

Visualizing Student Problem-Solving Data

Yvonne Chen
evechen@uw.edu

Eleanor O'Rourke
eorourke@cs.washington.edu

ABSTRACT

Abstract goes here

INTRODUCTION

An explanation of the problem and the motivation for solving it.

RELATED WORK

A large body of research has explored methods of integrating technology into in-person classrooms and using technology to deliver education online. We review this research focusing on technology-delivered curriculums, systems that expose student data for teachers, and adaptive learning environments.

Exposing Student Data for Teachers

Education research shows that teacher behavior has a strong impact on student achievement [6, 13, 12, 14], and that teachers can benefit from the availability of real-time student data [3, 8, 9]. 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 [8]. With access to real-time data, research suggests that instructors can intervene during a lesson when students are confused [5], alter the pace or content of instruction based on student engagement [3], immediately identify and assist students who are struggling [9], and choose topics of focus based on aggregates of student responses [8].

A number of technologies have been developed for exposing student data for teachers. One technology that is often used in lecture-based classes is the “student response system” or “clicker,” which is used to poll students on multiple-choice questions during class [4, 9]. A similar application designed for small classes is Plickers [11]. With the Plickers smartphone app, the teacher can scan the classroom while students hold up QR codes identifying a multiple-choice answer [11]. Researchers have also explored methods of providing instructors with access to student data outside of instruction time to monitor longer term academic progress [15, 2]. 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 [7].

In this work, we explore methods of exposing rich problem-solving data to elementary school teachers in real-time. Through a longitudinal study, we explore how teacher behavior is impacted by exposing information about student misconnects and progress through curriculum material.

Whether viewed in real time or after the fact, teachers benefit from the availability of student data. Accessed in real time, student data affords immediate intervention. In a study by Koile, students used Tablet PCs to complete exercise problems during class time. The instructor could access their answers right away on her own device. Based on the students’ answers, the instructor devoted 75% of course time responding to misunderstandings of course material, and would delay or hasten presentation of new material as appropriate [8]. With real time student data, instructors can intervene during a lesson when students are confused [5], alter the pace or content of instruction based on student engagement [3], immediately identify and assist students who are struggling [9], and choose topics of focus based on aggregates of student responses [8].

When timely intervention is less critical, instructors can view data outside of instruction time to monitor longer term academic progress [15, 2]. Kim et al developed an online system for teachers to author video lectures containing integrated multimedia exercises. The system compiled student responses as they completed exercises, and teachers reported that they were useful for “capturing students thought processes, identifying misconceptions, and engaging students with content” [7]. Even basic classroom response systems, which poll students to choose one out of a set of pre-defined responses, give teachers a quick aggregate overview of student understanding [9].

While research shows that student data can help teachers understand misconceptions and how they spend class time, very little is known about how to visualize student data to most effectively communicate with teachers. We have not encountered any papers that study the design of visualizations of student data. Enlearn has implemented two different methods of visualizing student data in real-time for teachers, however, both of their designs have had usability and readability problems when tested in classrooms. This shows how challenging it is to create effective visualizations for communicating student data in real-time. Enlearn’s first design, shown in Figure 1a, displays a table view of student problem solving pace and correctness. While this visualization provides information about the progress of individual students, it communicates nothing about the specific concepts that students are struggling with. Enlearn’s second design, shown in Figure 1b, displays a graph view of student problem solving that is organized by concept. While this displays information about the concepts that students are struggling with, there is no way

Paste the appropriate copyright statement here. ACM now supports three different copyright statements:

- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single spaced.

Every submission will be assigned their own unique DOI string to be included here.

for teachers to see which individuals are struggling. Most importantly, neither visualization provides teachers with actionable suggestions about how to assist students who are struggling. Prior work shows the utility of communicating student data for instructors, and Enlearn’s experiences with their current visualizations shows the challenge of designing effective means of communicating data, especially in real time for busy teachers. This motivates our desire to improve and strengthen the data visualization for the Enlearn software.

ENLEARN

For this project, we partnered with Enlearn, a non-profit company founded to develop tablet-based adaptive problem-solving software for K-12 classrooms. In our visualizations, we display real data from Enlearn students.

Enlearn evaluated two visualizations with teachers this spring. One displays student data in a table view, but provides no concept-level information. The other displays a concept graph, but provides no data at the level of individual students.

Enlearn found that neither visualization was effective in the classroom, in large part because the information provided did not match teacher needs. Another challenge is that teachers are extremely busy during class time and need visualizations that are both easily glanceable and that provide actionable information.

METHODS

To address the challenge of designing effective techniques for visualizing student problem-solving data for teachers, we first characterize the problem, then discuss our data abstractions, present our visual encodings for conveying the data, and 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 to a database: 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. The Enlearn adaptive curriculum is designed as an extension of the JUMP Math curriculum [10]. 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 them 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 choose to re-teach parts of the lesson.

While Enlearn’s initial visualizations tried to target these tasks, they did not provide adequate information for teachers to be able to quickly determine which students were struggling the most and what concepts they were currently working on. 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 adapting 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 focus our data abstraction and visual encoding designs.

Data Abstractions

With our improved understanding of Enlearn’s data structure and teacher’s 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 Skasko [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. And finally, to be able 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 compute averages and sort students automatically.

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 computation 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

Simulating Real-Time Data

Our visualizations are designed to display real-time student data; however, we do not have a constant stream of data to display. To simulate real-time problem solving, we use data collected from students last fall. We display a timeline that can be played and brushed, allowing both us and Enlearn staff to quickly see what the visualizations would look like in various real-time scenarios.

Concept View

The concept visualization is designed to help teachers determine which students need help in the present moment. Students are grouped by the concept that they are currently working on so that it is easy for the teacher to know what concept to target. The background color of each student bar indicates how well the student is performing on that concept. This view gives teachers a glanceable display that can help them rapidly determine which student to assist.

Aggregated View

The aggregate visualization is designed to give teachers a high-level overview of how each student has performed over the course of the lesson. This view displays all problem-solving information for each student across multiple concepts. The background is colored to show the student's aggregate performance. If the teacher wants to see concept-level information for a given student, she can click on that student to see the individual visualization shown to the right above.

Visualization Options

Tablet Graphic

We added a tablet graphic as a mockup for intended use of our visualization. The tablet's display area is restricted, but users can scroll within the tablet screen to navigate. To prevent confusion with multiple sets of scrollbars, we designed our

visualization to fit on a 13" MacBook Pro screen. The horizontal view and aggregate views are shown in a vertical and horizontal tablet layout, respectively. The concept view tends to grow vertically faster than it grows horizontally, since it only displays problem information for the student's current concept, whereas the aggregate view tends to grow horizontally much faster than it grows vertically, because there is a fixed number of rows based on the number of students in the class.

Correctness Glyphs

Enlearn's initial visualization used green and red filled circles to show the correctness of a single problem. However, these color choices conflict with red and green backgrounds used to denote student progress. We instead choose the common encodings of checks and crosses in a neutral gray. For the purposes of real-time intervention, student problem incorrectness is more useful to a teacher than problem correctness, so we color crosses slightly darker than checks to increase their visual significance.

Correctness Colors

The default 7-step gradient of colors to represent student problem correctness ranges from a muted orange to a muted green, with yellow as the midpoint - these colors were selected from ColorBrewer2.org, a site created to help designers select easily differentiable color schemes. We chose an orange-green spectrum because red and green are very common encodings for correctness in the educational field - however, we do realize that some users of our tool may be color-blind, so we also include an option to switch to an orange-blue color gradient.

Button Layout

To provide context to our visualization, the right side of our visualization contains information on the currently selected view, as well as the buttons for various interaction options: switching the view, switching the class session, and switching colors. These buttons are displayed inline with brief descriptions of their purpose.

Timeline

We display lines on the timeline where problem-solving actions occurred. These visual information scent cues [?] help to familiarize the viewer with the data set. While we designed this feature for our benefit as designers, it may be valuable when used by teachers to play back their lessons.

Animation

We use animation to signify changes. When the user clicks the button to switch views, the tablet and information box about the current view fade out, and then the new view fades in. In the concept view, student movement to a new concept is shown by a fade out-fade in transition, leaving a small gap amongst the students in their previous concept to highlight the student's prior location. To prevent viewers from becoming overwhelmed by too much visual movement, check and cross marks that signify problem correctness simply pop into existence.

Data Choices

Our system provides a selection of 5 datasets that the user can

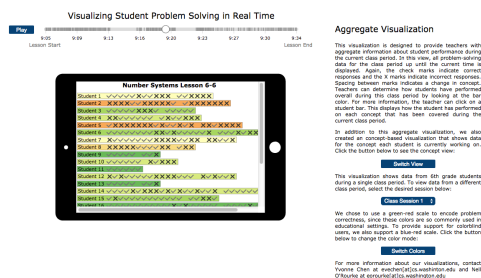


Figure 1. Caption here

switch between freely. The datasets all come from classes taught by a single teacher in a South Seattle school. We envision a teacher using our tool to replay and compare the progress of different class periods, or the same class period over different lessons.

Implementation

Enlearn's current visualizations are created using the D3.js visualization library. Hence, we also use D3.js, not only to be compatible within Enlearn's existing workflow, but also to prepare a foundation for a future web-based tool so that teachers will not be dependent on the Enlearn tablet software and can view student data from any computer.

RESULTS & DISCUSSION

Our final product is a visualization that integrates our concept view and aggregate views, playing back student progress over a math lesson via a timeline slider that supports brushing. Prior work displays student data at an assignment level, which does not afford immediate intervention in the classroom (may be a wild claim, must confirm). Our system simulates real-time, problem level visualizations that teachers can use to track student problem-solving progress. Through the concept view, teachers can immediately see which students need assistance and what to assist them with, and through the aggregate view, teachers can monitor longer term student progress over the course of a lesson.

On a more general level, our work is not limited to playing back data gathered from Enlearn math tablets. Any data set with information about problem time completion, problem correctness and problem concept grouping could be visualized using our tool.

Overall, attendees of the CSE 512 project poster session reacted positively, commenting that our system was visually clean and easy to understand. One viewer mentioned that he taught middle school students and would find our visualization useful for tracking their progress.

FUTURE WORK

As described above, the visual information scent cues overlaid on our timeline slider help inform viewers of gaps where students were not completing tablet problems. The teacher version of Enlearn's software collects data on teacher actions, such as sending a pause signal to freeze all student tablets to get their attention, or switching between lessons. Such information could be overlaid with the timeline slider as insight to

teachers. For example, this feature could show that during a time when the teacher is talking, some problem solving activity is still present. This means some students are off task and using their tablets instead of paying attention to their teacher. However, there is currently no feature to tell which student is associated with which tick marks. Calling out a student's associated tick marks when their name is clicked in the aggregate view would help teachers to monitor an individual student's activity patterns.

Multiple visitors to the poster session that our aggregate view was not as easily glanceable as the concept view. They all independently suggested that information about each student's progress over a single concept could be collapsed into a single representative glyph, such as a shape colored to represent correctness. This way, all students' data could be displayed in a matrix, with pertinent information readily visible at a glance and detailed information accessible by tapping on the student's name.

Another comment received during the poster session was that our animation transitions could be clearer to help teachers track the movement of students between concepts.

Although the above suggestions have potential to improve our work, an important next step is to evaluate our system with actual teachers. Enlearn has a preexisting relationship with Seattle school teachers, with whom they have collected tablet data over multiple trials. Since Enlearn's current visualizations are also created in D3.js, we can port our system into their software, to be evaluated in future classroom trials. Suggestions from the target audience are invaluable to substantial improvements to our work.

CONCLUSION

Our work makes contributions at multiple levels of Munzner's nested model for visualization design: a characterization of the data visualization needs of classroom teachers, a data abstraction for representing student data grouped by concept, and two new visual encodings for displaying student problem-solving data both streamed in real time and played back after the fact. Our system empowers teachers to easily identify students for real-time assistance, track lesson-level metrics and monitor student progress over multiple lessons.

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)*. IEEE Computer Society, Washington, DC, USA, 15–. DOI : <http://dx.doi.org/10.1109/INFOVIS.2005.24>
2. 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)*. ACM, New York, NY, USA, 267–270. DOI : <http://dx.doi.org/10.1145/2330601.2330666>

3. 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)*. ACM, New York, NY, USA, 1623–1632. DOI : <http://dx.doi.org/10.1145/1753326.1753568>
4. 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.
5. Timothy J. Hickey and William T. Tarimo. 2014. The Affective Tutor. *J. Comput. Sci. Coll.* 29, 6 (June 2014), 50–56. <http://dl.acm.org/citation.cfm?id=2602724.2602735>
6. 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. <http://aer.sagepub.com/content/42/2/371.full.pdf+html>
7. 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)*. ACM, New York, NY, USA, 1535–1544. DOI : <http://dx.doi.org/10.1145/2702123.2702186>
8. 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)*. ACM, New York, NY, USA, 119–126. DOI : <http://dx.doi.org/10.1145/1151588.1151607>
9. Alina Lazar. 2007. Engaged Learning in a Computer Science Course. *J. Comput. Sci. Coll.* 23, 1 (Oct. 2007), 38–44. <http://dl.acm.org/citation.cfm?id=1289280.1289288>
10. JUMP Math. 2015. JUMP Math. (2015). <http://jumpmath.org/>
11. Plickers. 2015. Plickers. (2015). <https://www.plickers.com/> [Online; accessed 19-May-2015].
12. Johnmarshall Reeve, Hyungshim Jang, Dan Carrell, Soohyun Jeon, and Jon Barch. 2004. Enhancing Students' Engagement by Increasing Teachers' Autonomy Support. *Motivation and Emotion* 28, 2 (2004), 147–169. DOI : <http://dx.doi.org/10.1023/B:MOEM.0000032312.95499.6f>
13. Kathryn R. Wentzel. 2002. Are Effective Teachers Like Good Parents? Teaching Styles and Student Adjustment in Early Adolescence. *Child Development* 73, 1 (Jan. 2002), 287–301. <http://onlinelibrary.wiley.com/store/10.1111/1467-8624.00406/asset/1467-8624.00406.pdf?v=1&t=i9yfhr3p&s=a0e68e6a2d2c2bf6c579e860419e72d88966d6ab>
14. 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. DOI : <http://dx.doi.org/10.1023/A:1007999204543>
15. 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. <http://offcampus.lib.washington.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=eue&AN=100711551&site=ehost-live>