A SYSTEM FOR CREATING LECTURE VIDEO CLIPSHOWS

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Greg Logan

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

This research achieves two main goals: First it proposes a set of extensions to the existing Opencast Matterhorn lecture video capture system, which should enhance its effectiveness and enable the collection of fine-grained datasets for further research. These extensions allow users to quickly and easily create, find, tag, annotate, and share 'clipshows' of their video recorded classes both publicly and privately. Second, the tracking data generated when users create or view the clipshows using these extensions are used to analyze the efficacy of the system.

Acknowledgements

This work, while bearing my name, was made possible by a cast of people much larger than I have space to describe here. Firstly, I would like to express my gratitude to my supervisors, Dr. Gord McCalla, and Dr. Jim Greer, whose expertise, insight, and patience made it possible for me to complete this thesis. They are both excellent supervisors, allowing me room to work while also giving guidance when needed. It is with their help I am truly lucky to have had such excellent supervision over the course of my Masters degree. Further, I would like to thank the rest of my committee (Dr. Julita Vassileva, and Dr. Kevin Schneider), and my external (Dr. Joel Bruneau) for their insightful comments. My work is better for your efforts.

I would also like to thank the various people I have worked with over the years in the ARIES lab in the Computer Science department at the University of Saskatchewan. I want to specifically thank Chris Brooks, who I started working for as a student in my second year, got me started in the area of lecture video capture with Recollect, and served as a frustration, someone to vent with, and a guide through both my undergrad and Masters. I would also like to thank Greg Scott Johnston, who helped me bounce ideas around, kept me sane, and helped me prank Chris. Adam McKenzie, Stephen Damm, and Kristofor Amundson also played an important role in helping to build Recollect, and Matterhorn in various capacities. I also found a valuable sounding board in Craig Thompson. Finally, I wish to thank the people over at U of S central IT, whose cooperation and help made this whole study possible.

On a more personal note, I wish to thank my mother, who is the one most responsible for getting me into computer science as a field. Watching her work when I was young inspired me first to use, and then to tinker with software. This, naturally enough, led to building (and breaking?) software, something I thoroughly enjoy both on a personal and professional level. I can never thank you enough for your encouragement throughout both my childhood and my degrees. Whenever I needed a push to get through a difficult spot, you were always there, and I hope I can return the favour.

Finally, I would like to thank my fiancée Becky, whose support and encouragement means the world to me now, and always. It is with her motivation that I finally buckled down and finished this degree. Her smiles and kind words when I was feeling down brightened my days, and her patience with my procrastination and complaints was more important than she will ever know.

Contents

Chapter 1

Introduction

Increasing numbers of learners combined with limited space and financial resources are putting pressure on educational institutions to provide more with a relatively static set of resources. Many universities are facing a space crunch. There are too many learners and not enough lecture hall space to seat them. At the same time it is becoming increasingly popular to provide significant portions of the class material in an online format. This is extremely useful for both presenter and learner because each can work in a more flexible timeframe.

With the explosion of content in online learning the question becomes *How can we make learning easier?* One of several current approaches lies in making more content available online, specifically course lectures themselves. As the proliferation of online-only universities[69][51] has shown, learners are willing both to engage with their peers online and to learn at their own pace. Producing properly structured learning material for consumption online is a difficult, expensive, and time-consuming process. Traditional brick-and-mortar universities have an advantage however: They are already producing course lecture content as part of their day-to-day business activities. Technician-free lecture recording is an inexpensive and simple addition that can rapidly create the video needed to form the base of an online video lecture system. These lectures can then be used both as a primary learning resource for remote learners or a secondary resource for review for assignments and in preparation for exams for face-to-face learners.

Making lectures available remotely, especially without incurring extra costs such as studio time, hinges upon one question: How easily can lecture material be adapted to an online presentation format, specifically one in which personalization or adaptation like that in an advanced educational system (AIED system) can be supported. Most class presentations can easily be captured with current lecture capture facilities with minimal information loss. Capturing the presentation itself can be done either with software on the presentation machine, or with a dedicated piece of hardware. Dedicated hardware has the advantage of being between the projector and any other input devices. This allows the capture to include things like notes written on the Sympodium, or the output of a document camera. Likewise, video of the instructor and/or learners can be captured easily using standard video cameras attached to the presentation machine (requiring additional software and processing power) or to an external capture device such as the Matterhorn Capture Appliance[67] or Galicaster[59]. These dedicated hardware devices have an advantage in that they are usually an all-in-one solution which will capture not only the presentation and the audio but also the presenter's camera and/or a second presentation stream as well.

All of this work raises the question of why. Why do we want to do all of this extra work so we can capture these lectures? Why are they important? Learners are usually expected to come to class, whether the recorded lectures are available or not. Is giving students the ability to watch class presentations later encouraging them not to attend? There are a number of reasons why lecture capture is important. First and foremost is for those learners who, for whatever reason, could not come to class, for examples students who are distance learners or learners with schedule conflicts. These learners can then review the lecture material at home on their own time[5]. This can reduce the load placed on the presenter and their support personnel by learners who did not attend class. No studies have shown significant declines in overall class attendance, and [11] contains a comment from a presenter that "[captured lectures] also played a part in lowering the attendance of learners who were largely disruptive in class, [and] improved the classroom environment."

In some cases lecture capture is the only feasible way to make a class available. One of the current popular themes in pedagogy and online education is the massive open online course (MOOC)[45]. These courses consist of large body of students (anywhere from 100 to 100,000+) in an online-only environment. A single 100,000 person class is likely larger than the current student population of many universities, and is likely tenable only in an online context. These online courses require instruction, which is best accomplished using the presentations already being made for the in-class environment. This presents an obvious opportunity for classroom capture. Even smaller, non-MOOC level educational videos are becoming increasingly popular. Groups like the Khan Academy[1], TED talks[68], and even YouTube have vast repositories of useful video which learners are increasingly using for knowledge acquisition.

Lecture capture is very important for the research side of computer science, especially in the fields of attention metadata, user modelling, and personalization. Three common features of these capture systems include tagging, annotation, and bookmarking. Tagging involves marking a section of the video with a single word, or short phrase which can then be searched and indexed by the system in tag clouds similar to those found in [7]. These tags can then be used by other learners to find relevant material when, for example, completing an exercise. Annotations are similar to tags in that they are used to mark sections of the material, but are typically much longer and are designed around note taking. Bookmarking, on the other hand, is a tool which learners can use to mark sections of the video so that they can more easily return to them.

The purpose of the tagging and annotation tools should be somewhat self-evident. They are a way for learners to take notes which are associated with a particular point in time in the lecture. Some systems allow the sharing of tags and annotations, so these tools can also be extremely collaborative. Bookmarking is a natural extension to how learners would otherwise use a lecture capture playback system. When reviewing a lecture for important material for which they do not have a tag or annotation, a learner is not going to watch the entire lecture. They are instead going to use milestones which they remember within that lecture and then skip around within the material until they find the correct section. Bookmarking merely removes (some of) the search time to find important material.

Next generation systems are moving beyond these simple tools and into more complex 'mashups' or (as they are called in the remainder of this thesis) clipshow tools. We chose to differentiate clipshows from conventional mashups for two main reasons. First, the usual definition of mashup involves making use of multiple, unassociated sources. For instance, taking Google Maps data and combining it with a restaurant ratings system (ignoring for the moment that Google Maps already has this functionality). The second, and perhaps more important difference is that this research is concentrated around video, something which so far mashups have not really addressed. A clipshow typically combines one or more video recording sources into a new output video containing portions of the original source(s). While very few existing learning systems implement a clipshow tool (an exception being [23]), we intend to demonstrate that they are a highly desirable addition. When creating a clipshow of an existing lecture video, a learner can add their own commentary or annotation, interspersed with the segmented original lecture. This means the learner can ask questions in context, or even add material that they feel is important but that the presenter did not include. More importantly when creating these clipshows learners can remove pieces of the existing lecture. A clipshow editing and playback tool with a lecture capture system presents an opportunity for creation of study aid clipshows.

As will be shown shortly, learning systems benefit from the inclusion of annotation, bookmarking, and social features. Annotation allows learners to quickly and easily add their own notes to learning objects, and these notes can be shared between learners as well as aggregated to create a text-based index of the class. Bookmarking allows learners to rapidly jump to the relevant sections of the material, and can also be shared to build a learner-centric repository of useful moments within a lecture. Aggregating these across the class presents the option of building a video syllabus of covered material in a highly effective way. Finally social features play an integral part in normal classes, so naturally social features should play an important role in online lectures as well. Despite the importance of these three factors there are many systems which implement only one or two of the three components.

In effect, we wish to use the learners' annotations and tags, combined with their viewing habits and bookmarks to create a social fabric within which the learners can share relevant sections of a class in an easy and seamless way. This includes sharing the notes, as well as their bookmarks without needing to send data outside of the system. While this is important it is not terribly novel: Many systems have implemented parts of this, but no single system implements all of it. The novel aspect comes from extending bookmarking to enable the creation of clipshows and then combining this extension with user tracking.

When we speak of bookmarking in video lecture capture systems, we generally mean marking a section of the video, from time A to time B. These boundaries can be arbitrary or can be restricted to the boundaries of PowerPoint slides within the lecture depending on the goals of the person performing this bookmarking. While this is useful, especially if the learner can annotate what the bookmark is about, it is not appropriate for large-scale bookmarking; the learner will rapidly reach a state of information overload if he or she attempts to bookmark all of the relevant sections of every lecture. This state of overload happens even faster when the system allows clip aggregation and social sharing. If a learner bookmarks a section publicly then everyone can share that bookmark. If 200 learners bookmark the same section, or even different sections within the same video then the information presented to fellow learners rapidly reaches a point where even if a particular learner knows what he or she is searching for, they will not be able to find it.

Imagine a system which would facilitate the creation, storage, and distribution of these video bookmarks (clips) in an easy to use way. This system would obviously need clipping functionality, but the ability to tag and annotate clips, and clipshows would also be required. Voting, and privacy controls would also be desirable, as would powerful clipshow editing tools. A further extension of the clips forming a clipshow metaphor is the ability to create series of clipshows, akin to an episode of a television show being a member of a series. Allowing users to effectively create new content out of existing material in an easy to use manner presents many interesting research opportunities. It also becomes an extremely important feature when considered within the framework of MOOCs. Imagine a class with tens of thousands of users, and providing some kind of personalization of the material for each user. Obviously there is a need for some kind of tool to create mashups and share these mashups with (potentially) a huge number of other users. This research is a step along the path towards creating these types of systems.

The overall goal of this thesis is twofold: First, to create a clipshow editing and playback tool in Matterhorn, and second to evaluate its educational potential. This leads to two questions: The first question is *can we aid the learner in finding relevant course material within our clipshow system?* And the second, and much more interesting question is *how does the use of a clipshow creation and annotation system affect learners?* The point of this type of system is to help the learner deal with information overload, so effectively we are asking *How can we effectively reduce the information overload associated with video recorded lectures on students?* We cannot rely simply on their grades to aid us in the evaluation: There are too many confounding factors for this to be a reliable method. Instead we will use the tracking data in our clipshow playback system, and surveys given around the end of term to determine student interest and gauge the effectiveness of the changes. This thesis will attempt to answer a number of hypotheses which, taken together, help show that a clipshow and annotation system aid students in their learning.

The first hypothesis is simple: Learners who consume clipshows are more interested and engaged in the class. It has already been shown that students who consume video lectures perform better than those who do not[9]. We believe that the primary usecase of the clipshow tool will be marking the important sections of any given lecture, removing unimportant moments that unnecessarily lengthen a lecture without adding much information. Formally stated, the first hypothesis is that *students who consume clipshows will be more engaged with the class material than those who consume the full source video or watch no recorded lectures.*

The second hypothesis is a simple extension of the first: *Learners who consume more clipshows per unit time spent watching captured material will be more engaged.* The premise of this hypothesis is that, while video lecture material is valuable, watching that material in such a way that only the most important portions are presented outweighs the benefits of watching the original version. While we believe that the number of clipshows containing little to no important material will be greater than zero, we do not believe that there will be enough of these clipshows to significantly affect our results.

The third hypothesis predicts that learners who create learning material as part of their studying process achieve greater success. I hope to show that *learners who create clipshows will be more engaged and enjoy the clipshows more than those who merely consume clipshows.* The idea behind this is that learners who take the time to understand the material well enough to be able to create coherent clipshows will have (re-)learned the material, and this will forge stronger connections with the class material.

The fourth, and final hypothesis will most likely be the most difficult, and most interesting to evaluate: *Learners who create 'useful' clipshows will feel the most engaged and interested.* These learners are likely to be those who are already engaged, or interested in the class material. For the purposes of this study, the definition of useful is a clipshow which is frequently revisited by others in the class. While it is quite possible that sometimes a clipshow creation tool will be bookmarking the humorous sections or outtakes from a lecture, we believe that those clipshows will not be viewed as many times as those which are academically useful. The learners who have taken the time not only to find the interesting sections, but also the academically useful sections to put into their clipshows will be more engaged.

Chapter 2

Literature Review

The e-learning field is a wide ranging area with many interesting systems. The ideal, modern mashup system for video lectures would enable presenters to easily create not only text-based but also video-based educational media. This hypothetical system would also allow users and learners to tag, annotate, bookmark, and create clipshows with material both from within a recorded classroom video, and from external sources. Unfortunately, no current system implements all of these features.

2.1 Hypertext and Information Overload

The simplest systems for creating educational media content revolve around hypertext authoring. Systems such as Moodle[49], Blackboard[37], TopCourse and the iHelp Learning Content Management System[14] all revolve around the creation and maintenance of learning materials in a higher education environment. These systems provide facilities for presenters and instructional designers to create "hierarchically structured hypertext courseware"[18]. The generated hypertext is very much like a classical textbook: There is an introduction, the main content and then a conclusion. While this approach works well, it does not allow learners to make notes and interact with their peers. This limitation necessitates learners keeping a separate notebook, and relying on 3rd party communication mechanisms which does not provide an opportunity for the system to mine their conversations for useful pedagogical information.

Some systems combine the authoring system with a tagging and/or annotation system. Systems such as KnowledgeSea II[18] and OATS[6] are examples of this class. The KnowledgeSea II system emphasizes the use of horizontal links. These links are "from a page to associated pages that are similar, [and] that can enhance the material presented on the page, explain it differently, [or] present and [sic recte an] example."[18] The content is presented as a map in the form of a grid of coloured rectangles generated using a self organizing feature map (also known as a Kohonen map). The map is organized so that the more semantically similar a pair of web pages are the closer they become, with the closest similarity pairings residing within the same cell. The cells link to auto-generated index pages which in turn can link to both internal and external resources. Initial versions of KnowledgeSea did not allow learners to tag or annotate resources; however, later versions do[32]. These later versions allow the learner to mark their comments as praise, a problem or just a note, as well as allowing the annotation to be private, anonymous or public. KnowledgeSea uses the categories to calculate an overall rating based on all of the annotations for that content and displays it as a thermometer graphic to the learner. This can be useful when deciding which material to review because it gives an immediate visual reference for which pieces of content in a given cell have been deemed useful by other learners.

OATS takes a similar, although not as centralized approach. The system itself does not provide tools to create or maintain content. OATS allows learners to highlight, tag and annotate arbitrary pieces of the HTML document. Any page which references the JavaScript files necessary for OATS to run and has tags or annotations is included in the system. Overlapping highlighted sections become more colour saturated, and the tags are added to a system wide tag cloud as well as a personal tag cloud. The learner can then search all available tags and annotations which allows him or her to find relevant material using free-form text input (assuming the material has been tagged or annotated).

Both KnowledgeSea and OATS are primarily text based. With such systems instructors could find or make video and audio clips and embed them in the material, but this is rare and requires significant extra work. More recent systems have begun using audiovisual media as core components, but this change did not occur overnight. Intermediate systems, such as the Multimedia Asynchronous Networked Individualized Courseware (MANIC) system [54] use a mix of both hypertext and audiovisual components. In the case of MANIC the author creates an HTML representation of PowerPoint slides as well as an accompanying audiovisual component. When the learners access a given section of the material the audiovisual component is played back in lock-step with the HTML component. This allows the author to simulate a full power point presentation with, for example, bullet points appearing as they are dealt with in the presentation.

2.2 Video Capture Systems

The most recent systems have been almost exclusively audiovisual, relying less and less on text-based content (which is still available, but more frequently in secondary support systems). These systems revolve around presenting the material as closely as possible to the original lecture material while simultaneously giving the learner as much control as possible over the speed and content. Each system typically takes a different approach to presenting the material. Some systems, such as the Classroom 2000[29] transform some of the resulting capture data into hypertext rather than present it as a video. In the case of Classroom 2000 the system detects slide transitions in the PowerPoint presentations using plugins installed on the presentation computer. This is a very important feature because it allows presenters to teach as they normally would without altering their normal presentations, although it does require that the presenter use PowerPoint for their presentation. The system processes the presentation on the fly into images and HTML which allows a very rapid turnaround time between a lecture finishing and being available online. Naturally, any notes made on a given slide using a Sympodium are captured as part of the image.

The three most complete and advanced systems reviewed in this chapter are Recollect, REPLAY, and virtPresenter. These systems rely exclusively on audiovisual media and make use of annotation and user tracking systems. The virtPresenter system[42] is similar to the Classroom 2000 system in that it hooks into the presentation machine, but it does not transfer the captured output into images and HTML. Instead virtPresenter uses these hooks to detect when the slide changes for the purposes of segmentation: Each slide change marks the end of one segment and the beginning of another. The system also runs the slides through an optical character recognition engine with the goal of extracting as much of the text as possible. This text forms a searchable index to aid learners in finding relevant sections of the videos. The virtPresenter system also makes extensive use of tracking features to aid learners in reviewing important sections of a given lecture. As a learner watches a video the system keeps track of which sections, down to the second, have been seen by that learner. The system makes use of these "footprints" in both individual and aggregate (what they term social[43]) ways. Both sets of footprints are displayed on the scrubber bar (the UI component which shows the learner where they are in the video file as well as allowing the learner to change the playback position within the video). The sections which have been viewed more often become darker. This enables learners to quickly visualize which sections of the videos they have already seen, as well as what other learners are watching or have watched. In [48] they show that learners found this the second most useful tool within virtPresenter, "second only to time-based navigation itself."

REPLAY[62] advances beyond virtPresenter. It does not hook into the presentation computer which gives the presenter far more freedom in terms of choosing the presentation tool(s). Instead, the system captures the audiovisual output streams and uses optical character recognition and audio analysis to determine where to segment the presentation into slides. REPLAY also has other pedagogically useful features, mainly concerned with camera targeting for both presenter and learner. The frame containing the presenter is dynamically cropped, or the camera is automatically angled in an attempt to keep the presenter centred in the frame. ETH Zurich (the university which developed REPLAY) also has a room equipped with multiple remote control cameras to capture questions from learners. The learner must first press a button on their desk which triggers an algorithm that selects the optimal camera to view the learner's face, rotates the camera to face the learner, and turns on a microphone embedded into the desk[74]. This allows learners to ask questions and appear in high fidelity (the same as the presenter) and removes the need for the presenter to repeat the question as well as any requirement for ceiling microphones.

Recollect[11], which was developed at the University of Saskatchewan in part by the author, is similar to REPLAY in that it makes heavy use of automation, but forgoes the expense and technical hurdles of automated cameras and desk microphones. Instead Recollect takes a simpler approach using fixed camera(s) and a single microphone worn by the presenter. In many respects Recollect is quite similar to virtPresenter in terms of equipment required except that Recollect does not connect to the presentation machine. This allows the system to capture arbitrary audiovisual inputs as long as they are connected to the capture hardware. Recollect makes use of a naive segmentation algorithm which by default inserts a slide transition every 5 minutes, although there has been some research into finding better ways[10][13]. Recollect also features an ability to create annotations and share notes in the player interface. Notes are associated with a given slide with the learner's notes appearing on one side of the interface, and the group notes appearing on the other. The notes are stored on the server and are associated with the learner. The group notes presented to the learner do not show the author's name. Recollect was originally designed as a research platform, and as such has extensive datalogging capabilities built in [11]. These logging capabilities created user traces with a very similar schema to that used in [50], and were implemented with the idea of generating useful data for data mining purposes. This data has since proven very useful, and resulted in both a paper[11] and a Ph. D. thesis[9].

Finally, we come to Matterhorn. The Opencast Matterhorn project[57][44][15] is a free, open source lecture capture system. It is a joint project undertaken by thirteen universities around the world (including the University of Saskatchewan, including work by the author) and combines many of the features, as well as the development teams from Replay, virtPresenter, and Recollect among others. Matterhorn features many of the abilities of these systems including source-agnostic capture, automated slide segmentation (using a very basic image differencing algorithm), and annotation tools. There are plan to incorporate some of the lessons learned about automated slide segmentation from Recollect[10][13], however the additional code was not complete in time for this study. The video is displayed in a set of Adobe Flash components, with JavaScript based components performing all other functions. For the purposes of the project Adobe Flash was chosen both as the video playback system as well as the video codec because of its ubiquity. Flash videos are also extremely easy to stream, which lowers the barrier to entry for institutions investigating the system. Further video codec support (mainly h264) and HTML5 playback components are also planned. There are a number of other commercial lecture capture solutions such as Echo360 and Panopto. They share many features and have various strengths and weaknesses when compared with Matterhorn, however there is little research involving these projects directly. These systems, including Matterhorn, currently all lack one important feature, that of enabling a learner to create a clipshow from an existing video lecture and share that clipshow with other learners. This research, among other things, adds that clipshow functionality to Matterhorn.

2.3 ITS, Data Mining, and Learning Analytics

We move now into systems which make significant efforts to personalize the learning environment, typically referred to as Intelligent Tutoring Systems (ITS). There are many of these systems, with two large commercial successes being Cognitive Tutor[24] and KERMIT[66]. Cognitive Tutor is a general domain intelligent tutoring system, whereas KERMIT is focused exclusively on the domain of database modelling. In both cases the system maintains a user model internally, and this is used to adapt both the feedback and content returned to the user. While these systems have been shown to be effective, they have induced little change in the in-class experience for the student. This may be in large part because in many of these ITSs the learner model is not exposed to the instructor, although in other cases ([20] [21]) they can be open not only to the instructor, but also the learner and their peers. Similarly, the iHelp system ([14] [19]) keeps learner models, with the goal of matching students with similar profiles for peer help sessions. User models can also be used for direct student feedback, as in Purdue's Course Signals[70], which gives learners simple feedback (a colour), along with study tips and positive reinforcement.

While Matterhorn does not currently keep user models per se, it does generate much of the same data that would be required for such functionality. This, combined with the relatively high level of user interaction make Matterhorn a highly attractive system upon which to base both educational data mining and learning analytics tools. All of this data has, historically, been analyzed by Educational Data Mining[3] researchers, although many researchers are now referring to a very similar area as Learning Analytics[63]. Regardless, EDM and learning analytics must be considered when examining the breadth of the learning systems available to instructors and institutions today. These types of systems are becoming more common place, and ever more important as universities move toward online education. For instance, as described in [3], EDM researchers have developed methods to infer if the student is gaming the system[4], off-task[2], bored or frustrated[31], or experiencing poor self-efficacy[47]. Researchers have even been able to extend their learner models and and have some predictive models to assess whether a student is going to fail, or quit their college courses[30][61][65].

Matterhorn has some basic support for learning analytics components ([1]), however these are still in the very early stages of development. As such, we elected to capture the raw data necessary for facilitating the analysis of clipshows at a later data, but we did not actually implement any analytics code directly in Matterhorn. We feel that this data, while not immediately useful in a learning analytics context due to a lack of tooling, will be useful at some point in the future once the tools are more mature especially given the volume of tracking data that Matterhorn can generate. One possible use would be as an input in a data aggregation framework, such as the one described in [17]. Assuming the class was also making use of online discussion forums, and other interactive systems the additional data gathered by Matterhorn could be invaluable in analyzing student behaviour and achievement. Finally, we explore an extension of the general idea of user modelling. In [46], McCalla outlines the ecological approach, wherein user model snapshots are stored with the learning object. This leads to a temporal trail, where the changes in a user's model can be mined for similar trajectories as other students. While there does not appear to have been much further research, we feel that this is a promising avenue of exploration when combined with the temporal nature of video watching.

2.4 Online Video Editing Systems

Despite the rising popularity of video streaming sites, there appears to have been little research done into how to make the videos editable by end users in a thin and light manner. There are obviously many professional

[1] https://opencast.jira.com/wiki/display/mh14/Analytics

video editing tools available (Adobe Premiere, for example); however these programs require a powerful computer, and require installation, which removes them from consideration for our purposes. Instead we will focus on editing tools which have been delivered in an online fashion, requiring no installation on the end user's computer. We found a pair of examples, however none of them were aimed at the educational audience. Our first example is the YouTube video editor, of which there are now two versions. The first, named YouTube Remixer[25], was based on Adobe Premiere Express (a remotely hosted version of Adobe Premiere), and has since been discontinued. The clipping tool was written entirely in Adobe Flash, and featured the ability to add transitions between clips, captions, and still images. The second, named YouTube Video Editor[26], was also written in Flash, but was not based on Adobe Premiere. This second version has recently received an upgrade[38] to make use of an HTML5 video playback component as well, eliminating the dependency on Adobe Flash. The second version supports the same functionality as the first, however it also added the ability to overlay music over top of the videos. This system is, effectively, the idea of what we seek to create with this research: A robust system which enables users to make use of multiple source videos, overlay effects, and share the resulting content. It is not, however, focused on the educational domain, and as such has some limitations. Obviously, YouTube's business model relies on repeat views, so any clipshows created must necessarily live on YouTube exclusively. Likewise, due to YouTube's varying licenses, some material can be used in this editor, and some cannot. For instance, a music video (restrictive licensing) cannot be mixed into a NASA video (Creative Commons licensing). A university hosted educational system system would not necessarily have these restrictions. Our second example of a clipshow type system is [33], where a group of students from Purdue created an HTML5 based video editor called Easyclip. The research in [33] focused on a comparison between the then-current Flash implementation of YouTube's video editor and the students' HTML5 based implementation. Their paper focused exclusively on the creation of a clipshow, rather than any of the other auxiliary features, and showed that their HTML implementation was preferred over the Flash implementation. While there are many other video mashup tools available, none of them appear geared towards education.

In this section we have demonstrated a need for systems to handle information overload, systems to handle video in the classroom, and systems to enable users to edit and share video. Our research proposes to combine all of these things. Classroom capture generates hours of information, but when a learner is searching for a given piece of information he or she can be overwhelmed with the number of available videos, much less finding the content within the correct video. Clipshows will enable user to self-organize and create their own videos based on what they, or their peers, need to know in a convenient way, rather than waiting on a top-down explanation from their instructors.

Chapter 3

Approach

The big question outlined in the previous section is simple: *How can we effectively reduce the information overload associated with video recorded lectures on students?* To explore this we developed a 'clipshow' tool for the Matterhorn system. This tool not only enables learners to bookmark sections of the video as they do in other systems, but it also enables the arrangement of sets of bookmarks into clipshows that play in sequence. Learners are able to cast votes and annotate the clipshows from within the same playback environment. Clipshow creation is similarly built into the playback environment. In this case the user can select start and stop points for their clipshow's constituent clips, and then order or repeat these clips as needed when saving the final clipshow. Intuitively clipshows can be very useful for learners because they can watch a lecture once, mark the sections which they find relevant, and then review only those sections as a clipshow of the original lecture for later studying purposes. Learners can even create multiple clipshows per lecture, e.g. one clipshow for studying purposes, and another which contains material that would be relevant for an assignment. These clipshows are stored as metadata (start and end times) rather than creating a new file consisting only of the required sections. This reduces server and storage load and enables fast and easy changes to existing clipshows. Tags and annotations from the source media appear at appropriate places in any clipshow; however, tags and annotations on the clipshow do not propagate back to the source media unless explicitly marked public. This prevents private annotations from cluttering up the public media or being taken out of context.

3.1 Clipshow Issues

Clipshows, like other learner interactions with the lecture, present many interesting issues which the system needs to handle. Firstly we must explore control: Who controls which learners can see a given clipshow? Like tags, annotations, and bookmarks these clipshows need to have access control settings. Allowing learners to create both public and private clipshows enables them to customize their review material to their own needs, without necessarily exposing their needs to their peers. Likewise, allowing public clipshows encourages learner engagement in creating useful clipshows of their classes to share with others. This can in turn foster further engagement by stimulating discussion around the content using the annotation system. The ability to share private clipshows would also be extremely useful, with learners forming study groups and sharing material that is relevant to the group without sharing it with the entire class.

The definition of a useful clipshow is in and of itself an interesting research question. For the purposes of this thesis I will define an academically useful clipshow as one which is both viewed many times, as well as voted to be useful in one or more of the academically relevant voting dimensions. This will hopefully prevent popular, funny, but non-academically oriented material from being determined to be useful, while also filtering out situations where a small cadre of dedicated students votes for a clipshow which was largely ignored by the class as a whole. While this definition is a useful first step, it is not infallible. A large number of students could vote that a clipshow is academically useful despite it being largely useless. This necessitates a final filtering step where the researcher will briefly examine those clipshows marked as useful and determine whether they are actually academically relevant. Clipshows which are not determined to be academically oriented will still be considered, but will be broken out into their own category when final data analysis occurs.

A more important issue around clipshows is determining how learners are using them, both in terms of creation and viewing. Can we use the tags of the clipshow, or the source media itself, to find out why a section was included in the clipshow? Why is a given clipshow popular? To begin to answer these questions we must start using the user tracking data. As part of some preparation work for this project, I added a user tracking system to Matterhorn. This tracking system is very similar to the existing Recollect tracking system, except that the data can be exposed more or less in real-time in the form of footprints similar to those in virtPresenter. The Recollect tracking components track how the user interacts with the user interface (slides, annotations, scrubber bar, etc) as well as what the learner is watching, when they are watching it, if they are watching, and where they are watching from. This allows us to come to some very detailed conclusions with regard to a given learner's behaviour.

Using the tracking data we can make some educated guesses around how useful a clipshow is considered to be by the learner cohort as a whole. Clipshows which are viewed often, and repeatedly are likely to contain academically relevant material. Clipshows which have large numbers of rewind operations are also likely to be academically relevant. Likewise, longer clipshows are more likely to be review clipshows, whereas shorter clipshows are more likely to be targeted at specific subtopic within the lecture. Using the tracking data and the clipshow lengths we will be able to determine how useful a given clipshow is to the learner cohort as a whole.

The above, however, is very simplistic in that it only looks at a few of the available dimensions. Modern video playback systems include features like voting, tagging, and annotation. Using these extra dimensions will give us the ability to look deeper into how learners are using the system. Votes in certain dimensions (e.g: 'Academically Useful') can be heavily weighted when evaluating how learners are actually using clipshows. Annotations and tags, however, do not have have the same reliability. While we can give the learners a preset vocabulary with which to tag, we cannot force them to use it. In the same vein, attempting to automatically parse user comments would be extremely difficult even with modern techniques. Therefore, while there will be a set of suggested tags that we will be looking for, users will be able to tag as they please. This means that tags cannot be weighted as heavily as viewing or voting for a clipshow.

3.2 Clipshow Capabilities

Clipshows would, for the purposes of the system, appear to be a completely separate lecture. There would of course be a link back to the source material, but a public clipshow could be embedded or shared in the same manner as any other full fledged lecture video. While this is not terribly novel it does create an excellent research opportunity. Assuming that the learner must first log in to the system, we can track not only who watched which lecture or clipshow, but also from what page. This data can then be fed back into some social clique detection logic. This feedback loop could aid the detection of learners who routinely create clipshows that are more useful to their peers (e.g: a personal blog with links to high value clipshows and explanations). In theory this data could even be used offer a rating/ranking system for clipshows presented to users; however, this is outside the scope of this thesis.

Clipshows can also be organized into series, much the same way as the source videos can be organized into series by Matterhorn. These clipshow series can contain clipshows from any class, across one or more source captures. In effect, a series is a clipshow at a larger granularity. Playing back a series causes each clipshow in the series to be played back-to-back, in order. This allows learners to create many different types of series. For example, a summary series of an entire class could consist of all of the summary clipshows for each lecture. The clipshows included in a series also need not be authored by the same user: as long as the author of the series has permission to view the clipshow then it can be added to the series. This brings up some interesting collaborative work scenarios, such as top learners taking turns creating the summary clipshow for each lecture.

These series will also likely have an effect on the popularity of the clipshows themselves. Clipshows which are included in a series are more likely to be watched, and this may lead to certain clipshows gaining popularity while others which are not part of a series may be left behind. We will be able to look at the influence that authorship of a series vs a clipshow has on clipshow popularity using the tracking data. The tracking system will be able to distinguish between playback of a clipshow in a series and playback outside of a series.

3.3 Potential Contributions

So far we have outlined the contributions to learning for learners, but there are also many contributions to both lecture capture and the field of advanced learning technology. For lecture capture these additions offer efficiency gains. To create a review lecture the instructor need no longer give an entirely new lecture of the same material. The instructor only needs to create a clipshow (or series) containing the relevant material from lectures which they have already recorded. Likewise, questions posted as annotations on a lecture can be answered as annotations on the lecture, preserving the exchange for learners who reach that material and wish to ask the same question later. Finally, the clipshow functionality itself could be used to craft answers to questions posed by learners. For example, if the learner asks a fairly involved question the presenter could create an answer from their pre-recorded video which includes all of the necessary context and explanation. The contributions to advanced learning technology are more research oriented. Not only will the user tracking framework be useful for future researchers, the data itself will be useful for things like recommender systems. Imagine, for instance, a system which takes the tracking data, merges it with the learner's grades and then automatically suggests content which would be relevant to the learner, as well as those learners with grades 5% higher. This would silently push the learner towards ever more important material without their knowing that the presented suggestions were ever anything other than random. This idea, however, is beyond the scope of this thesis.

Within the scope of this thesis is the creation of a clipshow building and viewing tool, fine grained data collection on learner actions with these tools, identification of useful clipshows, and studying connections between clipshow use and learner engagement. More on learner engagement will be presented in the analysis chapter.

Chapter 4

System Features and Implementation Details

Implementing clipshow functionality into Matterhorn was a three step process: First the logging components for Matterhorn were built, followed by the clipshow playback components and then finally the clipshow creation components. This order was chosen because of the implementation and priority of the tasks involved.

4.1 Logging

The logging components were the simplest to build (and indeed were already completed as part of another project by the author). The logging components consist of JavaScript hooks within the Matterhorn playback user interface which send signals back to the playback server when the user triggers certain events in the interface. These events are sent via HTTP POST[34] to the Matterhorn server in real time, which captures these in the (in our case MySQL-backed) database. The logged events include the obvious, such as starting and stopping playback, as well as the not-so-obvious such as changing the volume and opening the keyboard shortcut display. While the latter logging events may not seem relevant, the overall goal for this research logging system is to be able to reconstruct the user's session from beginning to end as exactly as possible without requiring installation of any extra software. The major drawback to this approach is that if the user goes offline and the JavaScript hooks are unable to communicate with the network, then the data captured during that period is lost. There is no facility to buffer these events, but we feel that the frequency of people going offline will be minimal considering the delivery mechanism of the video. This data is in no way exposed to the user, and is not even exposed to the administrator of the system. The only access to this trace data is available via direct database dumps. For our analysis we implemented a number of tools to render the data in more understandable formats. These tools were implemented in Python, using the Matplotlib graphics library.

The major goal of these logging components is to track which sections of the videos a given learner has watched, as well as how many times he/she has watched them. This data, specifically rewatched sections, is important because it gives us a very strong hint for important material within a given video. Sections that are watched, and then rewatched many times will no doubt contain something of interest; the difficulty lies in determining why the section is interesting. This evaluation can be aided by examining other learners' viewing habits: if the entire class rewatched the same section several times, then likely it is something of academic importance. If only a few students rewatch that section then it is more likely to be a gaffe or humorous comment. The users are also able to tag sections of the video using the built-in annotation components which we hope to use in our analysis of learner behaviour. Tags can also be applied to clipshows as a whole during the clipshow saving process. Clipshow series derive their tags from their constituent clipshows.

4.2 Clipshow Playback

The playback component (see figure 4.1) is based on the existing Matterhorn playback system. The current playback system has all of the expected pieces of a modern playback system, as well as many of the features of its parent systems such as automated segmentation, playback from these segments, optical character recognition of the contents of the segments, and search over the recognized text. The existing playback components were an excellent base which was easily adapted to be able to play clipshows. Playing a clipshow is very similar to playing a normal video; the only difference is the addition of a piece of metadata. This metadata consists of the start and stop points of each section (clip) of the clipshow. Playback begins not at the start of the full video, but at the beginning of the first clip. At the end of that clip the video will automatically skip to the beginning of the next clip and continue playback. This process is transparent to the end user, and the clipshow plays back as if it is one contiguous file. The key to the clipshow idea is that creating a clipshow is a very light operation: the only files created or modified are the clipshow metadata files. This means that the video playback component needed to be modified to take this into account. There are two obvious ways of accomplishing this. The first, and simplest, is to use progressive download (the default for Matterhorn), and the second is to use an actual streaming server such as Red5 or Adobe Flash Media Server (FMS). Progressive download is the method large scale sites like YouTube use to deliver their content. The client downloads the content (hopefully) faster than it is played back and then can instantly skip within the already downloaded content but must buffer for much longer when skipping to a point outside the buffered content. Streaming servers like Red5 or FMS deliver the content in a just-in-time fashion which means that regardless of which point the user jumps to in the video there will be a small delay for buffering. This delay is typically much shorter than the delay incurred for buffering in a progressive download system. While Matterhorn's playback tool supports both download types out of the box, the initial implementation of the clipshow playback tool used the streaming model. This choice was made because of the format of the clipshows themselves. If we could assume that clips would be within the already buffered sections, then there would be no problem; however, there is no such constraint. This means that a clip can end at any point, and jump to any other point in the video. To properly deliver content for this experiment we had to make use of the more dynamic delivery made possible by a streaming server. This also simplified our work with regards to Matterhorn's encoding codecs. By default, Matterhorn encodes to a VP6[52] codec, in a FLV[64] container. Since these are common codecs, there were no modifications required to get streaming working via Red5 with the default Matterhorn encoding parameters.

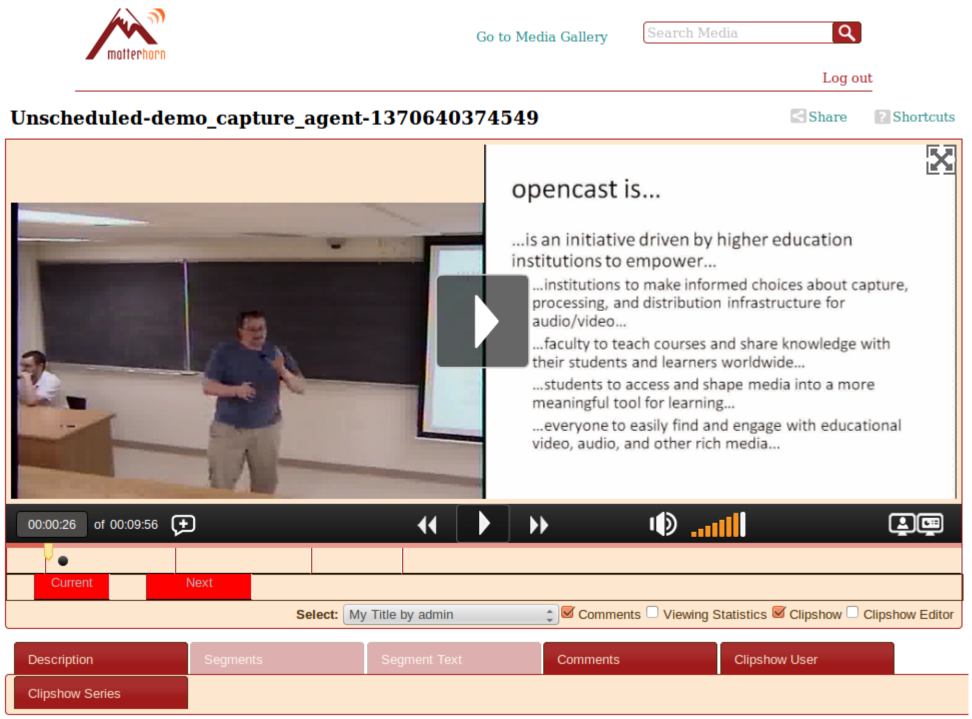


Figure 4.1: The playback interface during clipshow playback

While playback of the clipshow itself is a relatively simple affair using the Matterhorn playback tool, the method of presentation bears some investigation. The current implementation layers new JavaScript and HTML based UI elements into the existing playback tool which show the user where the clips are in the source video, as seen in figure 4.2 (a breakdown of the new components follows shortly). In effect, a user viewing a clipshow right now has a second entity watching with them (the clipshow playback script) which seeks within the video at the appropriate time. The implementation at this stage of the research makes navigation within the clipshow a somewhat transparent process in that the clips have titles ("Next", "Current", "Last") and colour coding which changes as the clipshow plays. Our original plans for a second playback mode where the clipshow is presented as if it were a full video remain as future work.

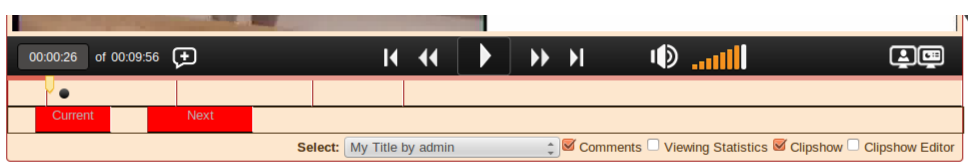


Figure 4.2: A clipshow, midplayback with an annotation dot.

4.2.1 Clipshow Voting

Users are able to vote (see Figure 4.3) on the clipshows along several different dimensions. The current implementation consists of the funny, useful (unspecified whether this is academically or otherwise useful), and dislike dimensions. The dislike dimension is mutually exclusive of all of the other, positive, dimensions. The votes are used when users search for clipshows. The generated lists of clipshows are assumed to be quite large, and would need sorting. These voting dimensions provide logical and easy to use voting dimensions. The sorting tools allow for both positive sorting, where the clipshow with the highest number of votes rises to the top, as well as negative sorting, where the clipshow with the lowest number of votes rises to the top.

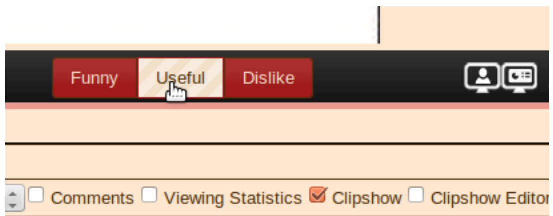


Figure 4.3: The voting interface.

The voting interface is only shown when the user is in clipshow playback mode, and the author of the clipshow is not the current user (ie: the user cannot vote on their own clipshows). This is presented as a button near the playback controls which opens a pane with a button for each voting dimension (currently 'funny', 'useful', and 'dislike', see figure 4.3). The clipshow toolkit will also add two tabs to the bottom of the player. These tabs are the clipshow search, and clipshow series tabs.

4.2.2 Clipshow Implementation Issues

Clipshows are not, however, true first class objects. Matterhorn's content indexing system (called "Search") currently has a poor API that is well beyond the scope of this thesis to repair, and will not easily accept new fields (eg: the clipshow ID). We endeavoured to work around this limitation by adding a listing of the number of clipshows for a given source video to the indexing system. This would be quite interesting to study (Do students choose the clipshow by a given author more or less often than the source media by the professor?), but given the difficulty and the timelines of this project we did not have the opportunity to truly fix the search API for this study. For the same reasons clipshow series do not appear in the search index.

Because of the thin and light nature of the clipshow, the normal video segmentation process that Matterhorn uses to attempt to automatically determine topic changes, does not map into the clipshow. This process analyzes the entire length of the source media for points at which the video signals change sufficiently[10]. Due to the nature of the clipshows we cannot reuse the existing segmentation data, and segmenting the video clipped out of the source video would require far too much computation and storage. This is especially true given that many clipshows could be created for each source video, each of which would require a separate segmentation operation. Matterhorn as a project is known for being extremely resource intensive, and adding the load of segmenting all of the clipshows is just not currently feasible. The users are still be able to see and use the segmentation; however, when they select a segment they will exit the clipshow playback mode.

4.3 Annotation

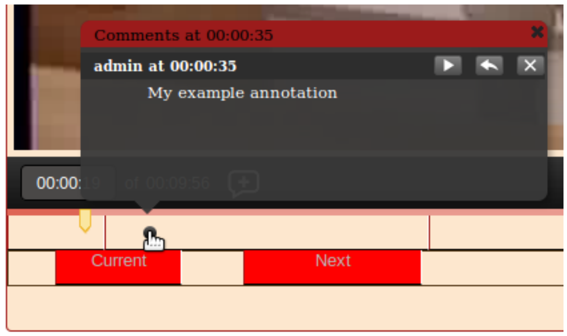


Figure 4.4: An expanded annotation.

Matterhorn's playback tool includes an annotation tool that was modified to work with the clipshow tools. These annotation tools allow users to create annotation threads at arbitrary points within the video. The current Matterhorn implementation has no concept of clipshows, but is otherwise quite capable of providing both tagging and video comment features. The existing annotation system required only minor modifications for clipshows. The main modification is a new field for each annotation which will contain the clipshow's ID, as well as a flag to set the annotation as either public or private. The field containing the clipshow ID will allow the system to show annotations made on the clipshows to be presented only to those who are watching a clipshow, which prevents confusing annotations from ending up on the source media. Annotations present in the source material for the clipshow are always displayed. Private annotations, regardless of whether they are on a clipshow or on the source media, will only appear to the author. This will allow an author to make private notes which could be used, for example, to mark interesting sections of the video in preparation for creating the clipshow later.

4.4 Clipshow Creation

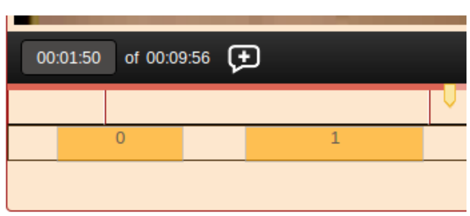


Figure 4.5: The clips on the clipshow timeline.

The creation of a clipshow is a simple affair. Once the user has enabled the clipshow creation tools two buttons appear which allows the user to add clips to the clipshow, and save the clipshow respectively. These clips appear below the scrubber bar, and are by default 1 minute long (see figure 4.5). The clips can be

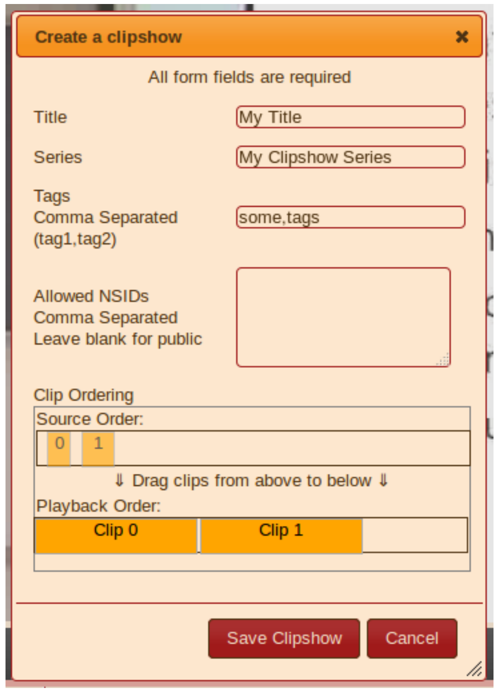


Figure 4.6: The clipshow saving interface.

lengthened or shortened by dragging the right hand side of the clip. They can also be repositioned within the source lecture by dragging the entire UI element. This has proven to be somewhat natural, although this presents some issues for users who wish to have to-the-second clipping accuracy. When the user selects the save button, a dialog pops up, prompting them for a title, description, tags, and set of allowed users (see figure 4.6). This set of allowed users (selected via their University of Saskatchewan student login (NSID) in this case) are the only users which are able to see the clipshow; it will not appear at all to other users. This allows the author to control who, exactly, can interact with their clipshows in a simple way. The dialog also contains a representation of the source clips as well as an empty area. The user must drag at least one of the source clips into this area to save the clipshow. Dragging multiple clips into the area will add all of them to the clipshow in the order in which they entered. Clips can be repeated by dragging in multiple copies. Once in, the clips can also be reorganized by dragging them within the selection area.

4.5 Finding Clipshows

Figure 4.7: The clipshow user tab.

The clipshow search tab (figure 4.7), by default, offers the user a random assortment of clipshows for the current video. The user can elect to search by title, author (both NSID and handle), or tags across all available videos or just the currently playing video. The user can also sort by one of the voting dimensions, which will reorder the results based on that criterion. The tab, due to space constraints, will display a maximum of 15 clipshows, so this ordering can be very important depending on the number of clipshows for the lecture. The tab also has a field which the user can use to change their handle. The user's default handle is their NSID; however, should they wish to change it they will be able to. All user generated data is linked back to their actual login name, so permission and annotations maintain ownership across handle changes, and abuse can be tracked back to individual users regardless of handle.

4.6 Clipshow Series

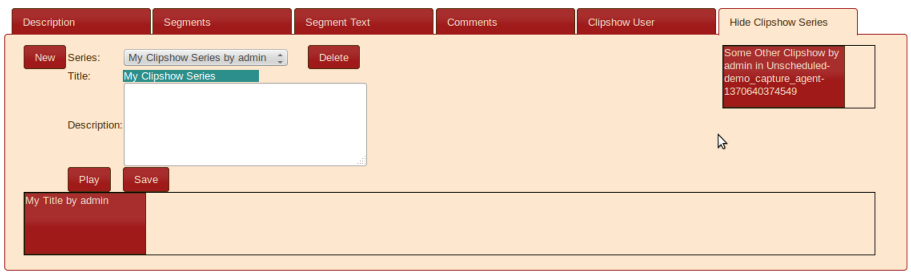


Figure 4.8: The clipshow series tab.

The clipshow series tab is similar to the clipshow creation pop-up. The tab allows the user to load existing clipshow series, or create their own. The tab has two areas: one contains the clipshows which are not already in the current clipshow series, the other is a horizontal film-strip style list of clipshows that have been selected for the clipshow series already. Users can drag and drop clipshows between the two to easily add or remove clipshows from a series. Series are played back in a similar manner as the underlying clipshows: There is a start/ stop button with which to start and stop playback of the series, as well as 'New', 'Save', and 'Delete' buttons for the obvious functions. When playback is started, the UI will switch to the first clipshow in the series and begin playback of the associated clipshow. At the end of that clipshow, as long as the series is still in play mode, the UI will switch to the next clipshow and will switch source videos if necessary. Each clipshow series has a title and a description. The tags for the series are the union of the author's tags for all the clipshows.

4.7 Code Details

Finally, we come to the transmission and storage mechanisms for the clipshows and their associated data. While Matterhorn makes use of a mix of JSON[28] and XML[71], all of the new tools we have created use JSON for communication purposes. This provides a robust, and easy to parse transmission language, while still maintaining readability. This data, and all our UI modifications make use of jQuery[60] for parsing and to power the user interface. On the server we added code to an existing Matterhorn bundle to ease future integration should we have time to polish this code. This also means that our server side code can make use of JAX-RS[40] for automatic marshalling of the incoming and outgoing JSON. This server side code also has access to the Matterhorn database, which is where we store our clipshows and their associated content. This is done using the Java Persistence API (JPA)[39] which acts as a translation layer between the database (text, numbers) and complex Java objects.

4.7.1 Java

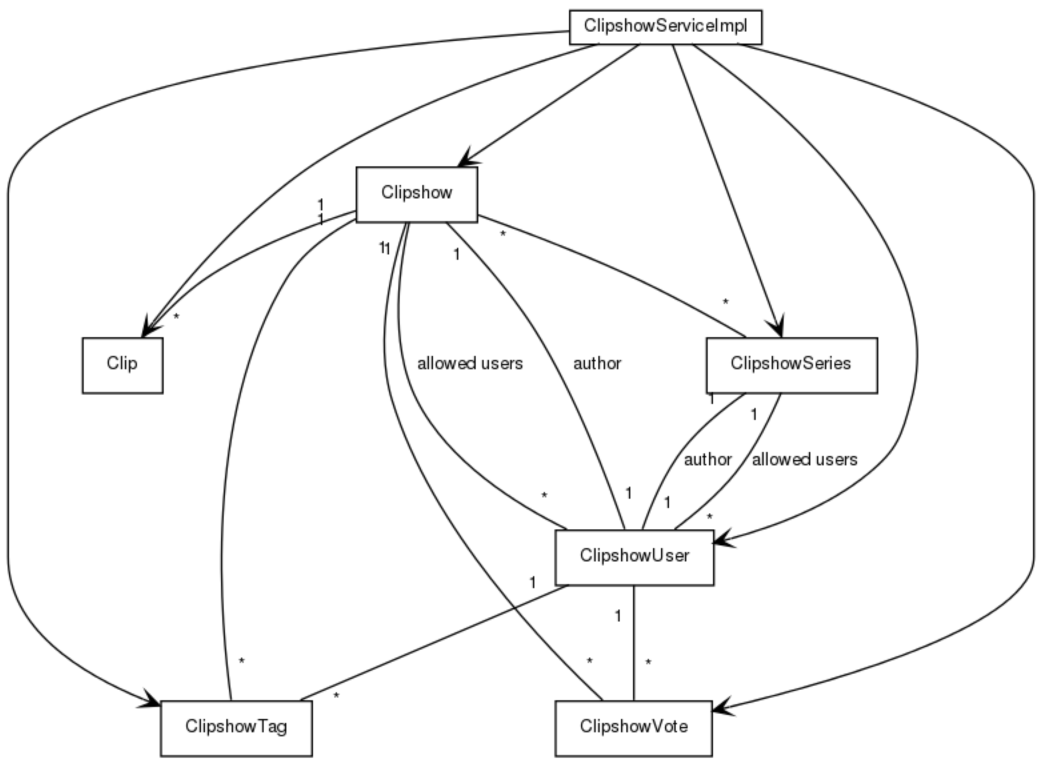


Figure 4.9: The classes underlying our system.

To keep track of all of the data in and around a clipshow, we created a number of classes, as seen in Figure 4.9. The three major classes were the Clipshow, ClipshowSeries, and ClipshowUser classes, with separate classes for tags, votes, and clips. The Clipshow class could be considered the parent of our clipshow metadata classes. It stores its title, its author's id, a flag to control if the clipshow is public, the list of allowed users, the list of clips, the set of tags, and the set of votes, as well as the source media's title and internal id. The ClipshowSeries class is simpler by comparison. It only stores its title, description, author's id, and list of clipshows. Its list of allowed users is derived from the union of the allowed users lists for each of the series' clipshows. This means that in some cases a user may only be able to see some of the clipshows, but the tools handle that by silently skipping the forbidden clipshow. The ClipshowUser class contains both the login name (for identification purposes) as well as the user's handle (for display purposes). It also contains lists of the clipshows and clipshow series that the user has authored, as well as separate lists of those which the user has access to. This allows the system to be extremely rapid when looking up which clipshows a given user might have access to. All of these classes are annotated with JAX-RS annotations, which means that sending or receiving instances between the REST endpoints and the JavaScript powering the user interface is as simple as returning an object from a function. These classes are also annotated with JPA annotations, which allows the system to transparently store objects in the database, without the need for manual serialization code. The final class in the above diagram is the ClipshowServiceImpl class, and contains the business logic to create, store, and manipulate clipshows and their relevant support objects.

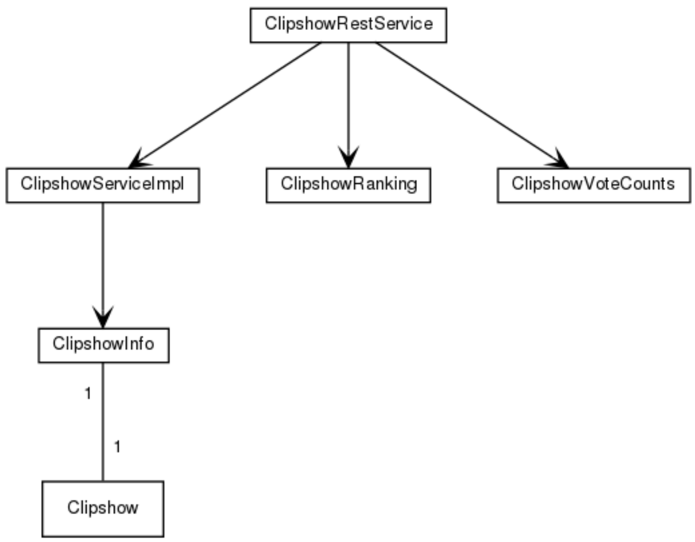


Figure 4.10: A simplified class diagram of the REST components.

In Figure 4.10, we see the class diagram for the REST components we created for this project. We have two principle classes in this case: the ClipshowServiceImpl, and ClipshowRestService classes. ClipshowServiceImpl contains the business logic which is required by the ClipshowRestService, and is an implementation . The ClipshowRestService, on the other hand, handles the logic involved in transforming the clipshow classes for transmission to the client. This includes parsing incoming clipshows, and retrieving requested data from the ClipshowServiceImpl class. The ClipshowRanking, ClipshowVoteCounts, and ClipshowInfo classes are JAX-RS wrapper classes to handle transferring information about the clipshows, without necessarily transferring the full clipshow itself. This avoids transferring the full clipshow to populate (for example) the clipshow selection dropdown in the playback interface.

4.7.2 JavaScript

The JavaScript code is based around the existing Matterhorn plugin architecture. There are four main modules, with a central library acting as a controller. These modules are playback, the editor, the series tab,

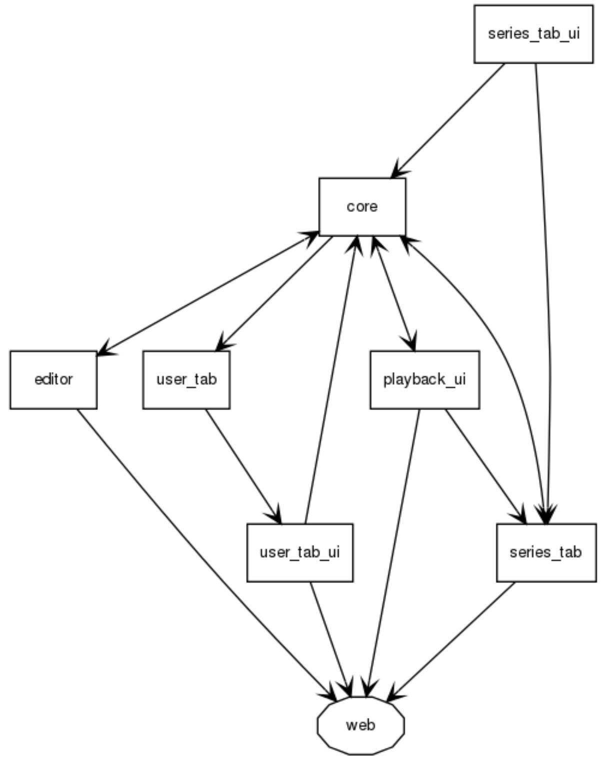


Figure 4.11: A simplified class diagram of the JavaScript components.

and the user tab. Each module is organized into a loosely model-view-presenter architecture, with an overall inter-module communication scheme as seen in Figure 4.11. Each module is also responsible for sending and retrieving its own data from the server, a design choice shared across the Matterhorn engage plugin codebase. Each of our components was developed independently, with the playback and editing functions being developed first. These two modules are the oldest code, and also the most confusing. Their presentation logic is designed to modify and overlay their components over top of the existing Matterhorn playback tools, which means that this code is also serves as the bridge between Matterhorn and the clipshow functionality. The series and user tabs, on the other hand, are both relatively self-contained. Each of these has a controller, and a user interface component. The controller is responsible for maintaining the state (viewable or not, etc) of the module, while the user interface component handles all of the business and presentation logic.

4.7.3 Communication Example

This separation of concerns helps us keep relatively clean sequence diagrams. For example, Figures 4.12 and 4.13 show the user interface, and server side sequences for saving a clipshow respectively. As we can see in Figure 4.12, the REST sequence is a set of simple user interactions, where the user enables clipshow editing mode, creates one or more clips, and then edits those clips before saving. While this diagram is simplified (there are a few internal steps that have been omitted), the overall flow of control is correct. As show in the diagram, the clipshow core controls which components are rendered, and then each component is responsible

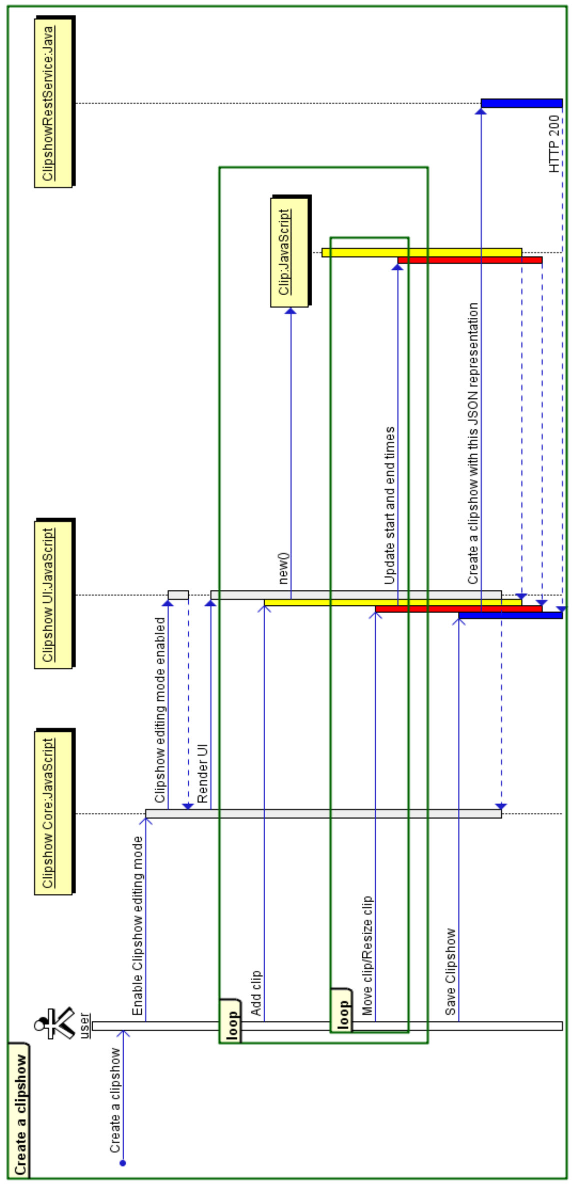


Figure 4.12: A simplified sequence diagram of the JavaScript components.

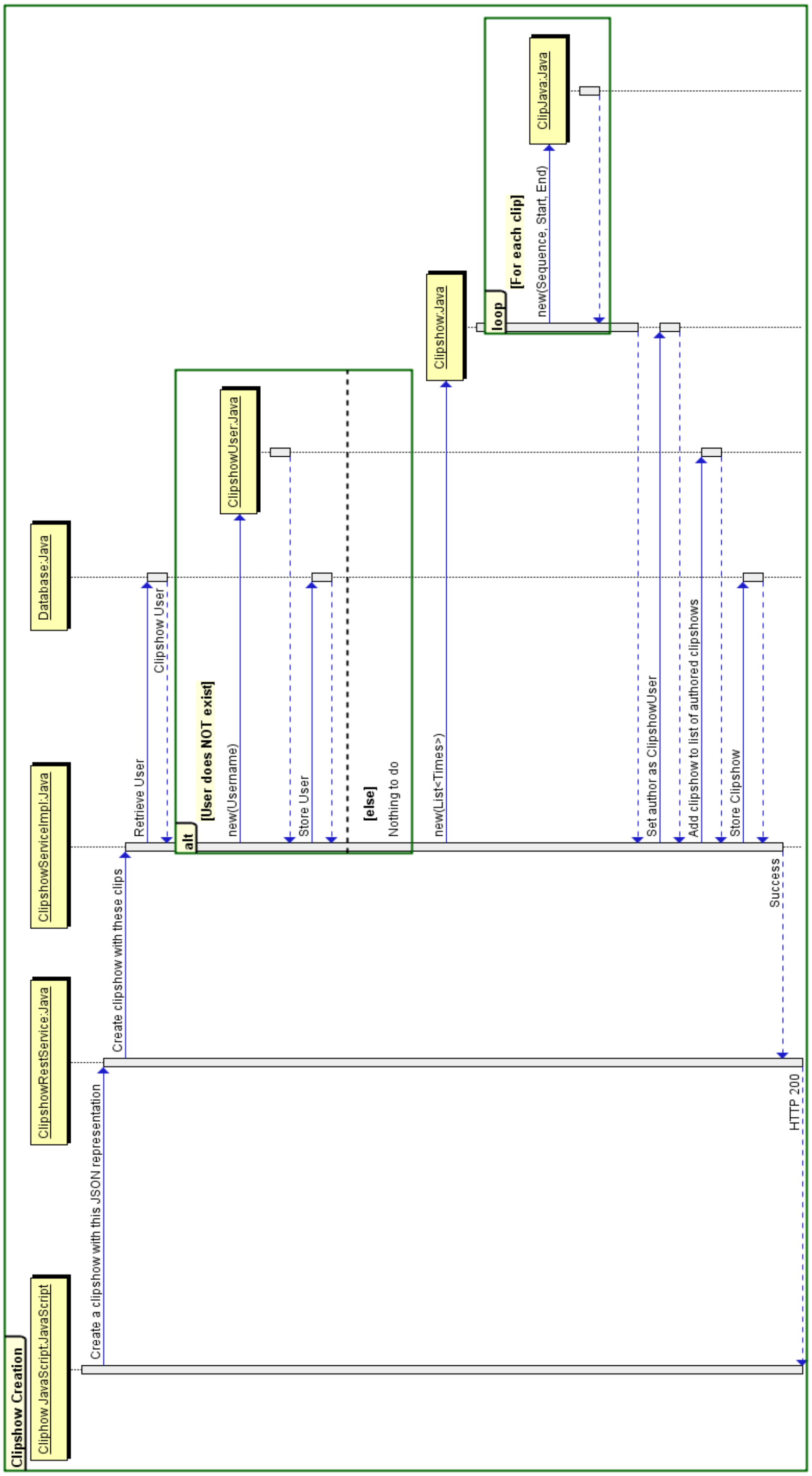


Figure 4.13: A simplified sequence diagram of the REST components.

for rendering itself. This style is shared by most of the existing Matterhorn components, so we chose to emulate that with our clipshow tools.

In Figure 4.13 we see the Java implementation of the same operation on the server side. Again, this diagram has omitted some steps (checking that the user is authenticated, for example), however it contains all of the important steps. The ClipshowServiceImpl class is responsible for nearly all of the business logic on the server side. All operations involving clipshows run through this locus of control, which also allows for easy unit and integration testing. This class is also responsible for all database communication for the clipshow components. Because all of the data classes are annotated with JPA annotations, persisting them to the database is trivial, which means we did not require a separate database access object.

In neither of these diagrams do we explore how the clipshow data is actually transferred over the network. In many cases it would be expected that some type of serialization of the data would occur, however in this case we did not have to do so. The model of the clipshow maintained by the clipshow editor is stored in a JavaScript object, which is then simply sent over the network as is in JSON form. The Java class receiving the clipshow data parses it from its string form using the JSON-simple library[41]. We originally intended to make use of JAX-RS to deserialize the clipshows, however this proved too difficult and a manual parsing routine was chosen instead.

Chapter 5

Study Design

To evaluate our tools we needed to study their efficacy with a real cohort of students. With that in mind, we sought out interested faculty, and requested access to the capture systems in the relevant rooms. We sought to study the efficacy of the clipshow tools by analyzing their effects on student engagement via surveys, as well as examining the detailed usage traces generated by Matterhorn. We planned to use the usage data to validate and check the results of the surveys. We also planned to make use of the two sets of data to attempt to draw out promising areas for future research. This would have been able to, if not validate the usefulness of the tools, then at least show that they were useful in some contexts.

5.1 The Plans

Determining the effectiveness of systems intended to support better learning can be very challenging. Most studies make use of logfile data generated by the learning environment. Less high tech approaches are still questionnaire focused, using student surveys and self-reporting of user satisfaction in a small scale cohort, over a relatively short period of time. For this thesis we planned to use NESSE's CLASSE[53] survey[1] to gather data on how engaged students felt with the class. We also made use of the automated data gathering tools built into the Matterhorn system itself. These tools record detailed traces of the user's interaction with the system, and are detailed enough that we were able to reconstruct the user's entire session after the fact.

The overall goal of this study was to determine the impact of this tool on the way that students learn, specifically in relation to the four hypotheses outlined in chapter 1. To do this we needed to be able to measure two things: How 'useful' a clipshow is, and how engaged (or not) a student is by the tool. The usefulness of a clipshow is a difficult thing to determine. We focused on the three following areas of usefulness, although there are certainly many others: academic relevance at the course level (e.g. a review of Thursday's lecture), relevance at the assignment or test level (e.g. everything you need to know for assignment 1), and relevance at a topic level (e.g. a broad overview of recursion). The easiest method of determining how useful a clipshow is lies in the analysis of the viewing and voting data. We can determine exactly who watched which clipshows, when they watched them, and how long they watched them for. Using this data we can distinguish the popular clipshows from the rest. However this does not tell us whether those clipshows are

[1] http://www.assessment.ua.edu/CLASSE/Overview.htm

truly academically useful or are popular for other reasons. Likewise, if we were to have a voting dimension for each of the areas we are interested in, we could find clipshows which have been voted to be useful but are relatively unwatched by students. The key then is in combining the two measures. There are many interesting potential questions in combining the measures: Why is a clipshow that may be marked as useful through voting not viewed many times? Why is a clipshow frequently watched when very few people have voted for it? For the purposes of our study, the reasons behind why a clipshow is useful are largely irrelevant. We are interested in whether a clipshow is useful or not, and how viewing or creating this clipshow affects the students involved. This is somewhat confounded by the clipshow series artifacts: A popular clipshow series will drive viewers to its member clipshows, but these clipshows might not all be equally useful. We suspect the number of non-useful clipshows included in otherwise useful series will be minimal, and therefore we assume the aforementioned effect to be insignificant for the purposes of our study. If we determine that this effect is significant we will be able to detect learners skipping over parts of the series using the viewing data. Likewise, if students choose not to make clipshows, and instead just seek within the video we will be able to tell by looking at the usage trace data.

As stated earlier in this document, the usefulness of a clipshow is a complex subject which is difficult to gauge. For the purposes of this thesis the operating definition of a useful clipshow is one which is viewed many times and is considered academically useful as determined via the voting mechanism. This prevents vote stuffing by determined authors, such as paying friends to vote for a clipshow which no one watched. It also prevents popular clipshows from being automatically considered useful unless users explicitly mark it as such.

Measuring student engagement is likewise a very difficult proposition. While it would be ideal for our study to be able to manually observe every student as they use the system and interview each one personally, this is not a tenable solution for research subject nor researcher due to the vast amount of time this would take for any significant number of students. Instead we must rely on surveys, which have been shown to be a reliable way of measuring learner engagement. We planned to use the CLASSE[53] survey[2], a validated online measure of student engagement in classes. The goal was to administer CLASS twice, once at the beginning of the semester, once roughly two thirds of the way into the semester. There would also be a final, paper-based survey at the end of the semester to help ensure that students actually participate (online surveys being increasingly ignored by many students, see [56] for details). This paper survey would focus on the aspects of the lecture watching experience that differ from the baseline Matterhorn experience because of the clipshow tools. The major focus of these questions will be around the social aspects of the clipshow experience. Specifically, we wish to address how the creation, consumption, voting, and modification (via clipshow series) affect the student's feeling of satisfaction with the system. Especially interesting will be the responses to questions surrounding the series components. Will they enjoy having their videos added to the highlights series? Will they want to restrict who can add their videos to a given series? There are

[2] http://www.assessment.ua.edu/CLASSE/Overview.htm

many questions which bear exploring; however, only a subset of those can be explored within the scope of this thesis. Voting on individual clipshows may also play a roll (c.f. YouTube 'likes'). Clipshows in popular series will likely attract more votes than those not in popular series. Many YouTube users seem to derive a great deal of prestige and enjoyment (and money, for popular users) from users liking or subscribing to their channels (analogous to clipshow series in our tool). We suspect that these new components would not be nearly as compelling for users if they did not have a way to gain social prestige from their activities. While these plans were well considered before the beginning of this study, the reality of attempting to run and coordinate all of the pieces meant that we had to deviate from our initial plans, as explained below.

5.2 The Reality

Our first step was a short pilot study using graduate students in the ARIES lab to evaluate the basic functionality and usability of the system, and gather feedback on both existing features as well as possible further extensions. This pilot study took place over a few days in the summer of 2012. Content for this study consisted of a short Matterhorn demonstration lecture. The participants were instructed to watch the full lecture one time, then create a clipshow. Once all of the participants had completed this step they were invited back on a second day and asked to use the clipshows to review the lecture. A survey was given at the end of each session to determine if the participant felt that this tool would enrich their learning and which parts they found most useful. From this initial pilot we found that the students thought this could be a useful tool, but needed more content to properly judge. From observing their interactions with the system we noted a number of usability issues which were fixed before the final study which provided the data analyzed in this research.

This pilot study was not, however, a truly fair evaluation of the tool. We suspect that the number one issue for real-world use of this tool in a classroom will be the cold-start problem: Students cannot watch clipshows without some clipshows first being defined, but conversely they will not take the initiative and define their own clipshows without a few bootstrap examples. The pilot study avoided this by making the creation process mandatory, but this is not tenable in a real-world classroom without investment by the instructor.

For the full semester study we decided on a two pronged approach to the bootstrap issue. Our initial goal was to provide the students with a tutorial and then if they were unable or unwilling to create the initial clipshows we would begin adding clipshows created by the principal investigator. We found three faculty members interested in participating. One faculty member taught both 2nd year and 3rd year economics classes (ECON 275 and ECON 354), and the other two were teaching separate sections of our first year computer science course (CMPT 111 01, and CMPT 111 05). The faculty member of CMPT 111 03 declined to participate. The two participating computer science faculty made their participation conditional on our system being able to share all of their videos across all three sections. We began the study in September

2012 but issues quickly arose which caused significant changes in our plans.

The requirement to share videos between all of the CMPT 111 sections should have been a simple task. Matterhorn has built-in support for user 'roles', which control access to various parts of the system. In theory, this meant that if we added all of the relevant roles to each of the sets of class video then all of the CMPT 111 students would have access to all of the CMPT 111 videos. Unfortunately, we experienced a bug in Matterhorn where only some of the students could see all of the videos, while others could see only some and others could see none at all. This issue was reported to us sporadically, and we were unable to reproduce it initially. A lasting solution was deployed after roughly a month, which we suspect caused a number of students in the CMPT 111 classes to give up on the system as a whole. This bug also required most of the time allotted to this study during that month, which meant that our tutorial on how to use the clipshow tools themselves was not posted until the middle of October. This tutorial was deployed at the same time to both ECON sections, and all three CMPT sections; however, it did not appear to stimulate much interest in the clipshow tools.

In order to salvage as much as we could, we made two decisions: We would begin creating clipshows for the CMPT 111 sections (given that the investigator was highly familiar with the concepts involved), and the economics professor was approached and agreed to add an optional assignment to his syllabus which necessitated the use of the clipshow tools. This assignment was a simple pass/fail assignment which asked students to use the clipshow tools to select the three main topics of a given lecture. A number of students created clipshows for this assignment, however they did not continue to use the tools after their assignments were complete. Full analysis of the results is included in the next section. The computer science clipshows that the principal investigator created were barely viewed, with a total viewing of roughly 5 minutes. We suspect these came too late in the term for many students to take the time to familiarize themselves with the toolset.

With these issues in mind, we re-evaluated our plans to use NESSE's CLASSE class engagement survey. This survey is geared towards measuring student engagement with the classroom materials, and we hoped that (assuming students used the system) we would be able to see differences between students who used and who did not use the clipshow tools. Instead, since very few students used the clipshow tools, and those that did, did not use them heavily, we decided instead to gear our survey towards measuring what students did and did not like about our system. Since we lacked the usage needed to fairly evaluate engagement, we felt that evaluating the effectiveness of each of our features would be the next best thing to generate the data required to show proof of concept. These surveys (see Appendix 7) were distributed both in paper form (to the economics sections), and online form (to the computer science sections). Our response rates were acceptable, although our online surveys had a much lower return rate than our paper surveys (full analysis in the next chapter). We suspect this is likely due to student survey exhaustion: The survey was sent out in the same timeframe as both the CMPT 111 final exam, and the university-level class evaluation survey. This appears to be a growing issue (see [56]), and future studies may wish to consider doing paper based surveys exclusively, or as a final step to encourage as much participation as possible.

5.3 Limitations

Before we begin, we need to first explore some of the confounding factors which likely affected the results. First and foremost were system stability issues. For the first few weeks of the academic term, our system was unstable, and therefore unavailable to some students. Particularly with the CMPT 111 sections we encountered an issue where the videos were available to some students, but not all of them. We presume that many of these students would have been early adopters had the system functioned as expected, but we suspect that many of them did not return after their first few attempts. Likewise, being that this system was a prototype, it was not as extensively tested as a real, production system. We received multiple reports of the underlying Matterhorn playback tool freezing, and other technical issues. While these may have been user error, we suspect the majority are due to bugs within the playback tools or the video streaming delivery system. The streaming system in particular had not been extensively tested at the University of Saskatchewan prior to this study. Another major confounding issue is that of tutorials. While the principal investigator did explain how to use the clipshow tools to the students in the two Economics sections, the Computer Science sections did not receive a tutorial until later in the term. A video based tutorial of the clipshow system was posted to all classes around the midpoint of the term. Moving on to less technical factors, many of the students surveyed were confused about what the tool was: Many of them thought they were using the clipshow tool when they were not! The goal behind the design of the clipshow tools was that the tools could take the feedback generated by this research and, with some code cleanup and improvements, be contributed into the main Matterhorn codebase. This meant that the tool was, by design, built into the default Matterhorn player. As will be shown later, many students believed that the Matterhorn playback system was the clipshow system, rather than the host of the clipshow tools. Likewise, several students expressed confusion over the point of the tool. These students were of the impression that the faculty, or instructional support staff for the class would be creating the clipshows rather than it being a tool for the students to use. These reasons are just some examples of why our study deviated from our initial plans outlined earlier in this document. Finally, we found that teaching style made an immense difference with regards to both difficulty, and interest in clipshow creation, which may have dissuaded some students from creating clipshows on their own.

5.3.1 Software Status Report

This research project created a fairly large volume of code, much of which was intended for contribution to the Matterhorn project once the research was complete. While Matterhorn has standards for code quality, for the purposes of this research the requirement was simple functionality. This means that the code does not have proper unit test or integration test coverage, nor is it particularly clean. In retrospect, enforcing the quality standards from the beginning would have likely yielded a more polished system, however there was insufficient time to refactor the affected code. In its current state the clipshow code is somewhat fragile, although it is also relatively compartmentalized from the rest of the Matterhorn codebase. To be able to contribute this code back (or indeed to release it publicly at all), the code would require lots of polish, and much documentation both in terms of API and functionality. While creating this documentation and adding that polish is a goal, we are unsure whether this will be accomplished in the time remaining.

5.4 Testing the Hypotheses and Approach to Data Analysis

Keeping the reality of this study in mind, we next had to determine how we would test our hypotheses. Our surveys were run only near the end of the class, so we are limited in our ability to determine student responses over time. While this is disappointing, it is not required to test our hypotheses. If we had access to this data we would have been able to examine how student responses changed as they made more use of the clipshow tools, however since students did not make extensive use of the tools this likely would not have been relevant regardless. Our tools did, however, generate a large quantity of usage trace data. It is this data, along with analysis of specific questions in our survey that will be used to test our hypotheses. Our data is relatively sparse however, so we employed an inductive data mining approach to attempt to find interesting patterns. While much of our analysis chapter deals with our initial hypotheses, we feel that the patterns found in the rest of our analysis also point out some highly interesting avenues for future work.

Chapter 6

Analysis

Keeping the reality of our study in mind, we were forced to re-evaluate our analysis methods. Our initial plans called for the use of survey data as a primary source, with the usage data being used as a backup to the surveys and a source for additional areas of future research. Given the lack of clipshow usage, we decided to treat our analysis as more of a list of interesting future research directions, rather than a strong set of evidence. We decided to use a more inductive approach, surfacing interesting patterns from our data rather than defining our goals in a top-down fashion. This study was always a proof of concept, and we feel that we have shown that this system has (or at least has the potential of) a great impact on students and teachers. We have also succeeded in highlighting several possible avenues for future research, many of which are relatively simple follow-ons to our existing code and experiments. With this in mind, we begin with our hypotheses, and then move onto our survey data and user feedback.

6.1 Initial Hypotheses

When we embarked on this project we had a simple goal in mind: Determine how the clipshow tools we were building would affect learners using the system. To attempt to answer this question, we chose four hypotheses to test, each attempting to isolate a pedagogically relevant cluster of user behaviour (similar to [9], chapter 6), in the hopes of, if not supporting the hypotheses, then pointing the way to further directions of study. These hypotheses are all based around student engagement rather than evaluations of the students. We chose this route because grades have too many confounding variables to be a reliable measure of student engagement. This does not mean that we are not using student marks; rather we chose to concentrate on other measures. Student marks were procured to add additional weight to our engagement measures. Student engagement in our case is being measured using two distinct tools: Matterhorn's detailed usage tracking tools, and in-class surveys. We have fine-grained user tracking tools built into Matterhorn[11] (a partial list of tracked events can be found in appendix C) which can generate second-by-second details of a given user's interaction with a video during a session. This means we can easily reconstruct any given user's session, and study how they viewed the videos including plays, pauses, and seeks. This detail includes the player's current position in the video, which means we can catch users who were, in effect, playing clipshows by seeking within the video. This played a key role in some of our analysis.

6.1.1 Hypothesis 1

Our first hypothesis is that learners who consume clipshows are more interested and engaged in the class. We took two approaches to analyzing our data to address this hypothesis. Our first approach involved simple marks, a summary of which appear in Figure 6.1.

Class	Number of Students	Class Average	σ	Average	
				With Clipshows	Without Clipshows
CMPT 111	289	70	21.4	84	71
ECON 275	50	72	17.7	75	68
ECON 354	26	71	11.4	71	70
Overall	365	70	20.1	74	69

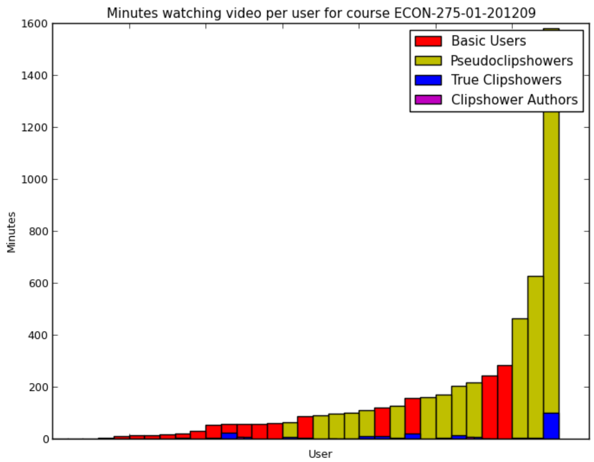
Figure 6.1: Student usage and grades. The averages with and without clipshows only count those students who used the system, whereas the class average counts all students in the class.

Class	Clipshow Users	Non-Clipshow Users
CMPT 111	0.13	1.37
CMPT 275	3.99	0.89
CMPT 354	1.29	1.29

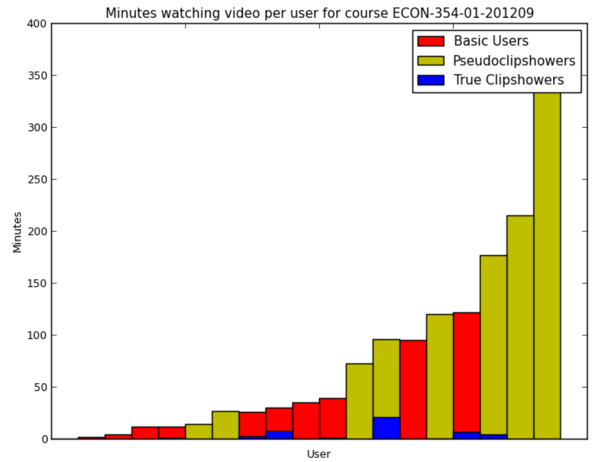
Figure 6.2: Average number of viewed hours for users who viewed any clipshows vs users who did not.

We hoped to have access to intermediate student grades so we could correlate when they used the tool with changes in their marks, however this data could not be obtained, and usage was low regardless. We instead chose to focus only on final grades. While the results show consistently higher final grades among students who used the clipshow tool, this is not statistically significant. In no case is there a significant difference between clipshow users and non-users. The correlation between marks and viewing time is encouraging in ECON 275, but still not significant. Even our most notable class (CMPT 111) is the result of an outlier: only a single student in that class made use of the clipshow tool. This becomes even more clear when we examine the viewing time in Figure 6.2. We found this highly disappointing because we had hoped to show clipshows produce results similar to [9] (specifically chapter 6), which found that "learners who watch lectures regularly have higher outgoing grades". Given the lack of student usage however, we do not have sufficient data to draw any conclusions. Our second approach involved using viewing trace data summarized for each class in Figure 6.3.

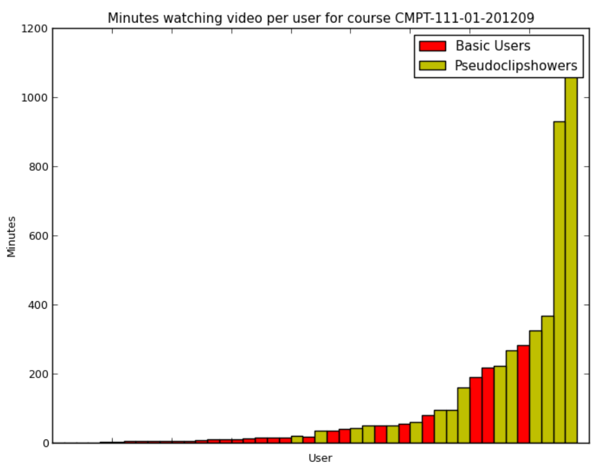
On first analysis, however, the results in Figure 6.3 do not appear promising. The *True Clipshow* category represents the number of minutes a given user has watched while in clipshow mode, with the *Basic Users* category representing the overall viewing time including both the clipshow and non-clipshow playback modes.



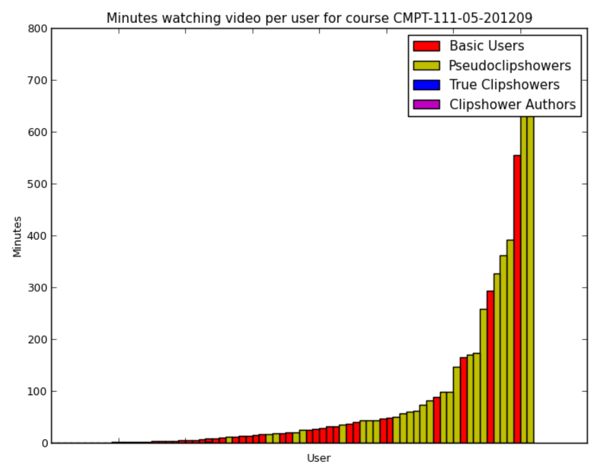
(a) ECON 275



(b) ECON 354



(c) CMPT 111 01

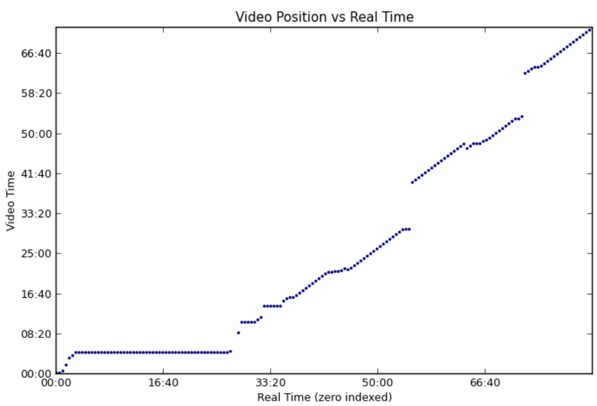


(d) CMPT 111 05

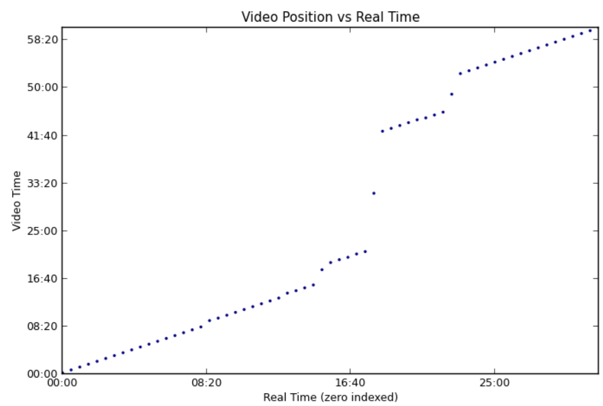
Figure 6.3: Minutes of video watched per student, broken into class sections and arranged from lowest usage to highest usage.

We will explain the third group shortly. As is clear in the graphs, very little time was spent actually watching clipshows directly. This can be attributed to a number of confounding factors discussed above.

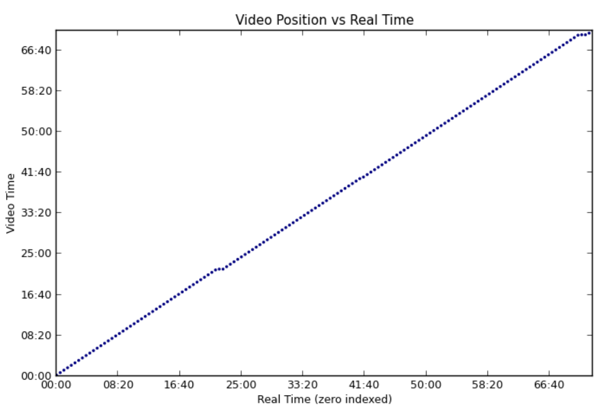
Upon deeper analysis however, we came to the conclusion that many of the users who were not making use of the clipshow tools themselves were in fact using the normal playback mode as a clipshow. We made use of Matterhorn's detailed logging to reconstruct each user's viewing session(s) with each video. This code takes the heartbeat signal sent by the Matterhorn player (every 30 seconds in this study), and graphs the user's current point in the playback. This gives us a detailed view of where the user was in each video at any given time within their playback session. This code is based on the code used to generate the graphs in [16]. This yielded many uninteresting graphs, but also a number of graphs similar to Figure 6.4.



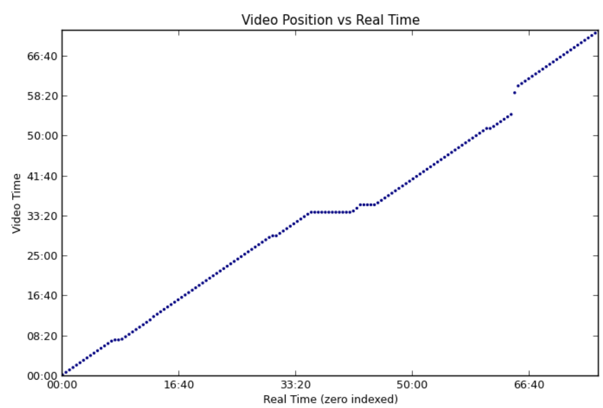
(a) Student A



(b) Student B



(c) Student C



(d) Student D

Figure 6.4: Two pseudoclipshower traces (A and B), with two normal traces (C and D)

In Figure 6.4 the actual time since the student began watching the video (wall clock time) is on the horizontal axis, and the player's position in the video is on the vertical axis. As we can see in Figures 6.4a and 6.4b, each student plays the initial few minutes, and then plays through the rest of the video, skipping forward occasionally. This is a very different pattern from traces like those found in Figures 6.4c and 6.4d, where the user is primarily just watching the video with little to no skipping within the media. We manually sorted through the viewing traces, sorting users who displayed similar skipping behaviour into a separate *Pseudoclipshowers*. To repeat, these users are making use of the normal Matterhorn playback tools, but using them in such a way that their usage traces appear very similar to a true clipshow user's. These users do not, on the other hand, always behave this way. In many cases these users act as pseudoclipshowers only part of the time, the only requirement is that they behave as one at least once. After labelling these users, when we look again at the graphs in Figure 6.3 we see that the majority of the high-usage learners are, in fact, making use of some aspect of a clipshow-like behaviour (the yellow bars). Unfortunately, because they chose not to make use of the clipshow tool itself we cannot be certain that clipshow users were more engaged with the class material, but we have a strong suspicion that they would have been had a tutorial been available earlier, with a more reliable tool. Further work is highly warranted.

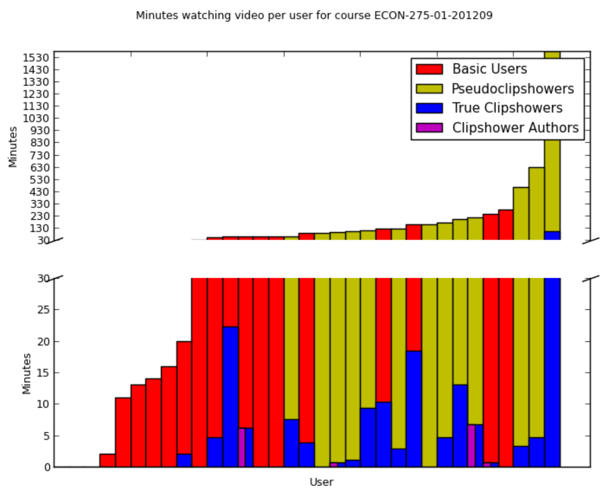
6.1.2 Hypothesis 2

Our second hypothesis is that learners who consume more clipshows per unit time spent watching captured material will be more engaged with the class material. Going back to Figure 6.3 we see very few users who actually viewed the created clipshows. For clarity, we present the same data with a limit of 30 viewing minutes.

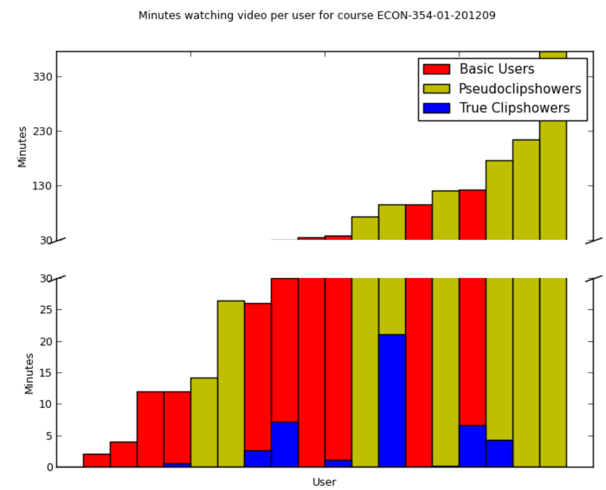
This version of these graphs makes the data much more clear: There does not appear to be any correlation between clipshow usage, and engagement. Indeed, clipshow usage seems to be more or less evenly distributed over the set of users. If we include the pseudoclipshowers, however, we see a somewhat significant bias where users who view more video appear to exhibit clipshow-like behaviour; however, we cannot quantify how much of a bias this is because we have no way of measuring how much of the pseudoclipshower's time would have otherwise been spent watching clipshows. In this case it is mainly a lack of data that prevents us from reaching any conclusions. Only about 4.5 hours of clipshow video was watched over the entire term, and the most time any given user spent watching actual clipshows (as opposed to pseudoclipshows) was roughly 25 minutes. Measuring pseudoclipshows turns out to be remarkably difficult because each viewing session of each user would need to be manually evaluated, preferably by multiple raters. This was not feasible with the resources allocated to this study, and would have been highly error prone had we had the resources. With a larger dataset, and a more effective tutorial on using the clipshow tool we suspect that more of the higher usage learners would have made more use of the tool.

6.1.3 Hypothesis 3

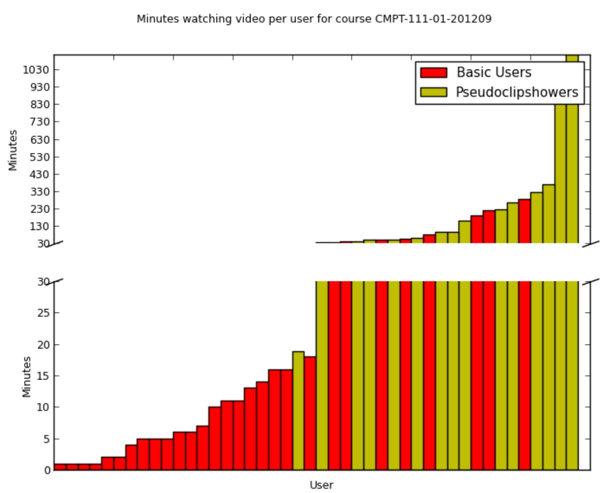
Our third hypothesis is that learners who create clipshows will be more engaged and enjoy the clipshows more than those who merely consume clipshows. This hypothesis presents a similar issue to the previous one: we lack a large enough dataset to truly come to any conclusions (see Figure 6.6). Regardless, we can report our observations and thoughts for future research. First and foremost, we must note that the vast majority of the clipshows created in this system were the product of two assignments, one in each Economics section.



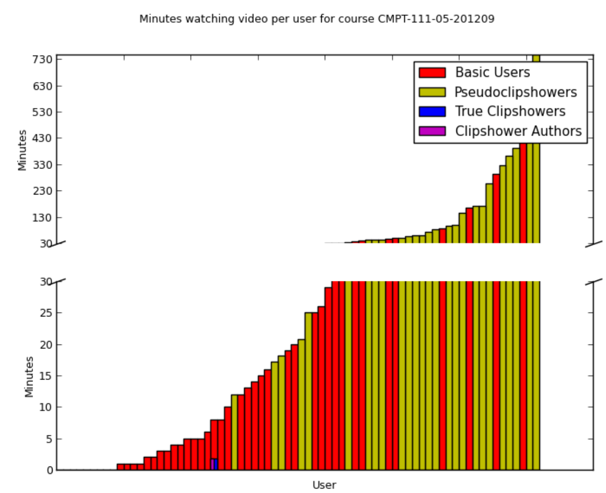
(a) ECON 275



(b) ECON 354



(c) CMPT 111 01



(d) CMPT 111 05

Figure 6.5: An enhanced view of the first 30 minutes of video watched per student, broken into class sections and arranged from lowest usage to highest usage.

The students in these classes were required to complete seven of nine assignments, and one assignment was to create a clipshow of the three most important points within a certain week's lecture. Eighteen ECON 275 students created clipshows, of which only three were not related to the assignment. Eight ECON 354 students created clipshows, and there were no clipshows that were unrelated to the assignment. No students created clipshows in the CMPT classes; however, the principal investigator as well as a teaching assistant both helped create a few clipshows to attempt to jump start usage in these classes. Unfortunately, these efforts only generated about 5 minutes of viewing time.

Class	Number of clipshow authors	Total number of clipshows	Number of private clipshows	Clipshows unrelated to assignment
ECON 275	18	30	13	3
ECON 354	8	14	5	0

Figure 6.6: Breakdown of clipshow usage related to the assignment

We found it odd that users were creating clipshows, and then not viewing them, but considering that most of the authors were only (apparently) authoring their clipshows as part of an assignment this does not surprised us terribly. Our tool was viewed as something that had to be used for the assignment, rather than something that should be used every day. Examining figure 6.5, we see that the clipshow authors are no more or less active than any other clipshow users. We posit that this is due to the small sample size, and the lack of interest or motivation to create clipshows. Interestingly, we found that for the lower level class (ECON 275) there was a marked increase in the grades for those students who turned in the assignment; however, it was not statistically significant (see Figure 6.1).

Class	Average for students who completed	Average for students who did not complete
ECON 275	73.6	65.5
ECON 354	67.4	65.5

Figure 6.7: Marks for students who turned in the ECON assignments

6.1.4 Hypothesis 4

Our fourth, and final hypothesis is that learners who create 'useful' clipshows will feel the most engaged and interested. We defined 'useful' clipshows as those which are frequently revisited by others in the class. This presents an issue for our dataset however. Very few of the clipshows were ever reviewed more than once. Indeed, while there were 27 users who watched clipshows other than their own, they only watched on average about 3.5 minutes of any given clipshow. This is heavily skewed in favour of a single clipshow which was watched for a total of 22.3 minutes, with the next most popular clipshows only receiving approximately 9 minutes of viewing. This presents an obvious problem when addressing this hypothesis in that while we do have some data, we do not have enough to perform any valid analysis. We must, therefore, conclude that further work is required.

6.2 Exploration Of Survey Data

While it is disappointing not to be able to report strong conclusions to our initial hypotheses, it does not mean that this study was in vain. Our survey data alone has provided us with many avenues for future exploration. The survey instrument (see Appendix 7) was given in class to the two Economics sections, and online to the three Computer Science sections. Our response rates were good for the in class sections; however, the online sections suffered from the normal low return rates[56].

Class and Section	Surveys Returned	Total Students	Return Rate
CMPT 111 01	16	113	0.14
CMPT 111 03	7	61	0.11
CMPT 111 05	22	115	0.19
ECON 275 01	20	50	0.40
ECON 354 01	9	26	0.35
Total	74	365	0.20

Figure 6.8: Survey Analysis

The results were highly interesting however. First, we examined the responses to question 2 (*Did you view any clipshows?*) across all of the students who responded to the study. We found that in many cases students were very confused about what exactly a clipshow is. We found 28 users who claimed to have viewed a clipshow, yet none of those users appear to have played back a clipshow at any point. Indeed, 11 of them do not even appear in our viewing traces! This speaks volumes about the study, and the efficacy of its organizational structure. In examining other survey responses, we discovered an interesting answer which may explain this discrepancy. One of the students replied to question 5 (*How did you find the clipshows you watched?*) with a statement indicating that they found them with the help of a friend who already knew how to use the system. Whether this student is confused about what clipshows are is irrelevant in this case; the important fact is that there is at least one student whose viewing traces are mixed in with another student's. Our system records the logged-in username for each event that it logs. It does not, however, have a way for students to indicate that they are not the only ones watching the videos. This means that, while we have provided features like private clipshows for study groups, they may not be making use of them because they prefer to watch a video as a group rather than using the collaboration tools built into the system. Future studies into this area must take this into account, either by controlling for the number of viewers, or by taking this into account in their analyzes. Our biggest concern with this study has been a lack of data, and that lack could potentially be explained by groups of students watching clipshows together rather than individually.

Unfortunately, we have no way to tell if this was a significant factor at this point.

The free-form response style of question 4 (*Please explain briefly why or why not*) provided us with some interesting results. For those who responded that they did not find the clipshows useful (remembering that many of these users are confused about what a clipshow actually is), most of them made some mention of how it would be useful to review material they had missed in class or had otherwise forgotten. Two respondents mentioned not being aware of the clipshow tool's existence until the presentation in their class. This was unfortunate, and further studies should be better prepared to present tutorials much earlier in the term. Two other respondents brought up a very salient point however. The classes they were in did not translate well to clipshows, either due to the material presented, or the pedagogical style of the professor. Indeed, the principal investigator had the same opinion when creating the clipshows for the computer science section. The faculty members in each of the the computer science sections took very different approaches to lecturing. In one case the faculty member spoke almost continuously, with little in the way of breaks. Indeed, even the way he structured and pronounced his sentences left little in the way of pauses. The other faculty member taught for roughly ten minutes, then allowed a work period wherein students would work in a supervised manner to complete their labs and assignments in class. This dichotomy led to very different clipshow creation experiences as well. In the former case, very little could be clipped out because each sentence depended upon the other. Most of the clips in those clipshows contain blips of the last sentence because the clipping accuracy was not perfect, and this led to some confusing moments where the player would skip from the end of a thought, to the end of another (the blip) before beginning the thought which was the point of the second clip. On the other hand, creating clipshows for the latter faculty member was extremely easy because his presentation style was already pre-divided. Very few times were any of his teaching segments broken up, and in those cases it was usually because he took questions for an extensive period of time in the middle of a teaching block. While neither of these pedagogical methods is inherently 'better', one most assuredly lends itself to video, and clipshows, better than the other.

In the past few paragraphs we have discussed primarily users who thought they used the system, and whether they were happy with it or not. Now we move onto users who did not use it, but who were gracious enough to provide their feedback regardless. There were few users who felt the system would be useful, but chose not to use neither the Matterhorn player as a whole, nor the clipshow tools. The major complaint of this group was that, while the system might have been useful, there was a significant learning curve. This was to be expected because we did not have time to do a full HCI study on the user interface itself. Instead we ran a pilot study within the ARIES lab, which is admittedly not a normal body of non-technical users. We felt the interface was usable, if a bit unpolished, but obviously some users disagreed. Future studies should take care to note that complex operations like creating clips need to be more fully explained with user interface hints and other appropriate user experience additions. Interestingly, one of the users who watched the most video was in this group; however, he or she left no comments. Another interesting point was one of the users who indicated that the videos only functioned when the user was online, whereas the power points for the class were available offline. This restriction means that the videos, while potentially useful, are only available when an internet connection is available. This makes printed power points more useful for studying on the go (for example on a bus). Unfortunately, adapting Matterhorn to this type of usecase is beyond the scope of this research. This comment appears again in the next group of users, but might be better addressed in future versions of Matterhorn. A planned feature in the upcoming Matterhorn 1.4 release is the ability to export video from the system into an easily downloadable format. One possible enhancement to this system is an ability to export the user's clipshows into a downloadable format using this tool. This would allow users to download their clipshows onto their mobile device for review on the go, and would enable several usecases for which the clipshow tools are currently unsuited.

The group of users who felt the system would not be useful, and indeed did not use the system was expected to be a quiet group. Considering the survey response rates, we did not expect to hear feedback from users who effectively chose not to take part. Instead, we received responses from 46 users (some who used the system in a minor way, some who never used it at all), with a variety of complaints. There were two major groups of comments left by these users. One group had the same complaint as was mentioned previously: the PowerPoint slides were available, and so they did not feel the need for video lecture technology at all. While this is unfortunate, there is little we as investigators can do aside from selecting study classes which are better suited to online video lectures for future experiments. Some of these users also commented upon various technical issues, either on their end ("something wrong with the internet in my home") or ours ("bad [camera] angle", "sound was fuzzy"). There were a few users who were disparaging of the system, either as an idea or as a specific implementation. Several of these users made comments similar to those found in [9]: In effect, these users chose not to use the tool not because of the pedagogical approach or technical hurdles, but because they came to class and therefore did not see a use in exploring the tool. This is to be expected, and it was gratifying to see that many of these students also felt that the system would have been useful in other circumstances (such as when they could not come to class, or the classroom environment was more conducive to video lectures in general), and bodes well for future studies. Had we been better prepared we might have been able to convince many of these students to participate.

6.3 Clipshow Feature Feedback

Our questionnaire also included a number of questions relating directly to features designed to support or make use of the clipshows themselves. One of our survey questions asked how often the user made use of the clipshow series tool, summarized in Figure 6.9. Obviously, no one made extensive use of the tool. Indeed most of the series created during our study were for experimentation. We had 9 students rate the series tool as somewhat or very useful; however only one of these users was actually a clipshow author. We saw a further 26 rate the series tool as being of little use, and a further 30 rated it as of no use at all. This may be at least partially explained by the lack of content with which to create series. Of the nine students who rated the system highly, only one of them created a clipshow series (and he created 11 of them to be exact, most with similar or identical content which suggests this system may be buggy or has a usability issue in that area). The remaining 26 series were created by other students (including 2 by the principal investigator), and have similar repeating usage patterns. These series display a uniform appearance of experimentation, with many of them named some variant of 'test', or random characters. Most of them also do not have clipshows associated with them, which is likely related to users not being familiar with the system despite the tutorial. Finally, there were only three users who attempted to use the series tool for playing back a series. Of these three, two were serious users who made multiple attempts (successful or not) to play back their own, but only one played back another user's series more than once. Of the two series that were played back by someone other than their authors, one was played back by two other users, the other was played back by only one other user. The series created by the principal investigator were not played back at all.

Class	Number of series authors	Number of series created	Number of non-test series
ECON 275	10	27	1
ECON 354	5	8	0

Figure 6.9: Breakdown of clipshow series usage

Likewise, the privacy settings wherein students would be able to create clipshows private only to themselves, or a select group, were not heavily used. As summarized in Figure 6.10, of the total of 44 clipshows created, 18 of them were private (the user had added themselves to the allowed list of users and no one else) spread over 10 different authors. Of these private clipshows, most of them appear to be draft clipshows: many of them share identical or very similar titles with public clipshows, or have titles which map to draft-type activities (blank titles, or titles like 'test'). In addition, many of the clipshows which were public were also named in similar ways, indicating that some students either did not care to use the privacy features, did not understand how to use them, or were not aware they existed. This strongly indicates the need for a draft mode, or ability to edit clipshows post-saving. The reality of our implementation timeline was that we did not have time to create either of these features, and students worked around our shortcomings using the privacy features. Future studies should ensure that one or both of these features should be implemented.

Class	Number of clipshow authors	Total number of clipshows	Number of private clipshows
ECON 275	18	30	13
ECON 354	8	14	5

Figure 6.10: Breakdown of clipshow usage related to usage of privacy features

Our questionnaire provided strong evidence that students did not use the annotation tool, which backs our viewing data. Of the survey respondents, only four responded that they had used the annotation tool. There indeed were four annotation authors; however, only one (possibly two given that one of our survey respondents omitted their username) actually authored an annotation. The majority of the comments in the survey were negative ("did not use", "i actually wasn't looking for that", "i don't really care what other students would care to say, as i am viewing them to obtain my own interpretation"), but there were a few positive instances which mentioned specifically that there was a lack of other comments ("there were no comments. would have been useful is [sic] there was [sic]", "i think i am the only one sometimes make comment there"). We suspect this is another instance of the cold start problem, where no one sees the tool as useful unless someone else has created some initial comments. We will propose a way to address this in the future work section. Clipshow voting fell along similar lines, with only two users casting a total of three votes. There were more users who thought they had voted (six) than the system registered as having actually voted (two). This may be explained by failures in transmitting the votes to the system; however we have no record of these failures and cannot otherwise explain why users believed they were voting but were not. Regardless, 26 users thought the voting tool would be somewhat or very useful, which is very large considering many of them never used it. 41 users disagreed, saying it was of little or no use. We suspect that the lack of clipshows inhibited the use of the voting tool, since the primary use of the votes is to enable the best of the clipshows to vote to the top of the search lists. Future studies should also ensure that they extend the voting dimensions. Our current implementation consists only of three dimensions, which may not be sufficient with a larger set of clipshows. Both of these observations are summarized in Figure 6.11.

	Number of students who thought		
Tool	the tool would be useful	the tool would not be useful	Number who used it
Annotation	14	48	4
Voting	26	41	2

Figure 6.11: Breakdown of annotation usage

6.4 Pedagogical Analysis

Our initial plans called for a each class to have access only to their own videos. When our CMPT 111 professors approached us with their desire to share their videos among all of their sections, we saw an opportunity to examine the impact of the professor's pedagogical style on their class's viewing habits. While there are a number of other, confounding factors (class duration, location, time), we feel that the results we found are interesting enough to warrant mention in this thesis. Of the three class sections total, two of the faculty were interested in making their lectures available to all sections (the remaining faculty member opted out of video recording entirely). While we expected some variation in both the number of students watching videos, and the total amount of video viewed, we were very surprised with the large variation we observed. In Figure 6.12 we see that each section varies in terms of student usage. We were able to publish a video tutorial about the system to the students in all three sections, but were only able to make in class presentations to the students in CMPT 111 sections 01 and 05, which may explain the relatively lower participation from the students in section 03. Whatever usage we had in CMPT 111 03 is entirely organic through word of mouth by the students.

Class and Section	Matterhorn Students	Total Students	Percentage Participating
CMPT 111 01	49	113	43.4
CMPT 111 03	17	61	27.9
CMPT 111 05	72	115	62.6
Total	138	365	37.8

Figure 6.12: Number of Students Using Matterhorn vs Total Number of Students

Note that these times are not restricted to the clipshow functionality; they include normal viewing times as well. We suspect that the very different participation levels are an artifact of both the pedagogical approaches used in the class, as well as their lengths. Section 01 was held three times a week for 50 minutes, and was taught by faculty member who speaks continuously leaving little in the way of breaks. Section 05, on the other hand, was held twice a week for 80 minutes, and was taught by a faculty member who taught for roughly 10 to 15 minutes, and then had an in-class work period for roughly the same length of time. This difference alone might explain the difference in participation levels: Some students in section 05 no doubt felt that work periods were a poor use of their time and began skipping class, and consequently had to watch the videos instead. There are an unknown number of confounding factors however, so we will not try to draw any conclusions regarding the actual class attendance rates. What is interesting however, is the data when looking at the distribution of student class vs viewed class, as shown in Figure 6.13.

	Viewed Class Section				Total	
	01		05			
Viewer Section	Min	Users	Min	Users	Min	Users
CMPT 111 01	3194	47	1760	19	4954	66
CMPT 111 03	414	16	210	8	624	24
CMPT 111 05	611	46	5118	62	5729	108
Total	4219	109	7088	89		

Figure 6.13: Viewer Section vs Section Viewed

A number of interesting points present themselves from this data. Intuitively we assume that the students will tend to watch the video from their own section as a primary resource: That faculty member is creating the exams, and other sections may not cover exactly the same material. In our case the faculty were in close collaboration in terms of the material covered in class, and they shared exams across all sections. This means that students could, in theory, watch the videos from another section and still cover all of the necessary material. We expected a certain level of inter-section viewing, but we were very surprised at the amount of time students spent doing this. Section 01 students tended to stick within their own section, with a few fairly high usage viewers also watching section 05's videos. Section 03 watched very little of either set of videos, but of the time they spent watching they preferred section 01 almost 2 to 1. This is interesting in light of section 05 and their strong preference for their own videos. Section 01 watched more than twice as many minutes of section 05 video, than section 05 watched of section 01 video. Interestingly, there were almost as many section 05 students watching section 01 videos as there were section 01 students, but those section 05 students did not watch very much video. This is especially telling when we note that section 01's professor spoke continuously (and thus it was hard to seek within the video to skip a section), whereas section 05's professor had long breaks within his videos. This implies that while section 01's videos were still viewed, section 05's videos were presented in a way which, despite being shorter in terms of actual lecture time, was more useful to students. Unfortunately, there were no clipshows created by students in any of these sections, and viewership of the clipshows created by the principal investigator was limited at best (roughly 5 minutes). We therefore cannot come to any conclusions regarding the efficacy of the clipshows for one class versus another, but we can attempt to approximate that using both the pedagogical methods of the faculty involved and the viewing time. If the same pattern were to hold true, section 05's 'chunky' presentation style would make for excellent, and easy to understand clipshows, and because they would be easy to create there would be many of them. On the other hand, section 01's continuous presentation makes creating clipshows more difficult, and those clipshows which were created had sound artifacts which were quite distracting when switching between clips. In this case there would likely be a number of clipshows (just as there was a significant amount of time spent viewing the videos), but there would not be as many clipshows when compared to section 05. Interestingly, there were a number of students who only watched the *opposite* section's videos. Most of these students were in section 05; however, there did not appear to be a pattern in terms of marks to predict which students would fall into this group.

6.5 Conclusions

In summary, we began this experiment with high hopes of being able to generate good data to support our hypotheses; however, due to a number of issues we were unable to show any strong results. We were able to show interesting usage traces which may correlate with users who would be interested in viewing clipshows but due to a lack of data we cannot truly support any of our hypotheses. Our survey data was much more interesting, and we were able to show several distinct groups of users (and non-users) each with their own sets of complaints and usage patterns. We also outlined several preliminary pieces of future work, more of which will be explored in the next chapter.

Chapter 7

Future Work

7.1 Overview

While our project came to some conclusions, there are many avenues of further exploration. We have grouped these changes into three major themes. The most obvious is the group of new features. Many of the initial implementations have missing features, or have obvious feature improvements available. The second theme centers around appropriate pedagogy and its effects on the usage of the clipshow toolset. The third and final theme is technical in nature. Our code, while functional, is by no means perfect and many current features require cleanup and improvement.

7.2 New Features

While the initial implementation of the clipshow toolset is complete, there will always be more to do. The first, and most obvious piece of future work is to further establish that these enhancements are indeed useful to students or faculty. Especially important to future studies are control groups. In our case we sought mainly to prove the concept rather than empirically show that this a strong addition to Matterhorn, and lecture capture as a whole. Once further work has shown positive results, there are many different avenues of research. We could, for example, attempt to automatically create clipshows from the viewing patterns on the source material. Sections of the video with high viewing numbers would be included in the automated clipshows, as well as those tagged with a specific, preset, vocabulary. This allows learners to both tag things they feel are important but not viewed many times, as well as 'vote with their thumbs.' For instance, each section of video would have a vote count that would increment by one vote every time someone watches that section, one vote for tagging that section with any tag, and a further two votes for adding that section to a manual clipshow. Adding this voting mechanism allows the system to more easily determine important sections because sections which have been tagged, or more importantly added to a clipshow can be assumed to be naturally more important than those sections which have merely been watched. Of course, there are problems with this approach. If learners create clipshows of funny content, or other presentation issues (every time the presenter says "um" for example) then those sections would have extra votes that ideally they would not have. Doubly so should these humour clipshows become popular. Tag analysis might help here; however, if no one tags these sections, or they are not tagged as something the system can recognize (funny, “um”, etc.) then there is no easy way to automatically exclude these votes. The system could define a set of tags with which learners could mark sections of the clipshow that do not belong. These sections could then be removed, and that removal could in turn affect the initial automated clipshow creation by adding semantic analysis of the removed sections to the automated generation logic.

These concerns naturally lead into the domain of privacy concerns in general. The enhancements proposed above would be improved further by adding controls around who can see which tags, annotations, bookmarks, and clipshows. Is a private clipshow truly private, or should the lecturer have access to it as well? The answer is complicated and may indeed vary depending on where the tool is employed, especially when we consider the differing privacy laws around the world. Therefore, part of the future work could be to add these privacy and access controls and then make them available to the system administrators. A second and perhaps more serious concern surrounds intellectual property issues. Who owns the clipshow? Who owns the original source material? Considering the differences in various countries’ intellectual property laws (ignoring for the moment any institutional rules) the answer would, again, vary and so another part of the future work would include adding support for advanced clipping options. These options would include blacklisting sections (e.g. a video that is shown in class) so that they could not be clipped at all or could only be shared internally (this would disable the default Matterhorn embedded player functionality).

Another planned feature that may encourage student usage of the system is an alternate playback mode. Right now the scrubber bar jumps around the timeline as the player reaches individual clips. This can be confusing, especially if the clips are out of order, or repeating. Our original plans called for an alternate playback mode which presents the clipshow as the only part of the video available, as if the user is watching an entirely separate video without a clipshow. This feature may not seem important, but if you combine it with adding clipshows to the main Matterhorn Search index you give the clipshows a powerful legitimacy: they are effectively first class objects at this point, rather than second class objects dependent upon their host video. This also leads into the idea of meta-clipshows, and recursively possibly to further meta-levels. Once the clipshow becomes a first class object, further clipshows could be built on top of it. There is potentially no limit to the depth of these recursions, barring technical limitations of the player being unable to play ever shorter videos. What effect this might have on learners is unknown, but this bears investigation regardless.

7.3 Pedagogy

Future work should also explore the effects of less technical changes. As we outlined earlier, teaching style has an immense effect on usage of both clipshow and non-clipshow viewing. Future studies should take this into account when choosing the classes involved in the study. Choosing groups of faculty willing to share their lectures among multiple sections of the same class can also lead to potentially interesting results. For example, in our study we had two faculty members with very different presentation styles sharing video between their two sections. This had a major effect both on the viewing time, as well as the viewing choices (which section to view) and should either be a focus, or controlled for future studies.

Our original goal with this study was to attempt to determine why learners are watching clipshows. While we did not succeed in this due to a lack of data, we can make suggestions to aid further studies. Some of the students who turned in surveys were puzzled at the lack of clipshow content, to the point that some of them directly asked why the professor in their class did not create any clipshows. While this is unfortunate because it implies a fault with our study, it does point to an interesting avenue for future studies: faculty, or staff assisted clipshows. We suggest that future studies employ the faculty, or (more likely) their teaching assistants to create clipshows immediately after the class recording becomes available. As Bloom showed in [8], students who are taught in a one-to-one learning environment attain, on average, much higher grades, and this could be one way to personalize some of the interaction between teaching assistant and student. For example, this could be a partial replacement for some tutorial sections, with the balance of the time being devoted to a question and answer format. The advantage here is that rather than having to wait until their next tutorial to review the content, students are presented with (in effect) a review class immediately. The production values in these clipshows would potentially also be much higher since the teaching assistant could be assumed to be at least somewhat familiar with the material already. Instead of just creating a clipshow, the assistant could annotate particularly difficult, or weak sections of the original lecture with detailed explanations. While this sounds like a heavy burden for already heavily worked teaching assistants, it might only be temporary. We suspect that, had we had clipshows being created from the beginning of our study, we would have seen eager or high-performing learners begin making clipshows. We hope that once these students have begun creating clipshows they will not stop as the teaching assistant provided ones slowly stop being created. The reverse may also work once the cold start problem has been overcome. A master teacher seeks to adapt their teaching to their students[22], and could make use of the student-generated clipshows to determine which sections of the class material their students are having trouble with. These sections could then receive extra attention in class and online from the teaching staff, be that teaching assistants or faculty.

7.4 Technical Improvements

There are also more mundane, but no less important pieces of future work to consider. The current iteration of the clipshow tools lacks an edit mode, something which several ECON students worked around by creating clipshows private to themselves to practice their clipping. Future iterations should include this option. Likewise, the tags associated with clipshows were only usable when the user directly searched for them. This made finding clipshows difficult if you did not know what it was already tagged with. Future iterations should include some kind of tag cloud, either in the player itself or better yet in the Matterhorn search index. This would enable students browsing the list of recordings for a given class to immediately see a user-generated tag cloud of the contents of each lecture. A common complaint was the clipping accuracy: Users were expecting professional grade editing tools. This complaint was the result of two issues within the study: the streaming server itself, and the clipshow editing tools. We chose Red5[58] as a streaming server, which seems to have a hard limit on how accurate its seeking could be. This obviously installed a hard floor on how accurate any seeking operation could be within the video. The larger part of the seeking problems was the actual clipshow creation tool itself. Students were expecting clipping resolution of 1 second in many cases, and the tools were not setup for this level of accuracy. Clips were defined by their start and end points (see Figure 4.5), and those points were set by dragging either end of the clip user interface element to the appropriate place. Unfortunately, due to limited screen real estate, the number of screen pixels available for positioning these clips was almost always less than the number of seconds within the video. This meant that even if the user stretched the clipshow saving dialog to the full width of their screen they would not have enough room to map one pixel to one second. A possible solution to this is a local magnification tool similar to those found in the accessibility tools bundled with most major operating systems. However, as we can attest from creating some clipshows, resolutions of 1 second may not be accurate enough to completely remove the blips from the beginnings of each clip. A second solution might be allowing users to insert transition notices between segments, however this would only be useful in some cases because it breaks the flow of the presentation. Both of these issues fall under the general umbrella of usability, which is an area that future studies must take into account. Our study was intended to be a proof of concept, rather than something that could conclusively prove anything. Correspondingly, our usability testing was limited, at best. In our case we made use of the graduate students within the ARIES lab, which is obviously not the ideal group of test users when designing a user interface for general use. Future studies should take a wider and more extensive approach to usability testing. Finally, the language used internally to describe a clipshow was created in an ad-hoc fashion, with little thought given to future expandability. Future revisions of the clipshow tools may be capable of including material from multiple sources, overlaying text, and fancy transitions. We suggest that one of the first things that should be replaced within the current clipshow codebase is this description language, in favour of something suited to the task such as SMIL[72]. This language is a purpose built language based off of XML[71] which is designed to do exactly what would be required in a more advanced clipshow player.

7.5 The Next Study

As we have outlined earlier in this thesis, there are many avenues to explore for the next study. Assuming most, if not all of the technical limitations and bugs were fixed, a future study would have to take into account a number of lessons learned during this experiment. First and foremost, a tutorial must be made available to the learners if not immediately, then within the first two weeks. We suspect that many of the students enrolled in the classes participating in the study could have been convinced to participate had we had a stable system and a tutorial available immediately. Likewise, as was shown with the Economics sections, a faculty member involved in ensuring that students at least experiment with the system is immensely helpful for motivating further student use. If the faculty member(s) are unwilling to motivate students in this manner, then a less drastic and but potentially less effective method could be used. In this case, the class teaching assistants could be enlisted to provide clipshows to circumvent the cold start problem, as discussed earlier. Another, more optional, addition to future studies would be in-person tutorials to show students how to make use of the clipshow tools. In terms of features required for future studies, we would highly encourage the implementation of a draft mode, and an clipshow editing tool. The draft mode, or at the very least an explicit announcement that the privacy tools should be used in-lieu, was a highly sought after addition to our current study, and should not be difficult to implement. The clipshow editing tool is a less obvious addition. In our study, once a clipshow was saved there was no way to edit it, or delete it. This led to the creation of a number of duplicate clipshows while users experimented with the system. Future studies should allow users to edit their clipshows (add metadata, make it private post-hoc) or delete them. In the future, we recommend that the faculty member involved be familiar with lecture capture as a whole, and Matterhorn as a system prior to the term of the study. This provides expectation management in terms of what lecture capture can, and cannot do. For example, the answer to *Can my lecture be professionally edited before being released to students?* or *How do I capture the output from my iPad?* may vary depending on available IT resources, while faculty new to lecture capture may expect this to be available. Finally, the pedagogical style of the faculty member(s) taking part should be evaluated for suitability for use with clipshows. As we outlined earlier, some pedagogical styles work better than others when clipped, and this may have a major effect on student uptake of the tool.

There are a number of other avenues that could be studied as well, either in conjunction with the above study, or as a standalone project. For instance, in [19] and [36], the authors studied using user models to match up users for peer help sessions. Similar usage pattern data is generated by Matterhorn, and given that clipshows can be used to identify mutually interesting sections of the video there is little reason why this idea could not be implemented here. Imagine a system where users could insert their own, new material into the video stream, intermixed with the existing lecture material. If a peer help session was requested, the user (likely a highly-interested one) could create a clipshow containing both class material and their own explanation. Building upon this, interaction graphs such as those found in [9] (chapter 5) and [12] could be generated to highlight highly effective learners, who could then be promoted in some way, whether they were involved with peer help sessions or not. For instance, these highly effective learners could be given first choice when choosing teaching assistants for the following academic term. They have already highlighted themselves as effective teachers, it will likely be fruitful for them to continue in a paid capacity. User models could also be used to improve search results, and independently suggest clipshows to students as well. For instance, as in [9], users could be clustered using some type of datamining tool (such as the WEKA toolkit[35]). Users in similar clusters could have their search results modified, pushing their suggestions towards content most appropriate for their usage profile. Indeed, even the user's reaction to their search results could be fed back into the clustering system. User who searched for something similar to one another, and then reacted the same way would be pushed towards a shared cluster for search results, akin to the recommendation systems employed by Amazon and Netflix. This amounts to collaborative filtering, such as described in [73]. This also plays into a potential weakness of our study: the definition of a useful clipshow. Imagine instead if we had a looser definition, one that included a reputation system for students. We could have an emergent definition of a useful clipshow based not only on the content of the clipshow, but also the reputation of the user. This might also have the effect of boosting usage. As users see their reputation rise, they might decide that creating more clipshows is a worthwhile endeavour. A final idea for user profile mining is that of the viewing profile itself. How a user views a video likely varies drastically, from users who sit and watch hours of video continuously, to those who watch in short 10 minute chunks. This difference could be exploited to further narrow the suggestions not only based on the content, but how other users watched said content.

Clipshows could also be generated automatically by the system to target certain usage profiles, an example of adaptive presentation adaptation from [27]. We could even extend the modelling both ways (c.f [46]), with snapshots of learner models being associated with clipshows and then fed back into further recommendations. This is especially powerful with the user tracking software built into Matterhorn. As discussed in [46], 'information about how the users interacted with the content, including observed metrics such as dwell time, number of user keystrokes, patterns of access, etc.' might be attached to a given learning object, in this case a clipshow. As the semester goes on, this clipshow would become a large repository of snapshotted user data, potentially providing a way to show a learner how they are progressing using multiple snapshots from the same clipshow, or a way of reliably recommending clipshow content to users with similar user profiles and progressions. Finally, learning outcomes and comprehension could be studied using the Matterhorn user tracking data, and clipshows. For example, in [55], Peckham examines reading comprehension tasks using software to determine which follows the user's reading behaviour and its pedagogical effectiveness for given tasks. This is similar to a clipshow in that the sections of the text (video) which are skimmed over (omitted from the clipshow) is material which is understood or unimportant, whereas sections which are reviewed slowly (included in the clipshow) is material which is difficult or not understood.

As a final thought, imagine applying a study like this against a class much larger than the current average class size. One of our major issues was the cold start problem. While many of the users would likely not be any more engaged with the tool than in our study, there would still be a fraction that would take part of their own volition. This may even be enough to overcome the cold start on its own. Once the startup pains have passed we imagine that many of the users who expressed dismay at the lack of content would become active. This effect could easily snowball: even if only 1 in 10 users would be interested in clipshows if they were being created, we could still have 100 users in a class of 1000. This would be more than enough to generate truly interesting results!

7.6 Conclusion

This thesis, while not providing conclusive results, does offer a glimpse into the possibilities that tools like clipshows provide for helpful learning. Even with suboptimal studies and an imperfect tool we found interesting correlations in our data. With the promising future of lecture capture (the major prerequisite for this type of study), we feel that the concept of clipshows will be an important addition to future systems. Our study was a proof of concept for the toolset, and forms the groundwork for more advanced studies in the future.

References

[1] Khan Academy. The khan academy. https://www.khanacademy.org, 2006. Accessed 2013-07.

[2] Ryan S J D Baker. Modeling and understanding students' off-task behavior in intelligent tutoring systems. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 1059–1068. ACM, 2007.

[3] Ryan S J D Baker and Kalina Yacef. The State of Educational Data Mining in 2009 : A Review and Future Visions. Journal of Educational Data Mining, 1(1):3–17, 2009.

[4] Ryan Shaun Baker, Albert T Corbett, and Kenneth R Koedinger. Detecting student misuse of intelligent tutoring systems. In Intelligent tutoring systems, pages 531–540, Heidelberg, 2004. Springer Berlin.

[5] Jack Barokas, Markus Ketterl, and Christopher Brooks. Lecture Capture: Student Perceptions, Expectations, and Behaviors. In World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (E-learn 2010), pages 424–431, Chesapeake, VA, 2010. AACE.

[6] S Bateman, R Farzan, P Brusilovsky, and G McCalla. OATS: The Open Annotation and Tagging System. In the Proceedings of the International Scientific Conference of the Learning Object Repository Research Network (LORNET 2006), Montreal, Canada, November 2006.

[7] Scott Bateman, Christopher Brooks, Gordon McCalla, and Peter Brusilovsky. Applying Collaborative Tagging to E-Learning. Proc. of ACM WWW, 3(4):1–7, 2007.

[8] Benjamin S Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher, 13(6):4–16, 1984.

[9] Christopher Brooks. A Data-Assisted Approach to Supporting Instrucitonal Interventions in Technology Enhanced Learning Environments. PhD thesis, Department of Computer Science, University of Saskatchewan, 2012.

[10] Christopher Brooks, Kristofor Amundson, and Jim Greer. Detecting Significant Events in Lecture Video using Supervised Machine Learning. In Vania Dimitrova, Riichiro Mizoguchi, Benedict du Boulay, and Art Graesser, editors, 2009 conference on Artificial Intelligence in Education (AIED09), pages 483–490, Brighton, UK, 2009. IOS Press Amsterdam, The Netherlands, The Netherlands.

[11] Christopher Brooks, Carrie Demmans Epp, Greg Logan, and Jim Greer. The Who, What, When, and Why of Lecture Capture. In 1st International Conference on Learning Analytics and Knowledge 2011 (LAK'11), pages 86–92, New York, New York, USA, 2011. ACM Press.

[12] Christopher Brooks, Jim Greer, and Carl Gutwin. Handbook of Learning Analytics: Methods, Tools and Approaches, chapter The Data-Assisted Approach to Building Technology Enhanced Learning Environments. 2013. In Submission.

[13] Christopher Brooks, G.Scott Johnston, Craig Thompson, and Jim Greer. Detecting and categorizing indices in lecture video using supervised machine learning. In OsmarR. Zaane and Sandra Zilles, editors, Advances in Artificial Intelligence, volume 7884 of Lecture Notes in Computer Science, pages 241–247. Springer Berlin Heidelberg, 2013.

[14] Christopher Brooks and Lori Kettel. Building a Learning Object Content Management System. In Proceedings of World Conference on E-Learning in Corporate, Healthcare, & Higher Education (E-Learn 2005), pages 24–32, Montréal, Québec, 2005.

[15] Christopher Brooks, Adam McKenzie, Denis Meyer, Markus Moormann, Matjaz Rihtar, Ruediger Rolf, Nejc Skofic, Micah Sutton, Ruben Perez Vazquez, Benjamin Wulff, Markus Ketterl, Adam Hochman, Josh Holtzman, Judy Stern, Tobias Wunden, Kristofor Amundson, Greg Logan, and Kenneth Lui. Opencast Matterhorn 1.1: Reaching New Heights. In Proceedings of the 19th ACM international conference on Multimedia - MM '11, pages 703–706, New York, New York, USA, 2011. ACM Press.

[16] Christopher Brooks, Craig Thompson, and Jim Greer. Visualizing Lecture Capture Usage: A Learning Analytics Case Study. In LAK 2013 workshop on Analytics on Video-Based Learning, pages 9–14, Leuven, Belgium, 2013.

[17] Christopher Brooks, Mike Winter, Jim Greer, and Gordon McCalla. The Massive User Modelling System (MUMS). In Intelligent Tutoring Systems 2004 (ITS04), pages 73–112, Maceiò, Alagoas, Brazil, 2004. Springer.

[18] Peter Brusilovsky and Ricardo Rizzo. Map-based horizontal navigation in educational Hypertext. Hypertext 2002 Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia College Park MD United States 1115 June 2002, 3(1):1–10, 2002.

[19] Susan Bull, Jim Greer, Gordon McCalla, Lori Kettel, and Jeff Bowes. User Modelling in I-Help: What, Why, When and How. In User Modeling 2001, pages 117–126. Springer, 2001.

[20] Susan Bull and Judy Kay. Student Models that Invite the Learner In: The SMILI:() Open Learner Modelling Framework. International Journal of Artificial Intelligence in Education, 17(2):89–120, 2007.

[21] Susan Bull and Theson Nghiem. Helping learners to understand themselves with a learner model open to students, peers and instructors. In Proceedings of Workshop on Individual and Group Modelling Methods that Help Learners Understand Themselves, International Conference on Intelligent Tutoring Systems, volume 2002, pages 5–13. Citeseer, 2002.

[22] William Buskist. Ways of the Master Teacher. Observer, 2004. Accessed 2013-06.

[23] Nicola Capuano, Anna Pierri, Francesco Colace, Matteo Gaeta, and Giuseppina Rita G.R. Mangione. A mash-up authoring tool for e-learning based on pedagogical templates. In Proceedings of the first ACM international workshop on Multimedia technologies for distance learning, pages 87–94, New York, New York, USA, 2009. ACM.

[24] Carnegie Learning and Theoretical Basis. The Cognitive Tutor: Applying Cognitive Science to Education. Technical report, Carnegie Learning, Inc., Pittsburgh, PA, USA, 1998.

[25] Jacqui Cheng. Adobe extends online video remixer to youtube and mtv. http://arstechnica.com/uncategorized/2007/06/adobe-extends-online-video-remixer-to-youtube-and-mtv. Accessed 2013-07.

[26] Alex Chitu. Youtube video editor. http://googlesystem.blogspot.ca/2010/06/youtube-video-editor.html, 2010. Accessed 2013-07.

[27] Owen Conlan. State of the Art: Adaptive Hypermedia. M-Zones Deliverable, 1:47–57, May 2003.

[28] Douglas Crockford. The application/json media type for javascript object notation (json). http://www.ietf.org/rfc/rfc4627.txt?number=4627, 2006. Accessed 2013-07.

[29] M. da Graça Pimentel, G.D. Abowd, and Y. Ishiguro. Linking by interacting: a paradigm for authoring hypertext. In Proceedings of the eleventh ACM on Hypertext and hypermedia, pages 39–48. ACM, 2000.

[30] Gerben Dekker, Mykola Pechenizkiy, and Jan Vleeshouwers. Predicting students drop out: A case study. EDM, 9:41–50, 2009.

[31] Sidney K DMello, Scotty D Craig, Amy Witherspoon, Bethany Mcdaniel, and Arthur Graesser. Automatic detection of learners affect from conversational cues. User Modeling and User-Adapted Interaction, 18(1-2):45–80, 2008.

[32] Rosta Farzan and Peter Brusilovsky. Social Navigation Support Through Annotation-Based Group Modeling. In Liliana Ardissono, Paul Brna, and Antonija Mitrovic, editors, User Modeling, volume 3538 of Lecture Notes in Computer Science, pages 463–472. User Modeling Inc., Springer, 2005.

[33] Alex Faulkner, Alex Lewis, Miles Meyer, and Neil Merchant. Creating an HTML5 web based video editor for the general user. Computer Graphics Technology (CGT411) Student Projects, Purdue University, Spring, 2010.

[34] IETF Network Working Group. Hypertext transfer protocol – http/1.1. http://tools.ietf.org/html/rfc2616, 1999. Accessed 2013-07.

[35] Geoffrey Holmes, Andrew Donkin, and Ian H Witten. WEKA: a machine learning workbench. Proceedings of ANZIIS 94 Australian New Zealnd Intelligent Information Systems Conference, 24(3):357–361, 1994.

[36] H Ulrich Hoppe. The use of multiple student modeling to parameterize group learning. In 7th World Conference on Artificial Intelligence in Education (AIED95), 1995.

[37] Blackboard Inc. Blackboard — technology and solutions built for education. http://www.blackboard.com, 2003. Accessed 2012-06.

[38] Google Inc. Youtube video editor. https://support.google.com/youtube/answer/183851, 2013. Accessed 2013-07.

[39] Oracle Inc. Jsr 317: Java persistence 2.0. https://glassfish.java.net/javaee5/persistence, 2007. Accessed 2013-06.

[40] Oracle Inc. Jsr 339: Jax-rs 2.0: The java api for restful web services. http://jax-rs-spec.java.net, 2011. Accessed 2013-07.

[41] JSON-Simple. Json-simple. \url{http://code.google.com/p/json-simple/}, 2012. Accessed 2012-09.

[42] Markus Ketterl, Johannes Emden, and Jörg Brunstein. Social selected learning content out of web lectures. In Proceedings of the nineteenth ACM conference on Hypertext and hypermedia, HT '08, pages 231–232, New York, NY, USA, 2008. ACM.

[43] Markus Ketterl, Robert Mertens, and Oliver Vornberger. Bringing Web 2.0 to web lectures. Interactive Technology and Smart Education, 6(2):82–96, 2009.

[44] Markus Ketterl, Olaf A Schulte, and Adam Hochman. Opencast Matterhorn: A community-driven open source software project for producing, managing, and distributing academic video. Interactive Technology and Smart Education, 7(3):168–180, 2010.

[45] Alexander McAuley, Bonnie Stewart, George Siemens, and Dave Cormier. The MOOC model for digital practice (University of Prince Edward Island, Social Sciences and Humanities Research Councils Knowledge synthesis grants on the Digital Economy report), 2010.

[46] Gordon McCalla. The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners. Journal of Interactive Media in Education, 2004(May):1–23, 2004.

[47] Scott W Mcquiggan, Bradford W Mott, and James C Lester. Modeling self-efficacy in intelligent tutoring systems: An inductive approach. User Modeling and User-Adapted Interaction, 18(1-2):81–123, 2008.

[48] Robert Mertens, Markus Ketterl, and Peter Brusilovsky. Social navigation in web lectures: a study of virtPresenter. Interactive Technology and Smart Education, 7(3):181–196, 2010.

[49] Moodle.org. Moodle.org: open-source community-based tools for learning. https://moodle.org, 2003. Accessed 2013-07.

[50] Jehad Najjar, Martin Wolpers, and Erik Duval. Attention metadata: Collection and management. In Workshop on Logging Traces of Web Activity: The Mechanics of Data Collection, pages 1–4, 2006.

[51] University of Phoenix. Online schools, classes, degree programs - university of phoenix. \url{http://www.phoenix.edu}, 1976. Accessed 2013-07.

[52] On2. On2 vp6. http://wiki.multimedia.cx/index.php?title=On2_VP6, 2003. Accessed 2013-08.

[53] Judith A Ouimet and Robert A Smallwood. Assessment Measures: CLASSE–The Class-Level Survey of Student Engagement. Assessment Update, 17(6):13–15, 2005.

[54] J. Padhye and J. Kurose. Continuous-media courseware server: a study of client interactions. IEEE Internet Computing, 3(2):65–73, 1999.

[55] T Peckham and G McCalla. Mining Student Behavior Patterns in Reading Comprehension Tasks. In Proceedings of the 5th International Conference on Educational Data Mining, pages 87–94, 2012.

[56] Stephen R. Porter. Raising response rates: What works? New Directions for Institutional Research, 2004(121):5–21, 2004.

[57] Opencast Project. Opencast matterhorn. http://opencast.org/matterhorn, 2010. Accessed 2013-07.

[58] Red5. Red5 media server. \url{http://www.red5.org/}, 2012. Accessed 2013-06.

[59] TelTek Video Research. Galicaster. http://wiki.teltek.es/display/Galicaster/Galicaster+project+Home, 2012. Accessed 2013-05.

[60] John Resig. jquery. http://jquery.com, 2006. Accessed 2013-06.

[61] Cristóbal Romero, Sebastián Ventura, Pedro G Espejo, and César Hervás. Data mining algorithms to classify students. In EDM, pages 8–17, 2008.

[62] Olaf A. Schulte, Tobias Wunden, and Armin Brunner. REPLAY - An integrated lecture recording infrastructure to interactively and collectively generate learning objects. ACM Press, New York, New York, USA, 2008.

[63] George Siemmens. What are learning analytics?, 2010. Accessed 2013-07.

[64] Adobe Software. Adobe flash video file format specification version 10.1. http://download.macromedia.com/f4v/video_file_format_spec_v10_1.pdf, 2010. Accessed 2013-07.

[65] JF Superby, JP Vandamme, and N Meskens. Determination of factors influencing the achievement of the first-year university students using data mining methods. In Workshop on Educational Data Mining, pages 37–44, 2006.

[66] Pramuditha Suraweera and Antonija Mitrovic. KERMIT: a constraint-based tutor for database modeling. In Intelligent Tutoring Systems: 6th International Conference, ITS 2002, Biarritz, France and San Sebastian, Spain, June 2-7, 2002. Proceedings, pages 201–216. Springer, 2002.

[67] Epiphan Systems. Matterhorn appliance. http://www.epiphan.com/products/other-applications/matterhorn, 2012. Accessed 2013-07.

[68] Design Technology, Entertainment. Ted talks. \url{http://www.ted.com/talks}, 1984. Accessed 2013-07.

[69] Athabasca University. Athabasca university : Canada's leader in online & distance education. http://www.athabascau.ca, 1970. Accessed 2013-07.

[70] University of Purdue. Course Signals - Stoplights for Student Success. http://www.itap.purdue.edu/learning/tools/signals, 2012. Accessed 2013-07.

[71] W3C. Extensible Markup Language (XML). http://www.w3.org/XML, 1996. Accessed 2013-07.

[72] W3C. Synchronized Multimedia Integration Language (SMIL 3.0). http://www.w3.org/TR/smil, 2008. Accessed 2013-07.

[73] O R Zaiane. Recommender systems for e-learning: towards non-intrusive web mining. In C Romero and S Bentura, editors, Data Mining in E-Learning, chapter 5, pages 79–96. WIT Press, 2008.

[74] ETH Zurich. ETH Zurich Multimedia Portal: video, audio, podcast, webcast, live-streaming. http://www.multimedia.ethz.ch/tec/playmobil, 2009. Accessed 2013-06.

Appendix A

Final Survey Consent Agreement

Department of *Computer Science* ***Participant Consent Form***

Project Title: **Data mining implicit and explicit user-generated video modifications for measuring toolset effectiveness**

Researcher(s):
Greg Logan, Department of Computer Science (966-2676), gdl420@mail.usask.ca

Supervisor(s):
Dr. Jim Greer, Professor, Department of Computer Science (966-2234), greer@cs.usask.ca
Dr. Gord McCalla, Professor, Department of Computer Science (966-4092), mccalla@cs.usask.ca

Purpose(s) and Objective(s) of the Research:

- The goal of this study is to determine the effectiveness of the new 'clipshow' tools in a real-world classroom environment.

Procedures:

- Students and instructors will be allowed to use the system however they see fit
- Students will be encouraged to fill out two or more surveys over the course of the term
- Normal Matterhorn usage data will be recorded for cross reference and verification purposes

Please feel free to ask any questions regarding the procedures and goals of the study or your role.

Confidentiality: Matterhorn usage data is associated with a securely obfuscated version of the student's NSID. Surveys will prompt for NSIDs, however these will be obfuscated in the same manner as the Matterhorn usage data before the results are given to the researcher.

- **Storage of Data:** Electronic copies will be stored on secure ARIES laboratory computer systems.
  - Any paper copies will be stored securely in Dr. Jim Greer's office. After 5 years the data, both paper and electronic, will be destroyed.

Right to Withdraw:

- Your participation is voluntary and you can answer only those questions that you are comfortable with. You may withdraw from the research project for any reason, at any time without explanation or penalty of any sort. Whether you choose to participate or not will have no effect on your position [e.g. employment, class standing, access to services] or how you will be treated. Should you wish to withdraw, please contact Dr. Jim Greer (listed above). Your data will be removed from the system. Your access to the system for the rest of the term will not be affected. Your right to withdraw data from the study will apply until the end of term. After this date, it is possible that some form of research dissemination will have already occurred and it may not be possible to withdraw your data.

Follow up:

- To obtain results from the study, please contact the researchers.

Questions or Concerns:

- Contact the researcher(s) using the information at the top of page 1;
- This research project has been approved on ethical grounds by the University of Saskatchewan Research Ethics Board. Any questions regarding your rights as a participant may be addressed to that committee through the Research Ethics Office ethics.office@usask.ca (306) 966-2975. Out of town participants may call toll free (888) 966-2975.

Consent:

- Consent in this study constitutes consent that, if applicable, students may be contacted for further studies related to this research.
- Once contacted, participants would be prompted to consent for any further research.

By completing and submitting the questionnaire, **YOUR FREE AND INFORMED CONSENT IS IMPLIED** and indicates that you understand the above conditions of participation in this study.

Page 1 of 1

Appendix B

Final Survey Instrument

University of Saskatchewan

Clipshow Survey

The purpose of this survey is to identify if, how, and why you used the clipshow tools built into the Opencast Matterhorn system that your class used for lecture capture this term. These tools are experimental in nature, so it is important that we determine which of our additions are useful, and which parts need further work.

Part I: Clipshows

1. Your NSID:_____________

2. Did you view any clipshows:
   Yes \ No
3. Did you find them useful:
   Not at all \ A little \ Somewhat \ Very
4. Please explain briefly why or why not:

5. How did you find the clipshows you watched?
   User Tab \ Dropdown \ Other
6. How many clipshows did you share with others (study group, email, etc)?
   None \ A few \ Some \ Lots
7. How often did you use the clipshow series tab/tool?
   I did not use it \ A little \ Some \ Lots
8. Would the clipshow series tool would be more useful if more clipshows were available?
   No \ A little \ Some \ Lots
9. Did you vote for a clipshow (Funny/Useful/Dislike)?
   Yes \ No
10. Would the voting be more useful if more clipshows were available?
    No \ A little \ Some \ Lots
11. What could be done to make you more likely to view a clipshow in the future?

Part II: Annotations

12. Did you use the commenting or annotation feature?
    Yes \ No
13. Did you find the ability to comment, or the comments left by others useful?
    No \ A little \ Some \ Lots
14. Please explain briefly why or why not:

Part III: Clipshow Creation

15. Did you create any clipshows
    Yes \ No
16. Why did you create those clipshows (circle all that apply)?
    Assignment \ Personal Interest \ Experimenting with the tool \ Other
17. How easy was the clipshow creation tool to use?
    Impossible \ Very difficult \ Difficult \ Somewhat easy \ Easy
18. Did you use the privacy features when creating clipshows or annotations?
    Never/Rarely \ Sometimes \ Often \ Very Often
19. If you chose never or rarely, why not?
    Too difficult to use \ Not enough time \ No interest \ Other
20. How many clipshow series did you create?
    0 \ 1-2 \ 3-5 \ 6+
21. Do you see the clipshow series tool as a useful feature?
    Yes \ No
22. How many private clipshows did you create (ones only you or a small group could see)?
    0 \ 1-2 \ 3-5 \ 6+
23. What were the purposes of these clipshows?
24. Did you create any clipshows with repeated clips?
    Yes \ No
25. Did you create any clipshows with out of order clips?
    Yes \ No
26. Do you think that either of these two features could be useful?
    Not useful \ A little \ Somewhat \ Very useful
27. How could the clipshow creation tool be improved?
28. Can you tell us what you liked about the clipshows?
29. Can you tell us what you disliked about the clipshows?

Thank you for taking the time to complete this survey

Appendix C

Partial List of Tracked Events

- Starting the application
- A heartbeat every 30 seconds, regardless of user action once the application has started
- Starting and pausing playback
- Fast forward and rewind
- Skipping ahead and in reverse (larger chunks than fast forward or rewind)
- Seeking within the video by clicking and/or dragging the caret on the timeline
- Searching the OCR text data
- Creating a clipshow (placing the clips within the source video, rearranging them, etc)
- Viewing a clipshow
- Creating a clipshow series (selecting the clipshows, rearranging them, etc)
- Selecting a clipshow series
- Playing/pausing a clipshow series
- Annotating a video/clipshow
- Replying to annotations
- Searching for clipshows, authors, or tags
- Voting for a clipshow
- Creating annotations