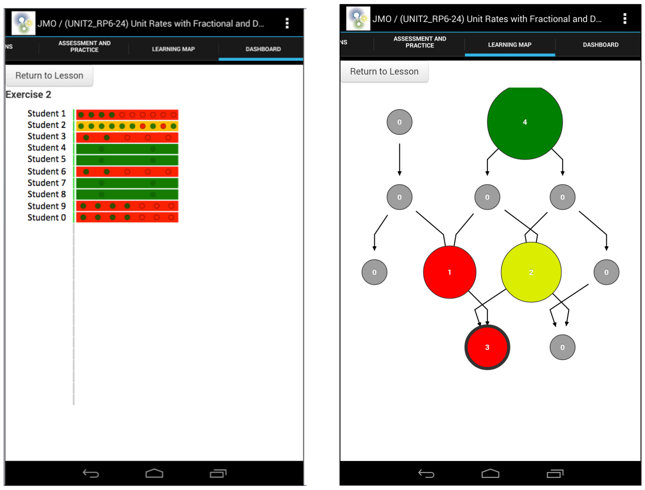
Education research shows that teacher behavior has a strong impact on student achievement [9, 18], and that teachers can benefit from the availability of real-time student data [4, 11, 12]. For example, Koile found that when an instructor was given access to student problem solutions through tablet-based technology in real-time, the instructor devoted 75% of class time responding to student misunderstandings [11]. With access to real-time data, research suggests that instructors can intervene during a lesson when students are confused [8], alter the pace of instruction based on student engagement [4], identify and assist students who are struggling [12], and choose topics of focus based on student feedback [11].

A number of systems have been developed for exposing student data for teachers. One technology that is prevalent in lecture-based classes is the "student response system" or "clicker," which is used to poll students on multiple-choice questions during class [6, 12]. A similar application designed for small classes is Plickers [15]. With the Plickers smartphone app, the teacher can scan the classroom while students hold up QR codes identifying a multiple-choice answer [15]. Researchers have also explored methods of providing instructors with access to student data outside of instruction time to monitor longer term academic progress [19, 3]. Kim et. al. developed a system for compiling student responses to MOOC exercise problems, which teachers reported were useful for capturing student thought processes, identifying misconceptions, and engaging students with content [10].

While research shows that student data can help teachers understand student progress and affect how they spend class time, little is known about how to visualize student data to effectively communicate with teachers. Most existing systems use traditional visualizations such as bar charts to display aggregate student data [6, 12] or show raw student solutions [2, 11]. However, these approaches are not appropriate for adaptive curriculums where students are working on different concepts simultaneously. Furthermore, we are not aware of any research that explicitly studies the utility of these visualization approaches for classroom teachers. In our work, we introduce two novel visual encodings for real-time student data based on design principles developed by characterizing teacher needs in the classroom.

Enlearn's Existing Visualizations

This spring, Enlearn ran a one-week study exploring the effectiveness of two different visualizations designed to expose real-time student data for teachers. The study revealed that both designs have usability and readability problems.



(a) (b)

Figure 2. Enlearn's initial visualization designs. Figure (a) shows a table view of overall student performance over the course of the lesson. Figure (b) shows a graph of concept interdependency, with students organized by their current concept. Neither visualization was successful at providing the actionable information teachers need during class.

Enlearn's first design, shown in Figure 2(a), displays a table view showing student problem-solving pace and problem correctness. While this visualization provides information about individual student progress, it does not convey which concepts students are currently struggling with. Enlearn found that this made it difficult for teachers to know how to best assist struggling students during class time. Enlearn's second design, shown in Figure 2(b), displays a graph showing concept dependencies. Each node is colored according to average student performance on that concept, and sized based on the number of students currently working on that concept. While this visualization displays information about the concepts that students are struggling with at a general level, it does not provide information about which specific students are struggling. Enlearn found that teachers were not able to effectively interpret this visualization during class.

More generally, teachers told Enlearn that they want visualizations that provide immediately actionable information that will aid them in effectively assisting students. The goal of our work is to build off these findings and design more effective visualizations for communicating student problem-solving data to teachers during class.

METHODS

This research explores the challenge of designing effective techniques for visualizing student problem-solving data for teachers. We make contributions and multiple levels of Munzner's nested model of visualization design [14]. First, we characterize the domain problem, next we discuss our data abstraction designs, then we present our novel visual encoding designs, and finally we describe our implementation.

Problem Characterization

The initial goal of this research was to understand the format of the problem-solving data collected by the Enlearn software and accurately characterize the tasks that teachers want to accomplish using this data. The challenges that Enlearn faced with their initial visualization designs shows the importance of understanding user needs; Enlearn's visualizations failed in large part because they did not effectively support teacher tasks. While we did not directly meet with teachers, we worked closely with Enlearn staff to understand the feedback they received in response to their initial visualizations. Here, we first describe the structure of the problem-solving data collected by Enlearn, and then describe teacher tasks. We provide a set of design guidelines for our visualizations based on this problem characterization.

Enlearn Data Structure

In Enlearn classrooms, each student solves problems on a personal tablet. Enlearn logs the following problem-solving information: the id of the student, the id of the problem, the id of the associated problem set, the problem completion timestamp, and whether or not the student answered the problem correctly. Enlearn's adaptive curriculum is designed as an extension of the JUMP Math curriculum [13]. In JUMP Math, students solve a linear progression of Exercise and Workbook problems. In Enlearn's adaptive version of the curriculum, students practice each type of Exercise or Workbook problem until they have mastered the concept covered by that problem. As a result, students may work on multiple practice problems of the type "Exercise Problem 3" before moving on to the next problem type. Enlearn refers to each problem type as a *problem set*, but we use the term *concept* in this paper since it better matches how teachers refer to the groups of problems.

Teacher Tasks

Through our discussions with Enlearn staff, we learned that there are two primary tasks that teachers want to accomplish using student problem-solving data during class. First, teachers want to determine which students need individual assistance and what specific concepts they are struggling with at any given moment. Ideally, teachers would also be able to see if multiple students are struggling with the same concept at the same time so that they can pull the students aside to work with as a group. Second, teachers want to determine how each student is performing overall during the current lesson. If many students are struggling with the content, the teacher may chose to re-teach parts of the lesson.

While Enlearn's visualizations attempted to target these tasks, they did not provide enough information for teachers to quickly determine which students were struggling and what concepts they needed help with. The overwhelming feedback that Enlearn was given by teachers after their trial was that the real-time visualizations need to provide information that is glanceable and immediately actionable. During class, teachers are busy adapting lessons, responding to students, and managing organizational challenges. They do not have time to explore a complex visualization of student data. Instead, they need a simple view that provides them with the exact information they need to accomplish in-class tasks such as assisting students and altering their teaching strategy.

Design Guidelines

From our characterization of the structure of Enlearn's problem-solving data and the tasks that teachers need to accomplish using that data, we developed the following design guidelines for our visualizations:

- Visualizations must provide actionable information that clearly displays student performance on practice problems.
- Visualizations must show data for individual students so that teachers can easily intervene when needed.
- Visualizations must display concept-level data so that teachers can assess student understanding of each concept.

We use these guidelines throughout the remainder of our work to focus our data abstraction and visual encoding designs.

Data Abstractions

With our improved understanding of Enlearn's data structure and teachers' real-time visualization needs, we developed a set of data abstractions. First, we identify the set of low-level data *operations* that need to be performed using the taxonomy developed by Amar, Eagan, and Stasko [1]. To assess student performance and determine which students are struggling, teachers must *compute derived values* of student performance across multiple problems or concepts. To determine which concepts students are struggling with, teachers must also *sort* students by concept. Finally, to quickly find individual students, teachers must *sort* data alphabetically. Since teachers do not have time to perform these operations manually during class, our visualizations need to automatically compute averages and sort students.

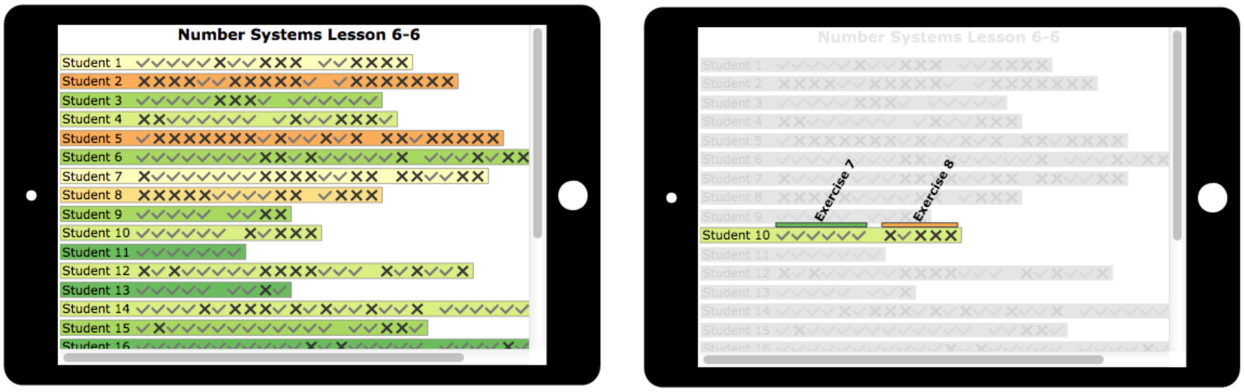
The raw problem-solving data consists of the problem id, problem timestamp, problem correctness, student id, and concept id. We transformed this data into a nested JSON object that would make it possible to perform the necessary computations and sorting. First, we group problem data by student so that we can calculate how well each student is performing overall. With each student, we further group problem data by concept so that we can determine how well students perform on each concept. To capture information about how students spend their time, we calculate a start and end time for each concept. For each problem, we store problem correctness and the problem timestamp. This nested *data type* allows us to compute the average performance of each student for both individual concepts and the entire lesson.

Visual Encodings

To support the tasks that teachers want to accomplish using student problem-solving data during class time, we developed two separate visual encodings. We describe the goals and structure of each visual encoding below, and then discuss our visual design choices in detail.

Concept Visualization

We designed the concept visualization to help teachers determine which students need individual assistance in the present moment. In this visualization, students are sorted first by concept and then alphabetically. For each student, we display a bar showing problem-solving information for the concept that the student is currently working on. The bar shows



(a) (b)

Figure 3. Our aggregate visualization of student problem-solving data. Figure (a) shows the main view, with student bars are colored according to their average performance across all problems. Figure (b) shows student-specific concept-level information that is accessed by clicking on a student bar.

check marks for correct problems and cross marks for incorrect problems, indicating both the number of problems a student has completed for this concept and problem correctness. The bar is colored according to the student's average performance on the concept. A screenshot of the concept visualization can be seen in Figure 1.

The goal of this visual encoding is to allow teachers to rapidly identify students who are struggling. One issue with the visualizations that Enlearn tested was that it was difficult for teachers to determine which specific concepts were currently struggling with. By grouping students by concept, we expose this concept-level information and make it possible for teachers to provide targeted assistance. Furthermore, this encoding makes it straightforward for teachers to determine if multiple students are struggling with the same concept. Teachers may choose to work with these students in small groups or re-teach the concept to the entire class based on this concept-level performance information.

Aggregate Visualization

We designed the aggregate visualization to give teachers a high-level overview of how students are performing during the current lesson. In this visualization, students are sorted alphabetically. For each student, we display a bar showing all of the student's problem-solving information for the current class period. Again, the bar shows check and cross marks to indicate problem correctness. We leave a blank space between the marks for separate concepts to visually separate them. The bar is colored according to the student's average performance during this class period. A screenshot of the aggregate visualization can be seen in Figure 3(a).

The goal of this visual encoding is to allow teachers to assess student's overall performance during the current lesson. In Enlearn's initial studies, teachers found the table summary visualization most useful, so we wanted to create something similar. With this encoding, we make it possible for teachers to identify students who are performing poorly or who have completed fewer problems. If students are performing poorly across the board, teachers may decide to stop problem solving and re-teach lesson content.

Figure 4. The color scales used to denote the percentage of problems students have answered correctly. We use a red-green scale to match common correctness encodings and a red-blue scale for color-blind users.

While the aggregate performance information provides a high-level summary for teachers, we also wanted to make concept-level information accessible through the aggregate visualization. We added an interactive feature to this visualization, allowing the teacher to access concept-level information for a given student by clicking on the student's bar, as shown in Figure 3(b).

Visual Design

In both visualizations, we display one horizontal bar per student that shows problem-solving information. We chose to use the common encoding of check marks and cross marks to denote problem correctness, and display marks in a neutral gray color that stands out against the bar background. Identifying incorrect responses is most important for teachers for the purpose of real-time intervention, so we color crosses in a darker gray than checks to increase their visual significance.

We color the background of each student bar based on average problem correctness. By default, we use a seven-step gradient of colors that ranges from a red to a green with yellow as the midpoint. We chose this encoding because red and green are commonly used to denote correctness in educational settings. While this encoding is the most natural for most teachers, it is not an appropriate gradient for color blind users. To provide support for color blind teachers, we include an option for switching to an orange-blue color gradient if desired. Our color gradients are shown in Figure 4. We selected our these colors using ColorBrewer [5], an application designed to provide a range of easily differentiable color schemes.

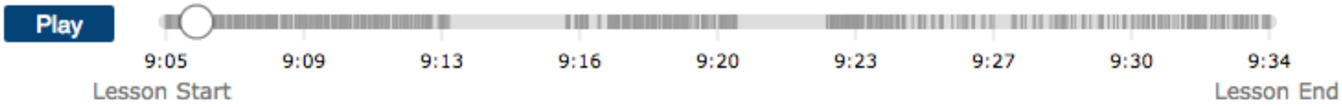


Figure 5. To show how our visualizations would look with real-time data, we simulate a data stream using this timeline slider. Users can play or brush through the data to view the progression of a single lesson. We display lines on the timeline bar to indicate where problem-solving actions occur.

Our visualizations are designed to be used in real-time with streaming data. This means that the displayed data is constantly changing. We use animation in both visualizations to smooth transitions and signify changes. This is particularly important in the concept visualization, since students move dynamically between concepts. We gradually fade student bars in and out when students move between concepts to make these transitions more gradual and visible.

Implementation

In this work, we focused on characterizing the data visualization needs of classroom teachers, developing appropriate data abstractions, and designing visualizations to target teacher needs. Our end goal was not to produce fully functional visualizations that integrate with real-time data streams and run on tablet devices; instead, we wanted to create an environment that Enlearn staff could use to explore and evaluate our designs. We therefore decided to simulate how our visualizations would look on tablet computers with streaming data. This approach will allow us to rapidly iterate on our designs with Enlearn staff before implementing a final product.

We used the D3 visualization library to create our visualizations. Enlearn's current data visualizations are also implemented with D3, so this approach is compatible with their current workflow. In the following sections, we describe how we simulate the tablet interface and streaming data, and discuss the design of our final system.

Tablet Graphic

To simulate how our visualizations would appear on a teacher tablet, we display the visualizations within a tablet graphic on our website. The tablet's display area is restricted to the dimensions of the graphic, and users can scroll as they would on an actual device. We display the concept visualization in a vertically positioned tablet and the aggregate view in a horizontally positioned tablet. The concept view is typically longer than it is wide, while the aggregate view is wider than it is tall. We chose these orientations to maximize the amount of visual information. When the user switches between the two visualizations, we rotate the orientation of the tablet graphic. We use animated transitions to reduce any jarring effect of switching between orientations.

Simulating Streaming Data

To simulate how our visualizations would appear with real-time streaming data, we use a timeline interface to allow users to play and scrub through previously collected data. This approach allows users to both experience the visualizations as they would appear with real-time streaming data and advance to interesting events to quickly assess the visual design. We display lines on the timeline bar to indicate where problem-solving actions occur. These visual information scent cues [17] are designed to help users become familiar with the data set. This is particularly important for our data because there are often long gaps in problem-solving when teachers pause students to clarify a concept or walk through an example. A screenshot of the timeline is shown in Figure 5. While we designed the timeline for our benefit as designers, initial feedback suggests that this type of interaction could be valuable for teachers as well. We discuss this further in Future Work.

Website Design

We designed our final website to display all information in a single screen, and sized content for optimal viewing on a screen with resolution least 1280px by 800px. We display the timeline at the top of the page, and auto-play the simulated data stream on load. Beneath the timeline we display the tablet graphic and the current visualization. To the right-hand side of the screen, we display textual information about the visualization to provide some context for new users. We provide buttons for changing visualization options inline with descriptions about the features of each button. When a user clicks the "Switch View" button, the tablet graphic switches between the concept and aggregate visualizations. The textual information associated with the visualization changes as well. We also provide an option for the user to switch between five different data sets of real class sessions. Finally, we provide an option for switch between our two color schemes. A screenshot of our final website is shown in Figure 6.

RESULTS & DISCUSSION

In this work, we have made contributions at multiple levels of Munzner's nested model of visualization design [14]. We contribute a characterization of the real-time data visualization needs of classroom teachers, describe an appropriate data abstraction designed, and present two new visual encodings of student problem-solving data designed to target teacher needs. Screenshots of our final visualization designs are shown in Figure 6.

The primary goal of this work was to present our final visualization designs in an environment that would allow Enlearn staff and teachers to explore and evaluate the designs before implementing production-level versions of the visualizations for use in the classroom. While we have not formally evaluated our visualization designs yet, we collected informal feedback both from Enlearn staff and from attendees of the CSE 512 poster session. Overall feedback has been positive; Enlearn staff expressed excitement about the potential of these new visual encodings, and poster session attendees liked the visual designs and expressed an interest in the problem space.

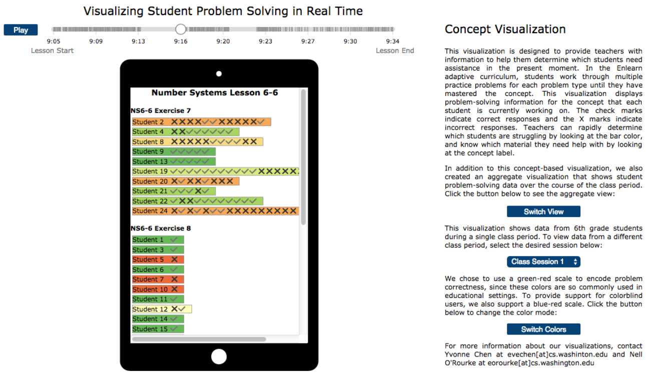
While feedback was generally positive, we received a number of suggestions for improvements and revisions to our designs. Both Enlearn staff and poster session attendees questioned the utility of the aggregate visualization. At the poster session, a number of people suggested that we remove the problem-level checks and crosses, and instead display a grid of colors showing how each student performed on each concept. This would allow teachers to view overall student understanding at the concept level. Interestingly, Enlearn staff gave opposing feedback. The suggested that we color student bars based on recent progress rather than progress across multiple concepts. They reiterated the fact that the primary goal of teachers during class is to rapidly determine which student needs help *right now*, and that they are not interested in viewing summary information during class time. This tension highlights the need for separate visualization designs to target in-class and after-class tasks.

Another suggestion that we heard from both Enlearn staff and poster session attendees is that the concept visualization may not adequately highlight struggling students. One suggested that we add a toggle to hide student who are performing well, so that teachers can focus on the students who need assistance. Another suggested that we sort bars by color to visually group struggling students. An Enlearn staff member also recommended that we wait to color student bars until at least three problems have been completed. In our current design, a single incorrect problem results in a striking red bar even though the student should try a few more times before receiving help. All of these options could improve the ease with which struggling students can be identified.

Finally, a number of people mentioned that the movement of students between concepts in the concept visualization could be jarring. If a teacher is watching the progress of a single student and that student suddenly disappears to another part of the visualization, it could be confusing. At the same time, students only move on to a new concept once they have mastered the previous one, so it is unlikely that struggling students will disappear. However, it is possible that a visualization that provides a persistent view of students but that provides filtering options to help teachers identify struggling students could be most effective.

One surprising result was that Enlearn staff were very excited about the possibility of providing teachers with the timeline interface so that they could review lessons after-the-fact. Staff members are also interested in using this interface to explore data from their studies; they have a mountain of log data from their 10-week trial this fall, but it is difficult to get a sense for how class time was spent and how students performed by looking at problem-solving data in aggregate. While we originally designed this interface to simulate a real-time data stream, it provides an interesting contribution in its own right.

Many of these design questions must be further explored by evaluating the visual designs with teachers, which we aim to address in future work. Despite these suggestions for improvements and extensions to our designs, overall feedback was very positive. At the poster session there was a lot of interest in the problem of visualizing student data; one attendee said that his mom is a middle-school teacher, and that he thinks she would love to have access to this type of real-time student data. Many others were interested in ways that our approaches could be extended to support large lecture classes or MOOCs. This seems like a ripe area for future research.



This visualization is designed to provide teachers with information to help them determine which students need assistance in the present moment. In the Enlearn adaptive curriculum, students work through multiple practice problems for each problem type until they have mastered the concept. This visualization displays problem-solving information for the concept that each student is currently working on. The check marks indicate correct responses and the X marks indicate incorrect responses. Teachers can rapidly determine which students are struggling by looking at the bar color, and know which material they need help with by looking at the concept label.

In addition to this concept-based visualization, we also created an aggregate visualization that shows student problem-solving data over the course of the class period. Click the button below to see the aggregate view:

This visualization shows data from 6th grade students during a single class period. To view data from a different class period, select the desired session below:

We chose to use a green-red scale to encode problem correctness, since these colors are so commonly used in educational settings. To provide support for colorblind users, we also support a blue-red scale. Click the button below to change the color mode:

For more information about our visualizations, contact Yvonne Chen at evechen[at]cs.washinton.edu and Nell O'Rourke at eorourke[at]cs.washington.edu

Figure 6. Our final website, shown displaying the concept visualization.

FUTURE WORK

There are a number of areas for future work. First, we need to evaluate our visualization designs with teachers and iterate on our designs. The informal feedback that we received from Enlearn staff highlights a number of open design questions that are difficult to answer without getting feedback from our target users. We would like to first iterate on our designs with the feedback we have received thus far, and then perform an informal evaluation of our simulated designs with teachers to help us further improve the designs. After these initial iteration cycles, we plan to work with Enlearn to port our D3 visualizations onto their platform so that we can formally evaluate our visualization designs in real classroom settings.

In addition to formally evaluating our real-time designs, we are also interested in exploring some of the alternative visualization designs suggested in the feedback we received. We are excited about the prospect of using the timeline interface to help visualize how class time is spent. In Enlearn classrooms, teachers use a tablet-based version of the curriculum teacher's guide to walk through lessons, example problems, and activities. All of this information is logged to Enlearn's databases. This type of data could be overlaid with the timeline slider to provide a complete picture of how class time is spent and how students perform. This could provide valuable insight for both teachers and researchers.

CONCLUSION

Our work makes multiple contributions to the field of student data visualization. We have identified a set of design guidelines for real-time data visualizations for teachers based on our characterization of the problem space. These guidelines focus on the importance of providing actionable information for teachers and presenting data at both the student and concept level. We characterize the operations that need to be performed on problem-solving data to address teacher needs, specifically *sorting* and *computing* average performance. Finally, we present two new visual encodings for student data, one that displays information for the concepts students are working on in the present moment, and another that displays aggregate performance across multiple concepts.

REFERENCES

1. Robert Amar, James Eagan, and John Stasko. 2005. Low-Level Components of Analytic Activity in Information Visualization. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization (INFOVIS '05)*.
2. Richard Anderson, Ruth Anderson, K. M. Davis, Natalie Linnell, Craig Prince, and Valentin Razmov. 2007. Supporting Active Learning and Example Based Instruction with Classroom Technology. *SIGCSE Bull.* 39, 1 (March 2007), 69–73.
3. Kimberly E. Arnold and Matthew D. Pistilli. 2012. Course Signals at Purdue: Using Learning Analytics to Increase Student Success. In *Proceedings of the 2Nd International Conference on Learning Analytics and Knowledge (LAK '12)*. 267–270.
4. Madeline Balaam, Geraldine Fitzpatrick, Judith Good, and Rosemary Luckin. 2010. Exploring Affective Technologies for the Classroom with the Subtle Stone. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. 1623–1632.
5. Cynthia Brewer. 2015. ColorBrewer. (2015). `http://www.colorbrewer2.org/`
6. H. L. Dangel and C. X. Wang. 2008. Student response systems in higher education: Moving beyond linear teaching and surface learning. *Journal of Educational Technology Development and Exchange* 1, 1 (2008), 93–104.
7. G. Gibbs and C Simpson. 2004. Conditions under which assessment supports students' learning. *Learning and Teaching in Higher Education* 1, 1 (2004), 3–31.
8. Timothy J. Hickey and William T. Tarimo. 2014. The Affective Tutor. *J. Comput. Sci. Coll.* 29, 6 (June 2014), 50–56.
9. Heather C. Hill, Brian Rowan, and Deborah Loewenberg Ball. 2002. Effects of Teachers? Mathematical Knowledge for Teaching on Student Achievement. *American Educational Research Journal* 42, 2 (2002), 371–406.
10. Juho Kim, Elena L. Glassman, Andrés Monroy-Hernández, and Meredith Ringel Morris. 2015. RIMES: Embedding Interactive Multimedia Exercises in Lecture Videos. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. 1535–1544.
11. Kimberle Koile and David Singer. 2006. Improving Learning in CS1 via tablet-PC-based In-class Assessment. In *Proceedings of the Second International Workshop on Computing Education Research (ICER '06)*. 119–126.
12. Alina Lazar. 2007. Engaged Learning in a Computer Science Course. *J. Comput. Sci. Coll.* 23, 1 (Oct. 2007), 38–44.
13. JUMP Math. 2015. JUMP Math. (2015). `http://jumpmath.org/`
14. Tamara Munzner. 2009. A Nested Model for Visualization Design and Validation. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (Nov. 2009), 921–928.
15. Plickers. 2015. Plickers. (2015). `https://www.plickers.com/`
16. M Steadman. 1998. Using classroom assessment to change both teaching and learning. *New Directions for Teaching and Learning* 75 (1998), 23–35.
17. Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. 2007. Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (Nov. 2007), 1129–1136.
18. S. Paul Wright, William L. Sanders, and Sandra P. Horn. 1997. Teacher and Classroom Context Effects on Student Achievement: Implications for Teacher Evaluation. *Journal of Personnel Evaluation in Education* 11, 1 (1997), 57–67.
19. Meilan Zhang, Robert Trussell, Benjamin Gallegos, and Rasmiyeh Asam. 2015. Using Math Apps for Improving Student Learning: An Exploratory Study in an Inclusive Fourth Grade Classroom. *TechTrends: Linking Research & Practice to Improve Learning* 59, 2 (2015), 32 – 39.