Data Storage and Viewing for Classroom Learning Partner

Jessie Mueller
Massachusetts Institute of Technology
Supervisor: Dr. Kimberle Koile, Center for Educational Computing Initiatives
May 12, 2011

1 Introduction

Researchers and educators have been interested in using new technologies in the classroom for many years. There are programs to give laptops to every student and create forums in which to continue discussions outside of the classroom. These programs strive to improve student learning, but often have no mechanism for investigating how, when the results come in, results came about. This situation prevents educators from understanding how to best use these technologies and diminishes research returns. This report describes a data storage and viewing system for a classroom interaction system called Classroom Learning Partner (CLP). The data storage and viewing system will allow teachers to easily track student work and help to illuminate CLP's effect in the classroom.

2 Background

Background Overview 2.1

Classroom Learning Partner (CLP) is a tablet-pc software system with a pen based interface enabling student-teacher interaction. It was originally developed by Dr. Kimberle Koile and her research team for use in undergraduate computer science courses (Koile et. al 2007a, Koile et. al 2007b). CLP is based on the believe that both teachers and students benefit from review and discussion of student handwritten and drawn work during class. based on a project that allows student to submit answers to questions in class. Currently, a new version of CLP is being developed for use in elementary school math and science education, funded by a four year NSF project called INK-12 (Koile, K. and Rubin, A, 2010). Using CLP, students write and draw answers to questions and submit them to the teacher.

2.2 System Components

CLP runs on a set of wirelessly connected of tablet computers, which take on one of three distinct functions, depending on the mode in which the software is run. Student have tablets on which they write their work, save it, and submit it to the teach. The interaction metaphor is that of an electronic notebook: students write their work on notebook pages, which the teacher has created using the CLP authoring tool. The teacher's tablet receives and presents the submissions to the teacher, and he or she can respond back to students individually. The teacher also can choose a subset of student submissions to send it to the public tablet. The public tablet is connected to a projector, and displays the student work anonymously it to the whole class for discussion. A photo of a classroom discussion using CLP is shown in Figure 1, and a close up on the student interface is shown in Figure 2.

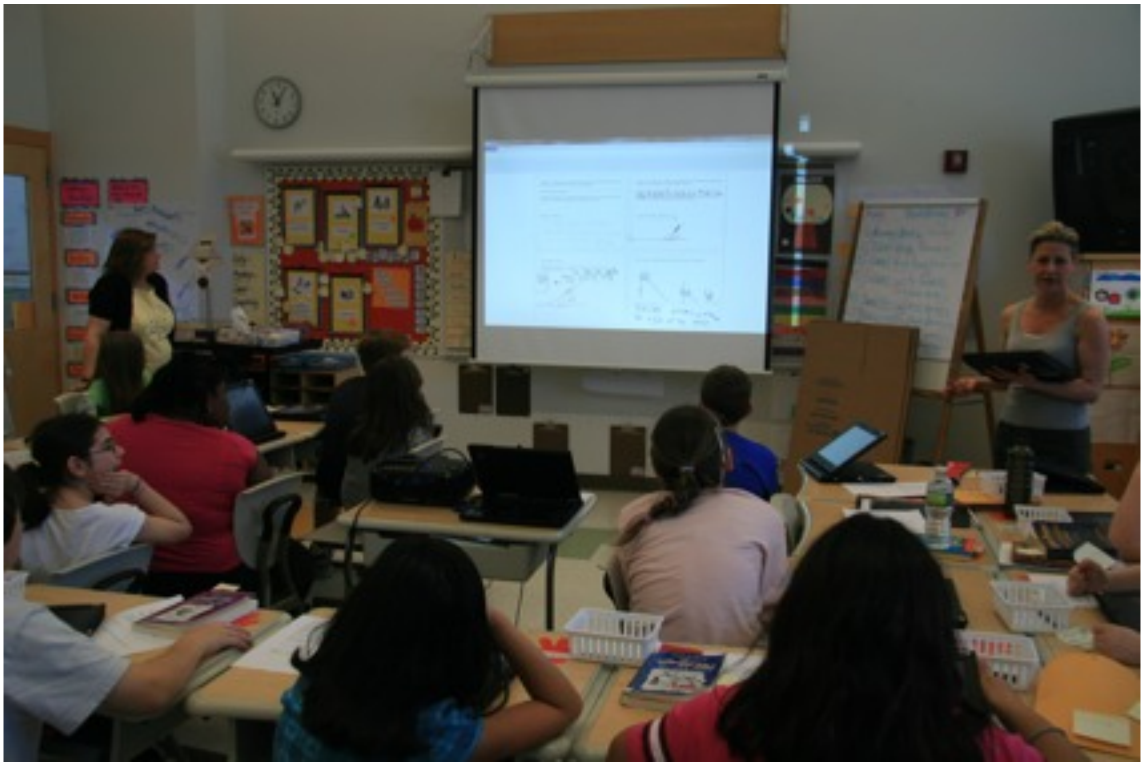

Figure 1: CLP use in a forth grade classroom. The teacher (on the right) receives work from student tablets and controls the projected display from her tablet.

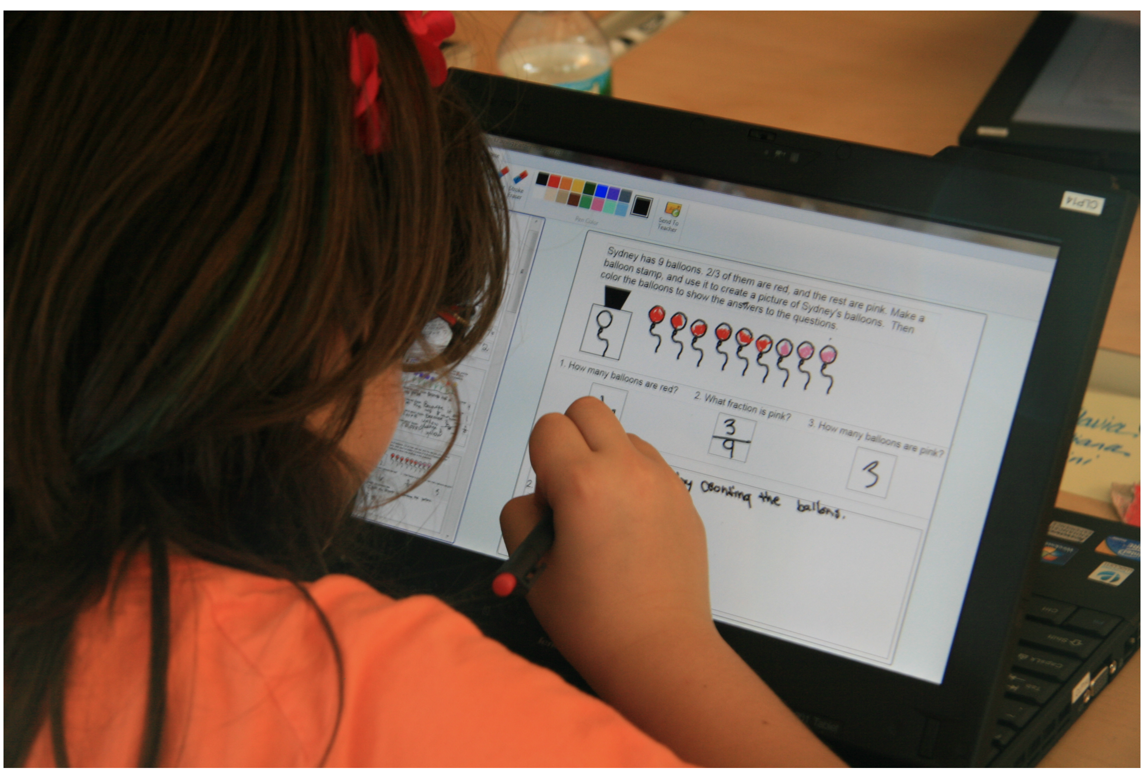

Figure 2: Student working on tablet. The CLP software enables student to submit numerical, written, and drawn work and explanations.

2.3 Type of Submissions

Student work is currently implemented in CLP in two ways. All written or drawn strokes are represented with a Stroke class, which stores information about location, time, and color of the stroke. On submission, all the drawn/written strokes on the page are compressed and represented in a string, which is wirelessly transmitted to the teacher tablet. Additionally, when an expected student answer is a graph or sketch, an invisible gird on the student notebook page captures the ink strokes and discretizes them for analysis. (See (Chao 2011) for more detail.)

This ink analysis is important for various fraction and math lessons prevalent in elementary school curricula. Figure 3 shows an example problem.

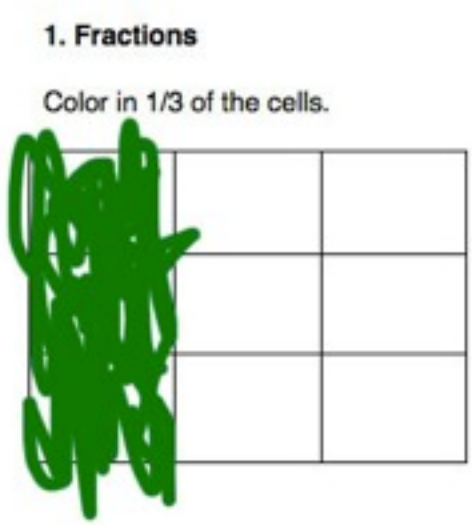

Figure 3: Example fraction problem

The CLP research team is implementing methods for interpreting of students' stokes, which will benefit the teacher by allowing submissions to be sorted. Areas of interest include interpreting shape, as well as deciding if a problem, with a numeric or graphical solution, is correct. Future work will feed results from such a preprocessor into the UI display system and database.

2.4 Organization of Submissions

Teachers can view and respond to student submissions and send student work, or their own lessons, to the projector. The submission-viewing interface shown in Figure 4 allows the teacher to organize student work based on some category. (See (Jozwiak 2011) for more detail.) Categorization is currently done by the teacher; however if the future it will also be based on the results of the AI-preprocessor.

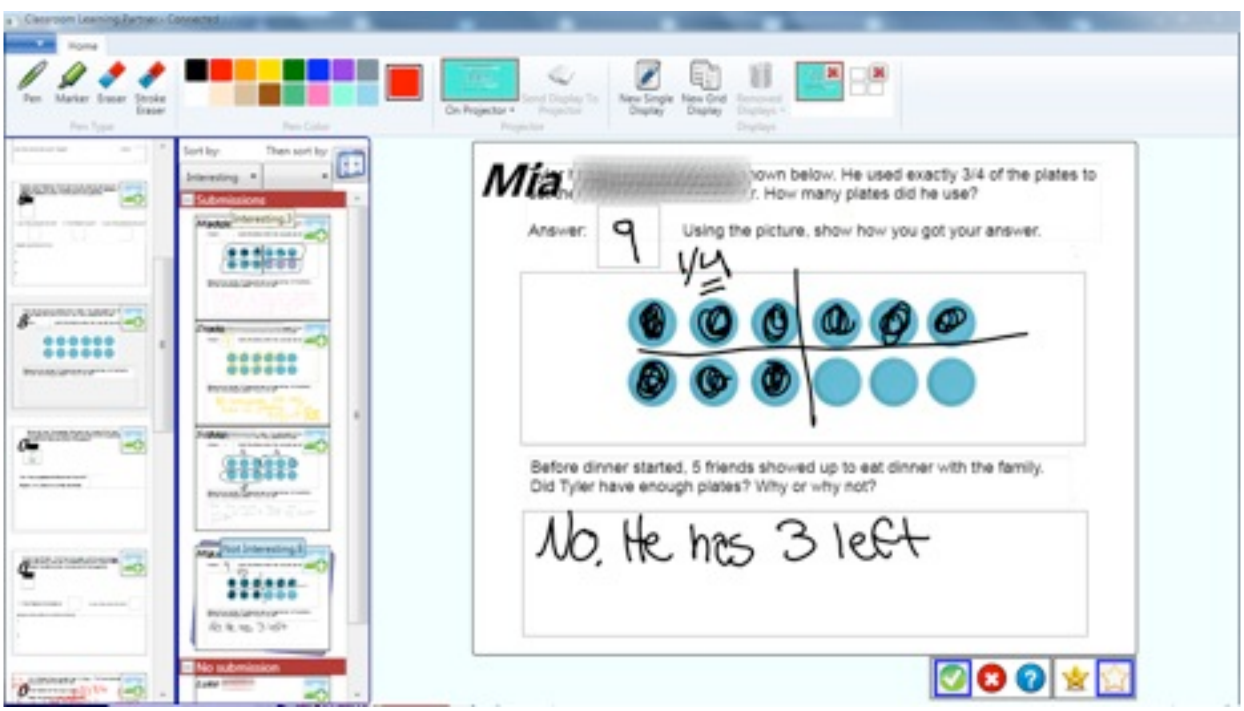

Figure 4: The teacher interface allows teacher to view student work. The second column sorts work for each page as either interesting or not interesting, as defined by the teacher.

2.4 Previous Data Storage Strategies

In previous versions of CLP, student work data was stored in a relational MySQL database. This database emulated an object-oriented database by having a different table for each relevant object. The schema used was functional but large and difficult to change.

3 Data Storage Goals and Challenges

The first two months of UAP involved investigating the data storage and viewing needs of the CLP project and proposing designs. During this portion I realized that the previous data storage model did not fit the current state of the system.

3.1 Goal and Assumptions

The data set to be stored consists of student work, classified in some way by an AI pre-processor. These classifications are a superset of the categories used in the teacher user interface to sort submissions. For example, consider the number line problem in Figure 5, with sketched work. This work could be classified by correctness, how many ticks are drawn, the average and standard deviation of space between ticks, and so on. The categories used to classify work are dependent on teacher preference and available pre-processors and will therefore vary from page to page.

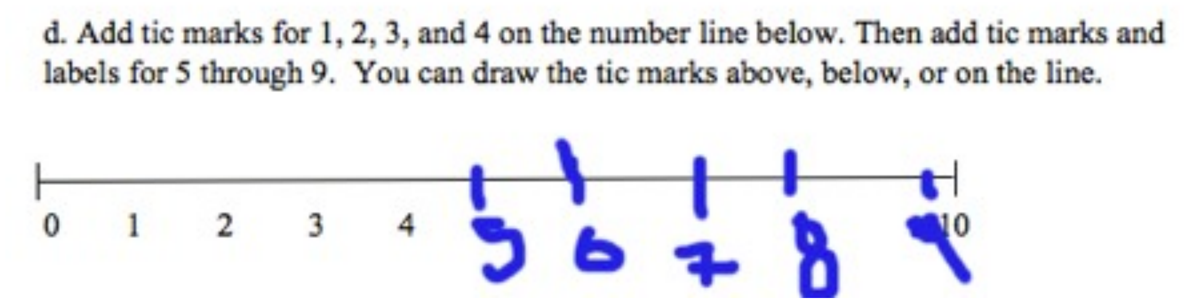

Figure 5: A sample number line problem.

Two types of users will be accessing our data viewing system: teachers and researchers. Both groups will want to be able to define queries on the data set, although likely over different sets of students and attributes. For teachers, the viewing system should function something like a grade-book. Teachers, by default, should only have access to work and data recorded from their own class, although they also may want to be able to compare student performance across years. Researchers will want to look at the whole data set. Researchers also will have great interest in various tablet-centric attributes of submissions. For instance, they may want to compare correctness of submissions with the amount of work the student erased on the page. Teachers are less likely to perform queries like that. Finally, as data will be viewed by a diverse group, some of whom may not be in possession of a tablet, the system should be system independent.

3.2 Challenges

Older versions of CLP used a relational database schema; however, the variable number and type of classification per slide make that now an unwise design decision. Consider the following classifications: *correct* (a boolean), *area shaded*(a double), and *words written*(a string). Each one is a value-pair, meaning it has a identifier, such as *correct* and a value, such as *true*. A graph for a science problem could be classified by all three value-pairs; for a fraction shading example only *area shaded* and *correct* make sense.

The value-pair data model does not fit with a relational database schema. There are ways to force it to fit but they cause performance or data integrity losses (Karwin 2010). For example, one solution, show in Figure 6, involves listing the value-pairs in one column of the database. Unfortunately, this scheme prevents users from being able to run SQL queries over those categories, since parsing must be done to extract them. Additionally, there is a limit to how many categories one page can have, since database columns have fixed character widths. Another proposed solution, shown in Figure 7, is to have a table just of these key-value categories, each one referencing the page in the database to which it applies. Since all the values for a category are stored under the same column, the database cannot mandate data integrity or data types. For example, *area shaded* value (a double) and the correct *value* (boolean) fall in the same column. Furthermore, queries that include all instances of a particular category, e.g, looking at all correct answers, could require a complete scan of the key-value table, which would be very large.

Submissions Table

PageKey	StudentKey	DateTime	Strokes	ValuePairs
page21	8234	1305475121	A8dsf3...	correct:true, numStrokes: 3, stampUsed:true....
page22	8234	1305475314	MSs3ra...	color:red, numStorkes: 9, percentShaded: 34 . . .
page21	8240	1305475241	M832cB...	numStrokes:7, correct:true, stampUsed:true ....
page22	8240	1305475322	sdf22W...	color:blue, numStorkes: 4, percentShaded: 34 . . .
. . .				

Figure 6: Example of listing value-pairs in one column. This design is poor because individual categories cannot be searched for and because the number of pairs that can be stored for one submission is finite.

Submissions Table

ID	PageKey	StudentKey	DateTime	Strokes
234	page21	8234	1305475121	A8dsf3...
235	page22	8234	1305475314	MSs3ra...
236	page21	8240	1305475241	M832cB...
237	page22	8240	1305475322	sdf22W...
...				

ValuePair Table

ID	Category	Value
234	correct	true
234	numStrokes	3
234	stampUsed	true
235	color	red
235	numStrokes	9
235	percentShaded	34
236	correct	true
236	numStrokes	7
236	stampUsed	true
. . .		

Figure 7: Example of listing value-pairs in separate table, using same dataset as before. This table grows quickly, cannot insure datatypes, and is not easily queried over categories.

4 Database Design

The most important concept in database theory is to choose the database that is appropriate for the job. Due to the project's data requirement, I propose a non-relational database design. A non-relational database does not have schemas or tables. Instead, it stores data and data types defined on the fly. Non-relational database are great for complex data structures such as the ones seen in this project, and are often faster than their relational counterparts. As a trade-off, these databases tend not to have support for transactions. This issue is not a problem for this project, as there are no long string of actions that need to be made atomic.

I propose the use of MongoDB, an open source non-relational database with an active development community. MongoDB is used in production on many sites and is also compatible with C#, the language used in CLP development. MongoDB has replication management built in, to limit downtime in the event of failures and to prevent data loss. The system is also easily scaleable. Relational databases scale vertically. It is difficult to divide tables between machines in such a way to create acceptable amounts of traffic between them. The more schools that use CLP, the more powerful a database server needed in order to maintain performance. MongoDB scales horizontally. As more schools use CLP, cheap machines can be added to maintain performance. Furthermore, MongoDB has built in functions for working with many requests to access distributed data.

Most importantly, MongoDB stores information in documents that are defined on the fly, in key-value-pairs. For example adding information from a page about a number line would look like this:

```
document = ( { "Subject" : "Math",
…"StudentID" : "stud834",
…"DateTime" : "1305820721",
…"PageBaseID" : "page21",
```

```
…"Strokes" : "lafAr3wrFLKDFaw3qwoirDLaslkfjaw321wriojfadslfkj ...",
…"Correct" : "Yes",
…"AverageSpaceBetweenTicksPx" : "30",
…"STDSpaceBetweenTicksPx" : "7",
…"LabelsUsed" : ["0","5","10", "15"]
…} )
db.pages.insert(document)
```

In this example, StudentID and PageBaseID reference other documents in the database that contain the full record, e.g., name, school, etc. for the student. Keys can be determined on the fly and can vary across pages. Data of arbitrary length, such as ink strokes, can be stored, as well as images or other file types. Data retrieval is in a straight forward, SQL like, way as well. To retrieve all correct answers on this page, one would use the following query:

```
db.pages.find({PageBaseID:'page21',Correct:'Yes'})
```

While this design represents a major design change from the MySQl database of the previous version of CLP, given my research I feel it best fits the current CLP project and supports future growth.

5 Data Visualization Site Design and Implementation

The second portion of this UAP was focused on designing a data viewing system, enabling researchers and educators to investigate trends visually. Since, as stated in Section 3, this system should be accessible to those without tablets, I designed a system for the web.

5.1 Motivating Sketches

The basic setup of the data-viewing interface was designed in my 6.UAP proposal. It involves a column on the left that specifies the teacher's classes for which data is available, graphs in the center show various metrics over time, and a button on the upper right for customizing the graphs (See Figure 8). The graphs are views of the data. A view has a type (histogram, time plot, etc) a label, and a query that selects what portions of the data to display. Versions for teacher and researchers differed in the selection of classes for which data could be used. Teachers do not have the option of choosing a class that is not their own, unless they have been given permission by that class' teacher.

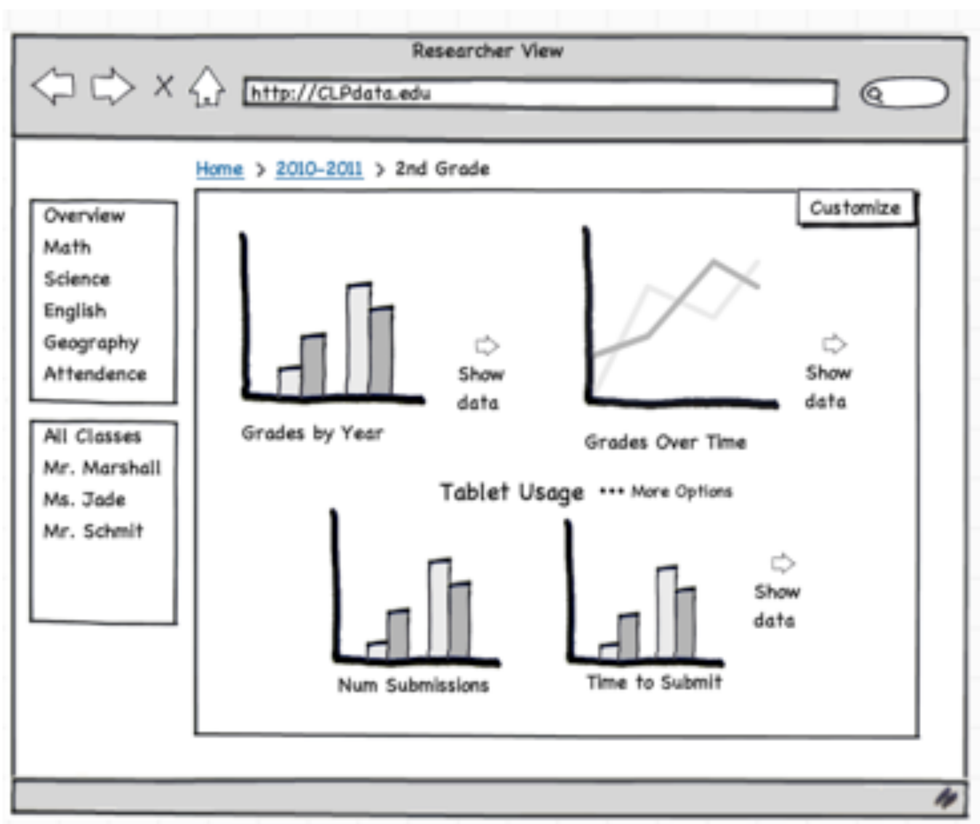

Figure 8: Researcher DataView

While these designs conveyed the importance of data viewing system, they glossed over many important aspects. For example, there is no ability to search different attributes nor is there a way to specify a data range to be queried.

5.2 Improved Data Viewing UI Design

For this UAP I refined this design and created additional UI screens. The original designs did not detail how to specify different queries. However, having certain data views displayed to the user automatically, as in the original sketch in Figure 8, improves efficiency and accessibility of the system. Teachers often want to look at the same metric, so having default measures for certain portions of the data makes sense. For example, a teacher might monitor a histogram of class averages for the week, as well as look at how the overall average was changing over time. While he or she w=could make other queries, these predefined ones serve as summary metrics that are easy to interpret. The screen shown in the original design sketch serves as this default.

For customizing data views, I designed a new screen . This screen lists the features of each available data view, together called a collection, and has a small preview of how the collection on the screen (See Figure 9). To prevent a page from becoming unwieldily, a collection is limited to 6 data views. Additional collections, however , can be defined. As shown in Figure 10, the user can create a new view by defining a type and label, and crafting a query.

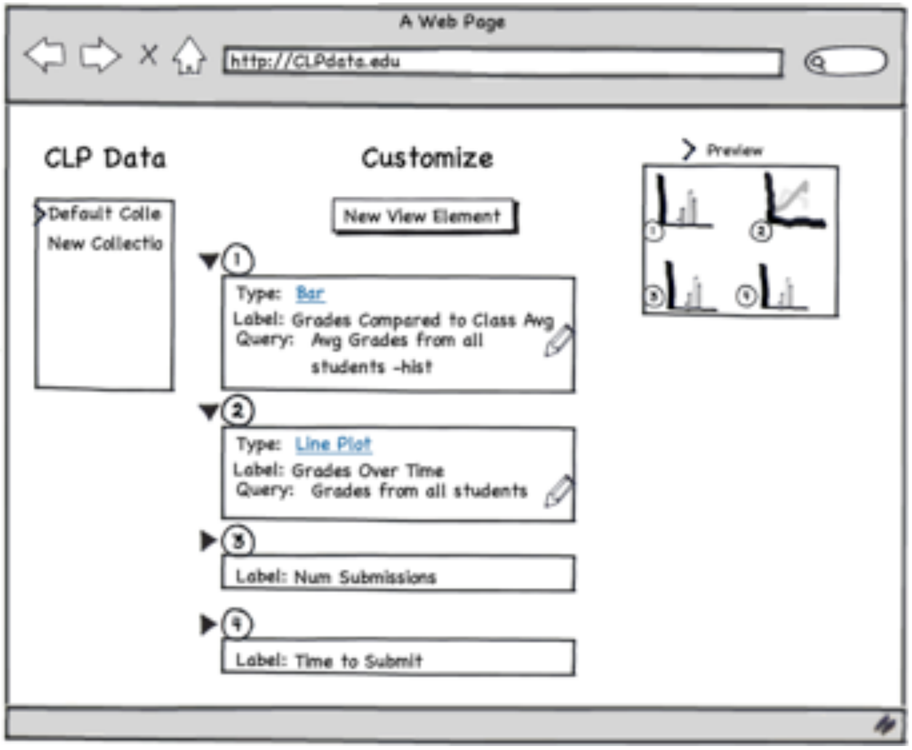

Figure 9: Design for customizing a collection of views. Individual views can be collapsed, reorder, and edited.

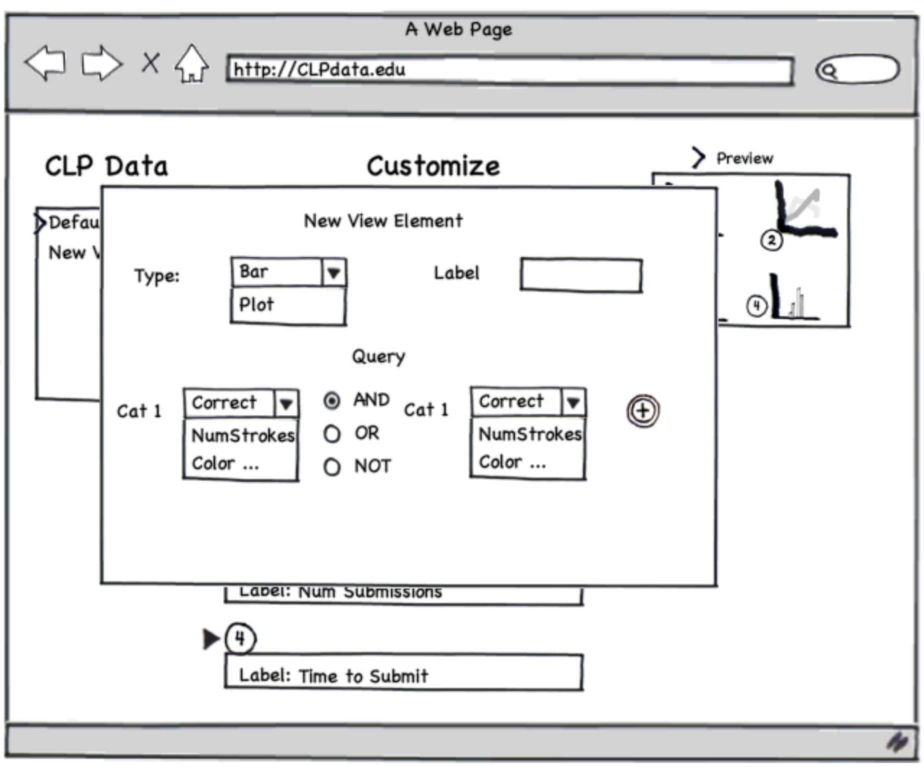

Figure 10: Design for adding a new collection of views. Possible categories are populated in drop down menus automatically.

6 Implementation of Data Visualization Page

I created a prototype web page to demonstrate the feasibility of the Data Visualization concept. This web page displays submission information graphically. The prototype system shows information in histograms and line graphs about average percentage (proportion of problems correct), number of submissions, and time taken to submit.

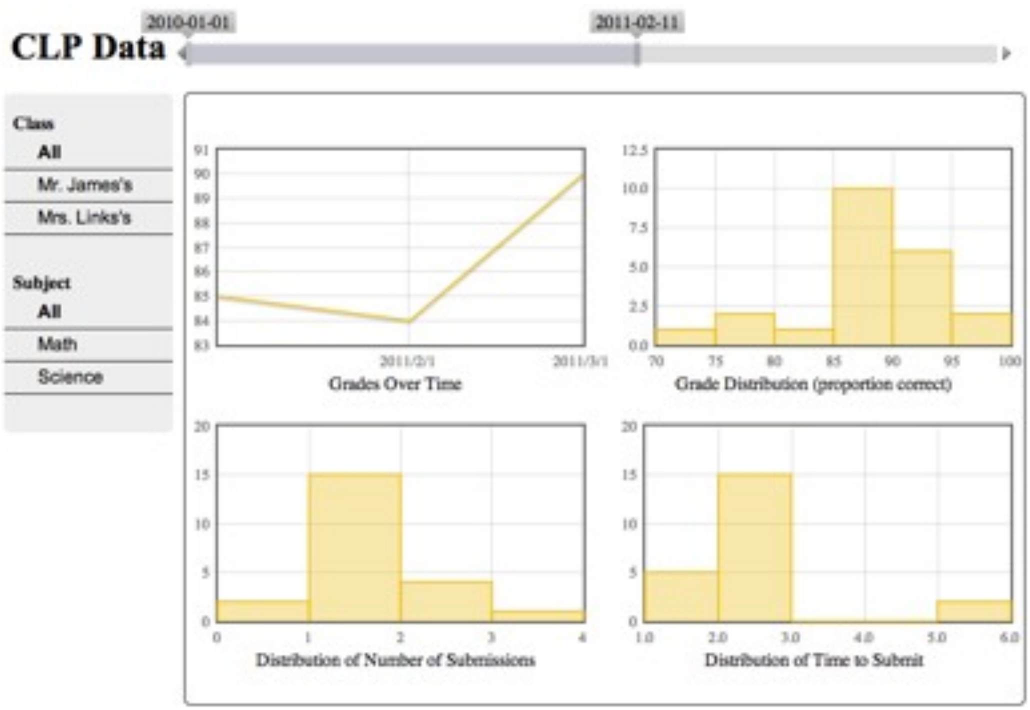

Figure 11: Screenshot of prototype data viewing web page. Prototype includes a date slider for selecting range of dates to query.

7 Future Work

I will be continuing this work next year as a MEng project. Goals for next year include having an end-to-end system that logs student work and makes attributes available online. Improvements to the online system will involve the implementation of the customization screens, as well as wide-spread user testing.

7.1 End to End Connectivity

This year, reimplementation of CLP began as part of a new INK-12 project grant. the organizing metaphor is that of an electronic notebook. The networking and submission system was implemented at the end of the spring. In the submission module, notebook pages are sent wirelessly back and forth between student, teacher, and projector machines.In the next iteration, we will add a database as a repository for the student notebooks, teacher notes, and public displays. A central research question will be whether we can insert the database in the communication paths among students, teachers, and the projector without significant time delay. We will begin by storing student submissions in a database, rather than having them sent directly from student machines to teacher and projector machines. The teacher and projector machines then will retrieve student submissions from the database.

The next step will be to store all student work, including that in student notebooks but not submitted, in the database. This is a more difficult problem because it involves much more data. For this task, the first step will be to model how the history of student work is stored. While taking full snap shots is the easiest solution, only storing changes students make is much more economical in terms of space. This method requires a way to represent change in ink strokes. Currently, strokes can be packed into a string. The addition of two strokes could be represented as two of these strings, but the subtraction could not be. Furthermore, it is unclear how erasing relates to strokes, since in CLP erasing can happen at the pixel level, not just at the stroke level. The second step is to decide when snapshots should be taken. Taking snapshots when students save work locally seems like an obvious solution. However, that may not be frequent enough. Classroom testing will help determine the necessary parameters.

An additional difficulty in capturing work is that the CLP system involves having a local classroom server to avoid internet connectivity problems with a main server back at the lab. Thus a small local database would have to be merged with the main database after classroom use. MongoDB has built in import-export tools, so I believe this merge is feasible. Working with images, however, may provide challenging. Cleaver naming and labeling schemes will help avoid potential overwrites.

The final necessary component will read data remotely and perform a wide range of queries through the web interface. While accessing data is simple, knowing what to access may not be trivial. The database can hold any type of category value-pairs, so informing the user of what kinds of queries are available is an interesting database and user interface problem.

7.2 Data Viewer Improvements

Writing an interface that captures the range of queries possible while still being efficient and learnable will be difficult. I will be attempting to duplicate much of the functionality of a command line interface in a graphical interface that is accessible to all researchers and educators. As stated before, this interface will also need to inform the user of what views and categories are present, so he or she knows what kind of queries are possible. My design sketches provide a framework for this but needs to be improved. It does not explore how users will know what categories are available or how to specify other parameters, e.g., how to separate test scores for boys and girls.

In order to deal with these challenges, I will use a iterative design process. This process involves making frequent prototypes, getting user feedback, and improving. Prototypes will start out as hand and computer sketches of at least three different designs. After choosing one, I will use paper prototyping, allowing users-teachers and researches-to 'click' through my system with their finger. Next comes a computer prototype, with canned responses and then the complete system made. I am fortunate to have teachers and other members of the CLP team willing to act as testers.

8 Contributions

For this 6.UAP I investigated the data management needs of CLP and designed an end-to-end system to accomplish the project's goals. Unlike the previous version of CLP, my system uses a non-relational database due to its flexibility and scaleability. This database matches the needs of the novel teacher interface and the sorting being implemented using AI methods. I also designed a data viewing system. This web interface will enable teachers and researches, even if working remotely, to perform meaningful queries over the data without any database knowledge. Finally, I implemented a proof of concept web page to begin to test the feasibility of my design. This data management project has moved in step with the software implementation from the last semester, and we have now reached the point at which the two can be integrated.

9 Acknowledgments

I would like to thank my 6.UAP thesis advisor Dr. Kimberle Koile for her guidance and interest in this project. Working on the CLP team for a semester and seeing how complex systems get designed, implemented and tested has been eye opening, and the resulting student empowerment from CLP is inspirational. I would also like to thank the other CLP group members for their help and collaboration as I worked through designs and the teachers and students we worked with for their insights and time.

10 References

Koile, K., Chevalier, K., Rbeiz., M., Rogal, A., Singer, D., Sorensen, J., Smith, A., Tay, K.S., and Wu, K. Supporting Feedback and Assessment of Digital Ink Answers to In-Class Exercises. (2007a) In *Proceedings of IAAI 2007*, July, 2007.

Koile, K., Chevalier, K., Low, C., Pal, S., Rogal, A., Singer, D., Sorensen, J., Tay. K.S., and Wu. K. (2007b) Supporting Pen-Based Classroom Interaction: New Findings and Functionality for Classroom Learning Partner. In *Proceedings of First International Workshop on Pen-Based Learning Technologies*, May 24-26, 2007.

Koile, K. and Rubin, A. NK-12: Teaching and Learning Using Interactive Ink Inscriptions in K-12. Submitted to NSF Discovery Research K-12, NSF 08-602, January, 2010.

Chao, Neil. Masters of Engineering Thesis, EECS, MIT, May 2011.

Jozwiak, Martyna. Masters of Engineering Thesis, EECS, MIT, May 2011.

Karwin, Bill. SQL Antipatterns: Avoiding the Pitfalls of Database Programming. The Pragmatic Bookshelf: Raleigh, 2010.