CONSTRUCTING AN ARCHITECTURE FOR AN INTERACTIVE EDUCATIONAL RESEARCH DATABASE – ISSUES OF DESIGN AND IMPLEMENTATION

Derek Beaton
University of Massachusetts at Dartmouth, Computer and Information Sciences Department
285 Old Westport Rd. N. Dartmouth, MA 02747

Stephen Hegedus, PhD
University of Massachusetts at Dartmouth, Mathematics Department
285 Old Westport Rd. N. Dartmouth, MA 02747

ABSTRACT

Educational research involves multiple inter-related sets of data. These are often hard to document or archive. We present the design of a simple system that allows researchers and participants (in our case mathematics school teachers) of educational research projects the ability to create datasets that can be synchronously linked, accessed, annotated and easily searched, for analysis purposes. From our preliminary work, which is part of a large-scale R&D project dealing with such complexity in mathematics education, we present features of an architecture that has been useful in assimilating data sources from multiple resources and analyzing such in a collaborative manner. Our on-line system incorporates the use of PHP and MySQL on a UNIX server. Source code is freely available.

KEYWORDS

MySQL, PHP, research, archiving journal, self sustaining system, pattern recognition

1. INTRODUCTION

The purpose of this paper is to present the preliminary developments of an archiving and retrieval system for a massive data collection project that is fundamentally initiated within a complex educational paradigm of mathematics education research. We present the fundamentals of a design and implementation procedure of a database that was conceived as a system of two parts: 1. A multi-user system of data objects, primarily raw artifacts of classroom interventions (e.g. digital video, classroom observation protocols) and 2. Processed attributes of raw artifacts that can be easily assigned to users through a simple user interface, but was later extended to a system whereby entries could be shared within users, analogous to a chat-like environment. This last feature seems almost obvious, and well documented, but we present preliminary data that highlight how such a system, within a classroom research environment, and which is sensitive to issues of user-interface and design, can produce advanced electronic data for classroom-based research evidence, allowing users (with little background in databases) to observe data artifacts of each others classrooms, critique and submit associated commentary, as well as offer researchers, deep and valuable insights into how users of the system observe, and reflect on the contributions of other users. We believe that it is in the creating and archiving of sensitive, complex data from classrooms (in our case high school mathematics classrooms) in orchestrated ways, that enables us to create avenues for users of the database to further create, in an easy and guided fashion, intelligent reflections of existing datasets. This creates an iterative procedure, embedded in our core design, of collecting raw research collection and archival, followed by user-embedded-in-the-data-archive reflecting, followed by analysis, and finally embedding their perspectives and reflections on the existing datasets, into, and hence extending, the existing dataset.

To enable our experiment, we have used PHP in conjunction with MySQL, to offer a simple, ubiquitous solution for users to access and use our database through generic Internet browsers. We chose this option initially for purposes of a proof-of-concept experiment to test the basic attributes of our architecture. Our future work will address the problematics of extending the core architectural features (which we report here) from the required interface attributes of Internet browsers to a more generic, cross-platform native-code version yet preserving the users abilities, and making more efficient, users actions on new and archived data.

2. BACKGROUND

Our main body of data arises from a huge data corpus gathered from over 8 years of classroom research focused on students' learning of core mathematical ideas in algebra, pre-calculus and calculus as well as teachers development in pre- and in-service education. We have developed simulation software on both graphing calculators and desktop PC's with a core aim of democratizing access (Kaput, 1994; Hegedus & Kaput, 2004) to difficult ideas in mathematics through interactive visualizations (e.g. screen-graphically and algebraically editable functions). The main data source of our work presented here is not in this new software development, but in the tracking of its use in real classrooms. This includes two main sources of data collection, encapsulated both inside and outside the classroom. First, the documentation of students interacting with peers and their teachers, via multi-source video data focused on specific groups and the teacher at a central display. Secondly, the reflection of teachers watching video clips (from source 1) of each other teaching similar concepts with the software (based around an intervention) and commenting on each other's practice via the on-line database. Our architecture allows researchers to quickly select pertinent episodes from classroom data being archived, produce related questions to that clip and submit it to participant teacher (still on-line within the same database), for response, critique or annotations. This creates a community of inquiry (in the spirit of Lave & Wenger, 1991), but in an electronic sense, where teachers can participate in research by observing other peers, privately or publicly, and commit their perspectives into the emerging dataset. The key here is that these reflections are "attached" to the original raw classroom data artifacts, thus making such a data artifact a renewable, multi-analyzed, data object. Continually iterating as long as the process is administered. These analyses are deeply attached to the original raw classroom artifact. So our architecture offers a system, whereby the data cannot only be collected, using a simple interface (details below), but offers a mechanism to create an organic system where educational researchers and partners can watch, reflect and deepen their understanding of raw data. So, a unique feature of the system is that the data, can be segmented and shared with partners of a project, whereby they can add their commentary on the episode subject to some cursory questions from the researcher. This system is architectured in such a way that it could be applicable to a multi-tier user system, where privileges are customizable, such as whether a person can read other people's entries (important in a school system where some teachers might not want to let their classrooms be shown to administrators, or where students should not see comments made by a teachers to other students). But teachers might want to interact, or view classroom episodes of peer-teachers.

Video as one source of analysis is widespread in mathematics education research and practice largely because of its inherent usefulness in proving the basis for a shared experience that can be re-examined as needed (for recent examples see: Boaler & Humphreys, 2005; Carpenter, Franke & Levi, 2003; Clarke & Hollingsworth, 2000). We are adding to these uses a dynamic capacity to annotate video excerpts and other classroom data, as well as to produce new annotated excerpts from archived video data in ways that are simple and transparent to a classroom teacher, yet supported by a powerful relational search engine in the background, with our on-line database.

3. BACKGROUND

Until a redesign of the database system in place, all electronic research data had been housed on an Apple Macintosh system running a FileMaker Pro Server. This system handled the majority of data that needed housing well. An advantage that the FileMaker Pro Server had provided was the ability to be cross-platform, through connecting to the server. FileMaker was unable to handle the amount of video that we needed to process.

A secondary system based on PHP and MySQL was implemented for a simple online “journal”. This web interface software application was used as a means of prompted self reflection for students in one of the observed classroom settings. With the implementation of an online journal, an unprecedented amount of data had been flowing in. With daily reflections from multiple classrooms, with many teachers, it grew unexpectedly. The system in place was not designed for simple or constraint based search methods, proving a detrimental roadblock for research. The new concept of journal reflections becoming research artifacts had to be stored using a MySQL database on a UNIX system. However, all previous data (such as video, debriefings, field notes) were housed on an Apple Macintosh system running a FileMaker Pro Server.

With two distinct systems, rich in various amounts of data, joining the two systems, a single housing was the obvious solution. Due to the large amounts of video data, it became apparent that FileMaker was not perfectly equipped to handle that data. A decision was made to migrate all data to a MySQL database. Unfortunately the loss of a user interface in the move had occurred. It was immediately recognized that a cross platform interface (much like FileMaker’s) was necessary for the MySQL database, which had no user interface. A new design was necessary to handle any form of data, existing types and artifacts, and unforeseen data types and structures that could be designed as technology and research methods evolve. The new design had to be completely dynamic, highly interactive, cross platform and easy to use as users were not previously given access to the FileMaker database because it was specific to artifacts. The new system had to be designed and equipped to handle the differentiation of “journal” users and “data-input” users. Leaving behind the existing notions of a database for the sole purpose of archiving sensitive data, it has now become a tool where data can be collected and generated.

4. ARCHITECTURAL SPECIFICS

Initial decisions in the design process were about who and how they had access. Our former system did not accommodate the complexity and user levels required by such a large research project. The initial decision was that four types of users at any given time will need to be involved in this database.

1) Superuser/Admin – Total System Control
2) Data Inputter – A “Sub-Superuser” they have the ability to edit certain aspects, such as adding new groups, users, classrooms, and data.
3) Teacher/Student – See only the journal front end that previously existed.
4) Navigator – A research style user – someone that only has access to the search abilities of the data, for analysis and consultation.

With each level named, and a decision about what they can do, they needed a navigation system. This system was a simple and unique system of “teeth” at the top of our page. We were held on to our “teeth” design, from web pages created for the SimCalc project (www.simcalc.umassd.edu), and the Mathematics Education Research Group (MERG) of the University of Massachusetts at Dartmouth (merg.umassd.edu).

The “teeth” system is based on the principal of being one click away from almost any where on the site. Keep it simple in the front (Raskin, J, 2000) and conceal the complexity for server side processing – our principal for design of the database front end, and the various views for the users. Upon establishing what our users can do, we encountered a problem – we encompass several styles of web interfaces, and infrastructures all in one. How could this problem be tackled simply? Each of our users has the potential of being associated to separate projects or separate groups (and quite often, many of each) while being a part of this project, or one of the classrooms that it will be affecting. A design and development altering decision needed to be made – how to differentiate users to different groups, with potentially different levels across the system? A self assigning registration process was implemented. This process requested a username, password, first name, last name, a group or project from a drop down menu and a code. This code is supplied by someone who is an admin for the database, most times that person is the teacher of a course being studied, or a course in which vast, rich amounts of data can be collected, and distributes this code to the participants in the course (for creation of the data) and any collaborators (for the observation and analysis of that data). This way, registration is made simple and administrators do not need to be concerned with user levels.

The next step after establishing our users was to establish what kind of material would be housed. An understanding of all forms of data to be placed in the database was utterly crucial, and the main reason that a new database was being developed. Several forms of data needed to be housed, and needed the ability to be a part of a relationship, and have many associations, to almost anything in the database. There are clear lines between what was called "data-sessions" and "analysis"; Data-Sessions are artifacts that are products of direct study and research, such as a video clip from a classroom, a test or handout from a day a classroom setting was observed, and analysis artifacts are processed data sessions and analyses. Lines blurred as we approached newer types of data: pieces of raw data that are themselves, analysis, but still data to be analyzed. Such an example would be a video clip of many people discussing and analyzing a previously unanalyzed piece of data. This could not just be placed into the database as analysis or raw data. We needed a new title, and to associate this piece of information to raw classroom observations, and already analyzed pieces of data. Birthed from this analysis-to-data session association idea – associating bibliographic and library entries to any form of data was the next logical step. Eventually a database where all "data-types" had a many-to-many relationship became our design.

The primary associated data of this design has loosely been called "data-sessions" – information stored under fixed pretences, where various categories of data can be stored. A system of related material needed to be both inherently and programmatically associated. The user needs to see and understand that piece A is related to piece B, because they are from the same session:

1) Originating School
2) Class
3) Teacher
4) Date
5) Start Time
6) End Time

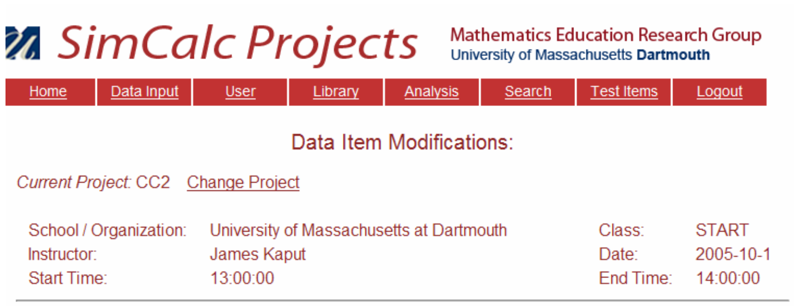

Figure 1: Example of created data session with required fields. The six distinct fields are crucial to each dataset, but allow for every dataset to be unique.

These fields can be changed to accommodate any setting of restraints and uniqueness of the data being uploaded. If these criteria are met when inputting new data – you are put into a state to continue uploading new pieces of data to an existing session. Within every session are subcategories to show that specific items are disjoint, though have some unique feature that they all share (data session parameters). Within the data sessions are "clusters" – these clusters are headings to various sorts of data, within one data session:

1) Video
2) Classroom Activity Documents
3) Homework Documents
4) Assessment
5) Teacher Materials
6) Field Notes
7) Debrief

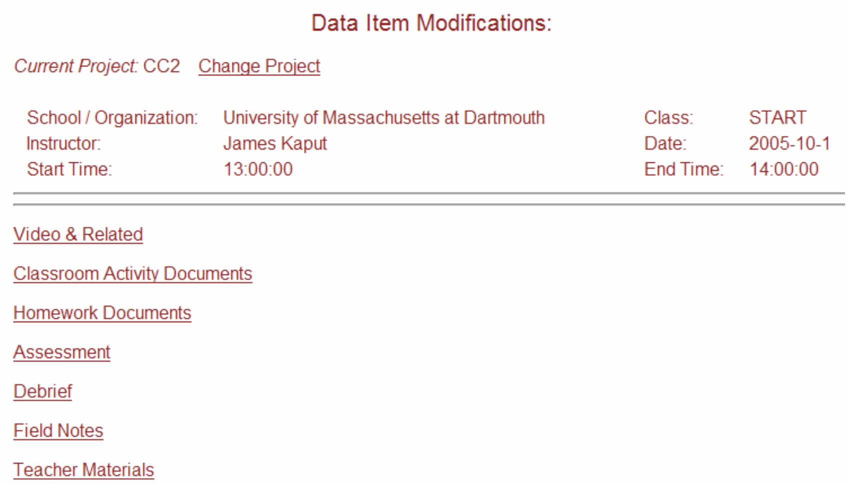

Figure 2. Inside the data session there are options for the "parent-child" structure of data session to data object categories.

Each of these clusters can house their own data, or even subcategories. This cluster style is a "parent-child" like relationship, just like the data-session to cluster is "parent-child" as well, where the clusters act as parents, which can hold their own data, but may have children, which can also hold a more specific data. These can be updated and changed at any time, for any constraint required. Currently, Video is an umbrella for four more subcategories:

1) Video Index
2) Video Transcript
3) QuickTime Clip
4) QuickTime Clip Transcript

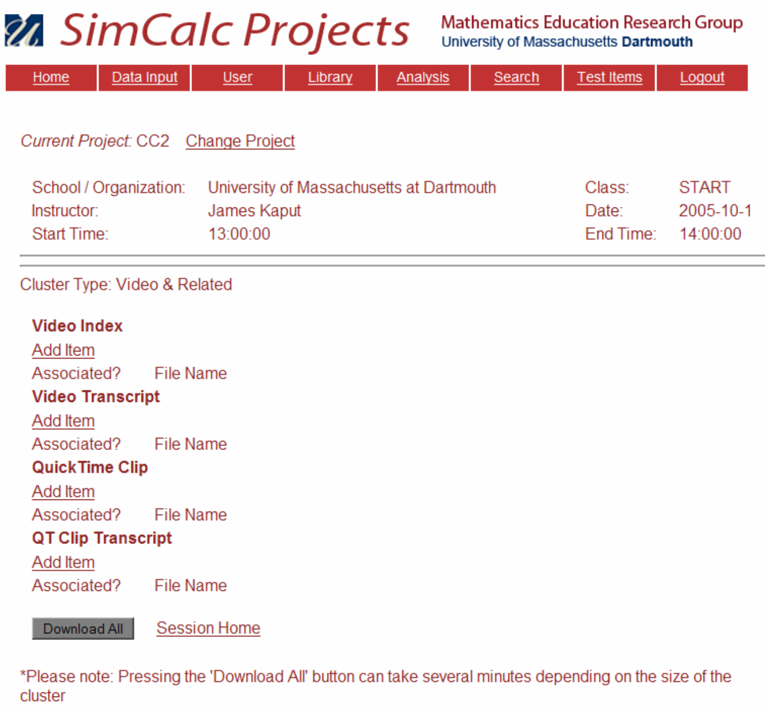

Figure 3: "Video" data object category selected; This shows the "parent-child" relationship between data object categories and their sub categories or actual objects.

All the previously discussed categories, titles and headers are absolutely crucial to the structure of the various projects housed under the “SimCalc Projects” title. Extremely detailed information is gathered throughout the academic year, and processed year-round. The organization of information input has made the analysis process more efficient. Organization in the education research field is the most important part of success.

With the adaptation of streaming video from archived video data, reflections of classroom scenarios can be accomplished through teacher interventions, providing a forum in person and on-line or on-line only in a analysis like setting created to make teachers understand the high and low points of strategies and classroom environments.

5. REFLECTIONS ON DESIGN

Interactivity and interface are very important issues when engaging in a technologically oriented research project. As “tech-savvy” as the world is becoming, there are huge divides between what the developers and designers would like to see, and what the users would like to see, for the face of a piece of software. A scrupulous amount of detail went into studying various interface formats for web based database interactivity. The user needed to feel comfortable with the database. There are many types of users, and all of them had to feel as though this was a natural process. In the design process, it was apparent that the development process had to make choices and options for the user crystal clear. What are considered options might be specifications to certain pieces of data, such as time, day, and place, but the users needed guidance without feeling trapped in a cumbersome system that helps make incorrect decisions because of poor set up.

The development process immediately shed light on exactly what flaws the original design had. Within days it was realized, that the user system needed a drastic change. A differentiation between the research data side, and the “journal” style entry system was needed. The “journal” style entry system had very restrictive permissions, where as each other user did not. It became apparent that the differentiation came in the form of “Project” and “Group”. A “Project” is that in which data is collected, stored and analyzed through the database. A “Group” was going to be the new medium in which data was created, the “journal”. There is commonplace between Projects and Groups – task assignment. Regardless of what user level a person may be, the same assignment could be assigned across the database. This task system is extremely versatile – it currently is used for assigning homework, journal entries, and project based work tasks, such as a time frame for a development cycle. Every task has a creator associated with it, so that the assigned person, may respond with a completed assignment, or an update. That same creator, can also respond with feedback. This system allows for a manager to travel the world and know exactly who is doing what, and when.

Object oriented code has some very real benefits but with it also comes some serious performance overhead. Working with PHP it became apparent that not everything could be objectified. Object Oriented PHP (PHP 5) was the front end language decided upon. Objectification had to be something that was partially developer initiated due to the fact that PHP 5 is still fairly new, with procedural programming roots.

Due to the fact that a relational database and not an object database was what was being designed, it was decided to use Objects to bring together the data from disparate logical locations, into one cohesive entity. This lightened the load on whatever procedural code there was allowing access to all the different parts of the database, and could interact with just the object. It also allowed the flexibility to change things in the database and only have to change the object code.

A decision was made to leave process oriented code as procedural, things like uploading a new piece of data, or submitting answers to a journal. Most of the time the code would be similar, so there were benefits from objectifying these as well, but the overhead involved with creating a class for each page would have been incredibly high. It would have easily tripled the amount of function calls. After weighing the costs and benefits, it was decided these pages were best left to procedural code. As the database grew it became quite clear how much this decision may have cost, because with each new feature large portions of similar code had to be changed.

6. CONCLUSION

This new system takes “databasing” in a new direction. A system now designed for archiving data as well as collecting data, is in place. Taking an archiving system and turning components of it into a reflection based data collection tool was simple to do programmatically, but very difficult to design, organize, and incorporate into an archiving system. With the diversity and complexity of our system, we have created the foundations of an open source system that can revolutionize databasing and research. A single piece of software on a single server is a fairly simple system to implement and start using. The foundations of this system are in educational research, but its reach is far beyond that.

6.1 Future Development

The future of this system holds many potential features. This system was intentionally designed so that it could be expanded upon quickly, and “dropped” into place for another environment, on another system. The sheer complexity of the data and premise behind this system, and the massive amounts of data really lend itself to being automated, almost entirely. The future development of this system will incorporate self associating algorithms. After the database recognizes that users are making an association, or a search and coming up with the same result(s) every time, the database will start associating sessions to one another. Similarly, the initialization of a name tracking system is in place, but not yet instantiated on the front end. This is leaps and bounds beyond conventional methods of tracking research subjects through time. Patterns will develop with subjects, and this system should be capable of doing this in the next few months.

With large amounts of video and image data, it is quite difficult to analyze this data. Key points are always remembered about a session, but the session itself may escape the analyzer. This is why Content Based Image Retrieval (CBIR) is one of the next steps to be taken in this system. CBIR is a process in which a palette or upload option is provided for the user to “provide” an image to the system, the system then thresholds the image into basic colors, shapes and a set size, and compares it to all the image and video data currently in the database. This process would either entail thresholding the video and image data on the fly, or preprocessing it as it is uploaded, cutting search time and complexity by incredible amounts. “Orphaned” video data (data that may have lost it’s time, place, and reason it was taken) can now be input and searched. This could be a crucial piece of data, with out all known parameters, which would make searching for this piece of data very difficult.

One of the distant goals for this system is natural language processing, and voice and language recognition. This may sound a little strange – but it would be an amazing step forward in video transcription, and data searching. Processing audio of a video takes time, patience, and a person to stare at a screen for hours. With natural language processing, it becomes far more simplified, and cost effective. To accompany that is the voice and language recognition system. This system would be used as a search tool. The user could speak a phrase and search by words, sentences, intonation or even their own voice.

The seemingly last stages of this system is to have in place a network of cameras placed in various research environments for streaming video observation. The streaming video can also be recorded, and immediately put into the correct place in the database. Have a network of cameras in research environments then eliminates the need to haul audio and video equipment from on environment to the next, and also eliminating the process of converting video.

Ultimately the goal of this system is far from what it stands now. The current system is an amazing step forward in the research world for analyzing and collecting data, but with the addition of the future systems, it becomes an ever evolving, automated research system. The sheer power behind this system will be immense. Imagine a system that can collect data, process that data, and make associations and detect patterns, automatically. This is the direction our system is going, down a revolutionary path which will change the face of research and databasing forever.

ACKNOWLEDGEMENTS

Dr. Stephen Hegedus and Derek Beaton would like to acknowledge Natalie Hegedus on design and layout of the interface, Jacob Sylvia and Andrew Hanson for development and all employees, associates and colleagues of the SimCalc Project for their feedback and critique of the system.

REFERENCES

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. New York: Cambridge University Press.

Raskin, J (2000) *The Humane Interface: New Directions for Designing Interactive Systems.* New Jersey: Prentice Hall.

Clarke, D.J. & Hollingsworth, H. (2000) Seeing is understanding: examining the merits of video and narrative cases. *Journal of Staff Development*, 21(4), 40-43.

Kaput, J. (1994). Democratizing access to Calculus: New routes using old routes. In A. Schoenfeld (Ed.), *Mathematical thinking and problem solving* (pp. 77–156). Hillsdale, NJ: Erlbaum.

Boaler, J. & Humphreys, C. (2005) *Connecting Mathematical Ideas. Middle School Video Cases to The Social Life of Information Support Teaching and Learning*. Portsmouth, NH: Heinemann.

Hegedus & Kaput (2004). An introduction to the profound potential of connected algebra activities: issues of representation, engagement and pedagogy. *Proceedings of the 28th Conference of the International Group for the Psychology of Mathematics Education (Vol. 3, pp. 129-136).* Bergen, Norway.

Kagawa, O. & Kambayashi, Y (1997) *Advanced database functions for distance education system: VIEW classroom* Proceedings of the 1997 International Symposium on Database Engineering & Applications