

Interaction History for Classroom Learning Partner

Casey McNamara

Advisor: Dr. Kimberle Koile, MIT Center for Educational Computing Initiatives

24 May 2013

Introduction

Traditionally, elementary school students practice applying math and science concepts by completing paper worksheets. Their teacher devises problems that will let the students hone and display their skills, the students work the problems, and the teacher collects and looks over the results to see how much the students understand. Classroom Learning Partner, a piece of software for tablet PCs in ongoing development by Kimberle Koile and her research group, is intended as a more technologically sophisticated version of this system, providing useful features beyond those of a paper worksheet to help students learn and educators teach. [1, 2]

As my Undergraduate Advanced Project, I contributed an interaction history feature to the Classroom Learning Partner software. My feature records the sequence of actions students take while working through problems. The existence of this record has a number of useful consequences: students can undo and redo their actions, enhancing usability from their point of view; teachers can replay a student's work on a page to gain a better understanding of how the student arrived at the final answer; and the software itself could in theory do some automated analysis of the history.

Background

The Classroom Learning Partner software runs on tablet PCs. Each student has a tablet running the software, which allows the user to edit a notebook consisting of

several pages. At any one time, a student sees a single page containing one or more problems and space to solve them. To work the problems, the student can write or draw with the tablet's pen. The software also has a number of more specialized editing tools, including stamps, which allow a student to draw an image once and make many copies of it, in order to reason concretely about numbers; and arrays, which have two versions: one that is just a rectangle with edges labeled by their length and a grid inside, in order to help students understand multiplication in general, and one that allows a student to create horizontal and vertical lines subdividing the array, in order help students understand the distributive property of multiplication.

The teacher has a tablet running a slightly different version of the software. The student tablets are connected to the teacher's over a local network. Students can send their work on a particular page to the teacher over the network. Once the students' submissions are on the teacher's machine, the teacher can show the submissions on a projector for class discussion or look them over later.

My new history feature enables students to undo and redo their own actions and replay a movie of all their actions on the page they're currently viewing. When a student sends a page to the teacher, the page's history travels across the network with it, so the teacher can review a student's history for a page or even play it back on the projector for the whole class to see. The history also saves along with the notebook, so going home for the day does not destroy the history.

Implementation

I had initially planned to take an existing history feature and make a history analysis feature that would automatically determine how difficult the student seemed to find the problem. It turned out that Claire DeRosa had created a history feature for her

May 2012 MEng thesis, but it was not compatible with the current CLP software because it took too much space to store the histories and because parts of the software her history system relied on had been redesigned. So I set about making a history feature that would take as little space as possible and be easily extensible to handle future changes in the design of the software.

I implemented the history with two stacks containing history items: events in the history that can be individually undone and redone. One stack stores the past, the other the future. Pressing undo pops an event from the past, calls its undo method, and pushes it to the future. Pressing redo pops an event from the future, calls its redo method, and pushes it to the past. A third stack, called the metapast, stores the entire history including the effects of undos and redos. Pressing replay undoes all events in the metapast and then redoes them one by one with suitable time delays in between so that a human observer can see the history. Replay uses the metapast rather than the past because if a student uses the undo button to reverse some mistake, the teacher viewing the history should be able to see the path the student initially went down.

My first design challenge was to make the history system extensible—I didn't want to make another system that would be obsolete in a year. To that end, I made an abstract class, `CLPHistoryItem`. This class has two main methods: `undo` and `redo`. The system for undoing, redoing, and replaying events deals only with these generic methods. There are several subclasses of `CLPHistoryItem`, such as `CLPHistoryAddObject`, which represents a stamp, array, or image being added to the page; or `CLPHistoryMoveObject`, which represents such a thing being dragged across the page. Each of these subclasses store whatever data they need to store in order to undo or replicate the action they represent and implement the `undo` and `redo` methods. I slightly modified event handlers in the CLP software that already trigger when an

object is added or moved to also create a history item storing the appropriate information and record it in the history.

Under this design framework, when a new type of action is added into the software to enable some new pedagogical technique, it will only be necessary to add a new subclass to represent it. While I was working on the history system, Liz Simon, another of Kimberle Koile's students, implemented a feature that allowed students to add dividing lines to arrays; this interaction was not picked up by the global event handlers because she implemented the dividing lines by modifying the array rather than by adding new objects to the page, but it was quite simple to add new subclasses for adding and removing array dividers and to make the event handlers notify the history appropriately that an array divider had been added or removed.

My second design challenge was to make the histories small. Because the pages need to get sent over the network to the teacher's tablet fairly quickly in order for students to discuss in class the work they've just submitted, there are substantial constraints on the amount of data the history can store. My initial implementation stored, with each history item, the object on the page that it affected: for example, a history item representing moving a stamp would store the stamp along with the parameters of the movement. This scheme worked fine in basic testing but did not hold up to students using the software for a whole hour of class time.

I therefore pared down the history items to store only a unique ID of the object. When an undo or redo affecting the object takes place, the software searches through the objects on the page for one with that ID, trading a bit of runtime speed for a significant gain in space-efficiency. The one exception was history items representing the removal of objects; these items cannot just store the ID of the object and look through the current page for it because the object no longer exists on the current page,

so they still store the object itself. The new history items, storing only the data they absolutely need, seem to be small enough: when a student sends a page to the teacher, history included, it arrives without noticeable delay compared to the situation before the history implementation.

Results

By playing back the history of how a student arrived at an answer, a teacher can learn how a student arrived at that answer. We took the tablets into a local special-education 4th and 5th grade classroom to see heuristically how having the history available could be useful.

We expected that students might use the undo and redo features to help them work more efficiently. I didn't observe any students doing this, perhaps because they had used the tablets before and were already accustomed to the workflow of using the tablet pen's eraser to erase mistakes. However, Kimberle Koile commented that having the undo button had been useful to her when writing the problems to give to the students: she'd accidentally deleted the text box containing the instructions and was able to undo the deletion rather than retyping the whole problem.

We expected that having the history could help determine how easy students found the problems, whether they could give the right answer immediately or spent time trying different possible approaches. Replaying the histories we collected indeed allowed me to make subjective judgments of this. On a problem about identifying coins, for example, one student gave all the correct answers on the page she submitted as her final answer, but replaying reveals that she initially had some of the coin values misidentified. (See Figure 1.)

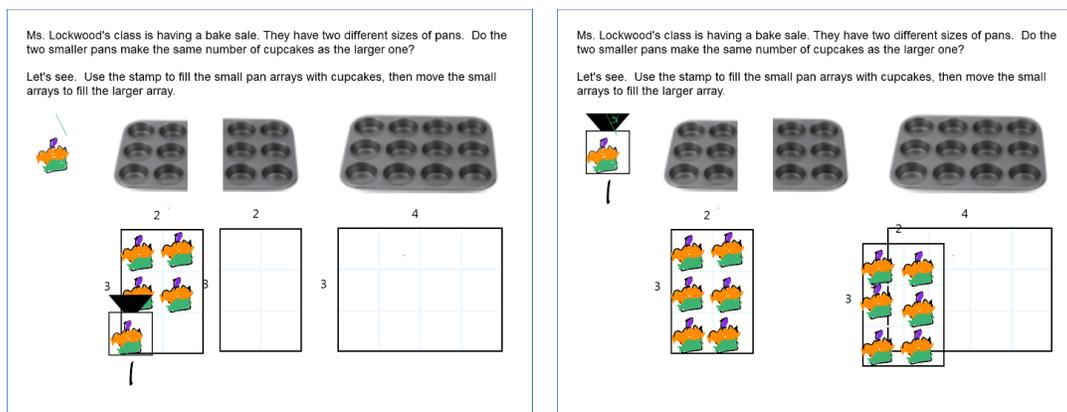
| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1. Let's review coins.</p> <p>Fill in the blanks with the name of each coin and how many cents it is worth.</p> <p>This coin  is a <u>Penny</u>. It is worth <u>1¢</u>.</p> <p>This coin  is a _____ . It is worth _____ .</p> <p>This coin  is a <u>Dime</u>. It is worth <u>25¢</u>.</p> <p>This coin  is a <u>Quarter</u>. It is worth <u>5¢</u>.</p> | <p>1. Let's review coins.</p> <p>Fill in the blanks with the name of each coin and how many cents it is worth.</p> <p>This coin  is a <u>Penny</u>. It is worth <u>1¢</u>.</p> <p>This coin  is a <u>Nickel</u>. It is worth <u>5¢</u>.</p> <p>This coin  is a <u>Dime</u>. It is worth <u>10¢</u>.</p> <p>This coin  is a <u>Quarter</u>. It is worth <u>25¢</u>.</p> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Figure 1. a. Partway through the replay

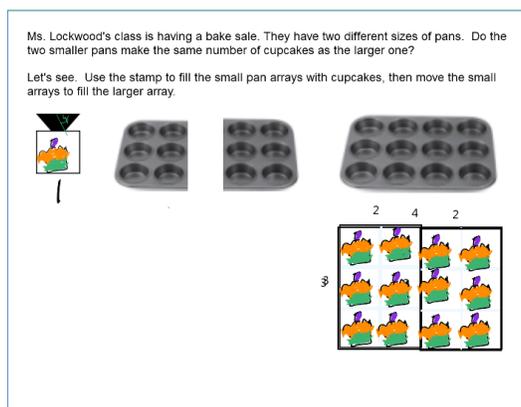
b. Final solution

We also expected that having a record of history would allow for problems where the desired solution is not a static picture, but rather the animation of the process by which it was created. Timothy Mwangi, for his MEng thesis, is looking into making an animation-creation feature for teaching students to construct visual mathematical arguments; history replay will be a major part of such a feature.

We tried out some simple process problems with the special-education 4th and 5th graders and were able to play back their responses. In one problem, students were asked to stamp cupcakes into two small rectangular arrays representing cupcake tins, then drag the two small arrays (each 3 squares by 2 squares) onto a larger one (3 by 4) to show that $3 \times 2 + 3 \times 2 = 3 \times 4$. See Figure 2. for some snapshots from the replay of one student's history on this problem.



a. Stamping cupcakes into smaller arrays b. Moving the small arrays into the larger one



c. Final state, with the process complete

Figure 2. Student work showing movement of smaller arrays onto larger one to show equivalence

This student's history on this problem is not particularly surprising, because the instructions walked the students through the process they were meant to show. One could imagine telling students comfortable working more independently, however, to "show visually that $3 \times 2 + 3 \times 2 = 3 \times 4$," giving them the latitude to decide how to show it, and replaying the histories to view the students' solutions.

Finally, we expected that having histories would enable teachers to see where in the process of solving a problem a student went wrong, even if the student erased his work. The special-education classroom was not ideal for gathering such examples, because the students were not generally expected to independently complete multistep problems of the sort where one can do the work for a subproblem, erase it, and move on, but there were a few such problems having to do with computing change.

One problem asked students to compute the total cost of two items ($\$2.35 + \$1.25 = \$3.60$) and then the change one would get after paying for the items with $\$4.00$. A student gave the correct answer for the total cost and an incorrect answer of $\$1.60$ for the change; he showed the subtraction problem he did to get $\$1.60$, but playing back his history for the problem reveals further details. First, he also wrote out the addition problem he did to get $\$3.60$, but later erased it, indicating he did the correct addition longhand rather than doing it in his head or getting the answer from somewhere else. Second, when doing the subtraction problem he first wrote down $\$7.60$ —which would be correct if it were an addition problem—then changed the 7 to a 1. (See Figure 3.) These details confirm that the student understands how to do multidigit addition but does not really understand how the algorithm for multidigit subtraction differs from that for addition or when to apply subtraction instead of addition.

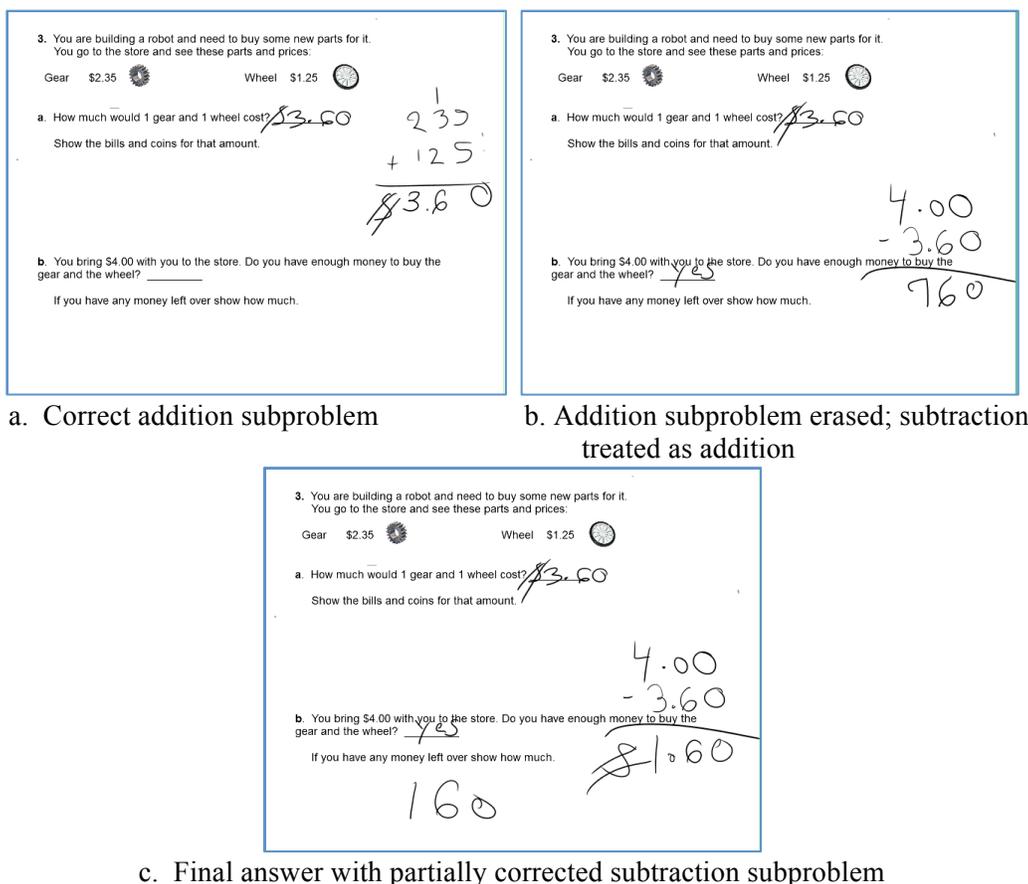


Figure 3. Screen shots from replay of student work in computing total money and change

Future Extensions

I'd originally planned to make a feature that would automatically analyze interaction history for values such as time spent and amount of erasing in order to estimate how difficult a student found a problem, and this analysis could still be useful. In working with actual histories, however, I've become increasingly convinced that automated analysis would not make such a subjective overall assessment in a way teachers would find consistently useful. Instead, I propose a few less glamorously analytical but perhaps more helpful extensions.

The replay feature is currently very basic: it undoes back to the beginning and replays at a fixed pace through to the end. This limitation makes watching a student's history a time-consuming undertaking, even if one only cares about a particular part. It would be nice to add a more complex video-playback UI to the replay feature, so that a user could pause the replay, speed it up or slow it down, or jump to a particular position. This functionality would be useful to teachers wanting to use the replay feature to go over students' work without wasting too much time watching irrelevant parts of the movie; it also would be useful to students replaying their own work, perhaps as part of constructing an animation to explain a mathematical argument.

It's not clear from viewing the final version of a student's page whether anything interesting can be learned by replaying it, and replaying, even with fast-forward, takes time. To save teachers' time, I would add an "unerase all" button that would selectively undo all removal events. If a student went straight to writing the final answer, this button would not change the page at all; if a student erased and rewrote several times, the button would make the page look quite cluttered, indicating that a replay might be informative.

Overall, the history feature performed well in the classroom. It shows promise as a tool for providing teachers with feedback about how their students arrived at particular answers, and thus, will help teachers gain insight into what their students do and do not know.

References

- [1] Koile, K. and Rubin, A. Machine interpretation of students' hand-drawn mathematical representations. In *Proceedings of WIPTTE (Workshop on the Impact of Pen and Touch Technology on Education)*, 2013.
- [2] Rubin, A., Storeygard, J., and Koile, K. Supporting special needs students in drawing mathematical representations. In *Proceedings of WIPTTE (Workshop on the Impact of Pen and Touch Technology on Education)*, 2013.