

Spelunker's Trail -- Visualization of Media Object Usage and Navigation
Trajectory of a Web Based Learning Environment

By

Yunzhen Lynn Qu

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degrees of
Bachelor of Science in Electrical [Computer] Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

May 21, 1999

[June 1999]

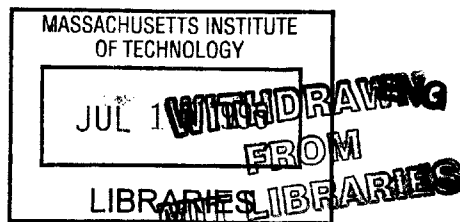
© Copyright 1999 Yunzhen Lynn Qu. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and
distribute publicly paper and electronic copies of this thesis
and to grant others the right to do so

Author _____
Department of Electrical Engineering and Computer Science
May 19, 1999

Certified by _____
Richard C. Larson
Thesis Supervisor

Accepted by _____
Arthur C. Smith
Chairman, Department Committee on Graduate Thesis



Acknowledgements

I would like to thank Professor Richard Larson for giving me this thesis opportunity. Also the other PIVoT staff members who were always encouraging and eager to help: Professor Steve Lerman, Laura Koller, Steve Niemczyk, Dr. Nish Sonwalker, Phil Bailey, Erik Kangas, and last but certainly not least, David Mycue.

I would also like to acknowledge my wonderfully supportive friends: Rob Jagnow, David Manowitz and Richard Li, who helped me with Java; Tom Burbine who lend me his Tufte books; and Rebecca Xiong and Anand Karasi who helped me brainstorm on several occasions.

Lastly I want to thank my parents for their constant nagging (just kidding!) and frequent pep-talks.

Table of Contents

ABSTRACT	4
1. INTRODUCTION	5
1.1 <i>PIVoT</i>	5
1.2 <i>Spelunker's Trail</i>	6
1.3 <i>Motivation</i>	6
2. BACKGROUND.....	9
2.1 <i>Learning with Advanced Technologies</i>	9
2.2 <i>The Flashlight Program and Other Related Work</i>	9
2.3 <i>VISUALIZATION</i>	13
3. DESIGN.....	14
3.1 <i>AGGREGATED USAGE DATA</i>	18
3.1.1 <i>STUDENT PROFILES</i>	19
3.2 <i>INDIVIDUAL NAVIGATION TRAJECTORY</i>	20
3.2.1 <i>SEQUENTIAL VS OTHER DISPLAY MODELS</i>	17
4. IMPLEMENTATION	26
4.1 <i>USER EXPERIENCE</i>	26
4.1.1 <i>VIEWING CLASS USAGE</i>	27
4.1.2 <i>STUDENT PROFILE, REVISITED</i>	28
4.1.3 <i>Viewing Individual Navigation Trajectory</i>	29
4.2 <i>DATA FILES</i>	31
5. DISCUSSION/FUTURE WORK	32
6. CONCLUSION/SUMMARY	28

***Spelunker Trail – Visualization of Media Object Usage and Navigation Trajectory
Of a Web Based Learning Environment***

By
Yunzhen Lynn Qu

Submitted to the
Department of Electrical Engineering and Computer Science

May 21, 1999

In Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Computer[Electrical] Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Spelunker's Trail is a set of design specifications of a software tool. This tool demonstrates a student's usage pattern a multi-media "hypercourse," such as the Physics Interactive Video Tutor (PIVoT). It addresses both aggregated, overall usage for groups, as well as low-level, individual usage by specific students. It displays their activity within the web-based learning environment, using time-series and scatter plots. The tool also links usage data to student profiles in order to reveal possible correlation between the various elements of student behavior. The real identities of the students in the database are hidden, so to prevent artificial incentives for the students to use (or avoid using) the hypercourse as a mean of directly affecting their grade.

Thesis Supervisor: Richard C. Larson
Title: Director, MIT Center for Advanced Educational Services

1. Introduction

The advancement in Information Technology has led to a growing trend towards web-based, multi-media learning. Unlike the traditional classroom-based setting, this type of “Hypermedia” environment allows highly interactive, self-paced, and asynchronous learning, all of which center around the individual student, rather than the professor. This concept has enormous potential for continuous improvement, since it is based on development in web technology. However, as it is also a relatively new concept, there needs to be a lot of exploration and evaluation in order to find the best way to leverage such powerful technology to provide optimal content.

The objective of this project is to design a tool for evaluating web-based learning experience. The deliverables are detailed specifications of the front-end display model and user experience, along with guidelines for back-end integration with the system. The specifications will later be illustrated with a prototype of a software package, whose application is to demonstrate how the Physics Interactive Video Tutor (PIVoT) is used by individual students as well as by the class as a whole.

The rest of this section gives some background on MIT’s efforts in developing multi-media learning environments and describes the PIVoT project in more detail. It also explains the motivation of creating such an evaluation tool. Subsequent sections of this paper recite classic and current research work in related areas, outline the design and implementation of the software, discuss future work to be done, and end with a summary and conclusion of this research and development effort.

1.1 PIVoT

PIVoT is a collaborative project between MIT’s Center for Advanced Education Services (CAES)¹ and the Physics Department. It is a web-based, multi-media learning environment (also called a “Hypercouse”²) for students in 8.01, the introductory course

¹ <http://www-caes.mit.edu>

² Sonwalkar, Nishikant, et al, *Hypermedia in Education: A Case Study*. Cambridge, MA: MIT CAES Hypermedia Teaching Facility, 1998

in Newtonian Mechanics. The most prominent component of PIVoT is a large collection of digitized video clips. In these video clips, Professor Walter Lewin, a renowned Physics lecturer at MIT, addresses the students' most frequently asked questions. Other features include an hypermedia textbook, Java Applet simulations, still and animated graphics, self-tests, and online access to TA's.

Given the ability to access Professor Lewin's instructional video and other related course material at any time they wish, students can learn at their own pace. In addition, the infrastructure of the course content database allows connection between all pieces of relevant information. For example, a page with a self-test question may contain a link to a video clip in which the solution is explained. On the same page there may also be another link which points to a related section in the textbook, which in turn has a link that launches a simulation of the concept. Such a networked system greatly enhances navigation capabilities, enabling each student to explore and focus on areas where he needs the most help.

1.2 *Spelunker's Trail*

The tool we are developing allows one to look at how a multi-media learning environment, such as PIVoT, is used by the intended audience. It gives an overall view of aggregated usage by a group (i.e. a whole class over a semester), as well as provides more in-depth exposure of navigation paths from individual sessions. The prototype will be implemented in the form of an on-line software package, accessible via downloading Java applets off of the World Wide Web.

1.3 Motivation

The main purpose of developing this software tool is to provide insight into students' learning patterns. Hypermedia courseware such as PIVoT is a relatively new concept and requires substantial analysis and feedback. This product allows education specialists and course staff to evaluate the effectiveness of this new form of learning and explore further development potentials.

For example, we can look at whether there is a correlation between students' grades and their usage of PIVoT – do students who use PIVoT more frequently tend to receive better grades? Additionally, we can compare the usage pattern of students who receive good grades to those of ones who have not been performing as well – does there seem to be a commonality within the same group? If so, perhaps there can be a “recommended usage” feature.

Furthermore, since the time spent within PIVoT is categorized by type of experience, or *mode*, we might discover that students seem to prefer some elements of the various learning modes more than others. For example, a large quantity of the video clips may be frequently viewed by many students, whereas, in contrast, the self-quizzes are rarely visited. Then we would want to investigate why some modes seem to be more appealing, and possibly either improve or eliminate the less popular ones. As content providers, we may also wish to focus further development efforts on putting more substance into the most effective modes.

Yet another important reason to have this tool is to see *who* use a hypercourse like PIVoT and *how* they use it. For example, do the liberal-arts majors, who are only taking the class because they are required to, use it? If so, do they actually care about learning the concepts, or are they only looking for answers to help them complete a problem set or study for an exam? Indeed, this would be an excellent general observation to make of all users of PIVoT – do they in fact meander through the site consistently throughout the term to educate themselves about the course material, or do they just use it to address specific questions for problem sets or exam preparations?

On a higher level, such a tool is also motivated by the quest for discovering the appropriate balance between IT-based and other methods for education. The fundamental question is: have we really used technology effectively to help us improve the learning process? The answer to that must be found by studying the means, not just the ends, of learning. We want to involve the staff of the course to focus on the choices the students made to help them learn better. We want to show that our investment in the

technologies has paid off in the way we expected. The Background/Related Work section will discuss this issue in further detail.

2. Background

2.1 Learning with Advanced Technologies

In recent years, conventional classroom interaction between teachers and students has been supplemented by an increasing number of software and hardware tools. The introduction of computer and communication technologies has enabled various applications that enrich and hopefully facilitate the teaching/learning process.

There is an enormous amount of research and development efforts dedicated towards technologies for improving education: multi-media, distance learning platforms, the Internet, quantitative tools, publishing tools, to just name a few. Syllabus magazine, published by the San Jose based Syllabus Press³, is an informative resource which has feature articles, case studies, product reviews, and profiles of technology deployments and how they can support teaching, learning, and administrative activities.

Internationally, colleges, universities, and even grade and secondary schools have long begun to implement pedagogical curricula which explore the application of the latest technologies. MIT's Center for Advanced Education Services, Cornell's Office of Distance Learning⁴, and The Open University in Great Britain⁵ are just a few examples of institutions for higher-learning who are actively involved in these activities. In many cases, a small number of students is selected initially as a "pilot group" to test out these cutting-edge programs. If the outcome is favorable, then often these programs are deployed on a larger scale and opened up to the entire student population.

2.2 The Flashlight Program and Other Related Work

Educational institutions are investing an enormous amount of effort and money in computing, video, and communication technologies. Changes in administrative disciplines will also be required to accommodate these new programs. As such, they would really want to know whether all this is working, and whether the results are worth

³ <http://www.syllabus.com/>

⁴ <http://www.dl.cornell.edu/odl98/about.dl/about.dl.stm>

the expense and the stress. This sub-section takes a look at some of the related past and current research work, including the strengths and weaknesses of innovative educational tools, and what technology can and cannot accomplish.

Various tools are being developed to assess the merit of adopting expensive programs that make use of the latest technologies. One organization which has been active in this arena is the Flashlight Program⁶ from the non-profit TLT group⁷, an affiliate of the American Association for Higher Education. Flashlight's mission, according to director Steve Ehrmann, is "to help institutions develop their own capacity to do studies that can improve teaching and learning with technology... also (to) do studies for institutions when they ask for that (e.g. external evaluations for grants or accreditation)."⁸ Its team of experts tries to do all this in two complementary ways: by developing and distributing toolkits and by working with institutions directly (e.g., through consulting, workshops, or newsletters).

In one of his papers, *Asking the Right Question* (Ehrmann 1995⁹), Ehrmann points out that "if you're headed in the wrong direction, technology won't help you get to the right place." And if we "rush out and buy new technologies without first asking about appropriate education goals, the results are likely to be disappointing and wasteful." Along similar lines, Hal Abelson, Professor of Electrical Engineering and Computer Science at MIT, has said that it would be a big mistake for educators to become so mesmerized by technology that we end up "doing less by doing more."¹⁰

Ehrmann also states that what matters most are "educational strategies for using technology, strategies that can influence the student's total course of study... (it is) not so much what happens in the moment when the student is using the technology, but more how those uses promote larger improvement in the fabric of the student's education"

⁵<http://www.open.ac.uk/>

⁶<http://www.tltgroup.org/flashlight/FLbackground.html>

⁷<http://www.tltgroup.org/>

⁸ Ehrmann, Steve, via email; April 6, 1999

⁹<http://www.learner.org/edtech/rscheval/rightquestion.html>

Thus, he asks the faculty and staff to “know thy students and what they are learning, (because) without asking hard questions about learning, technology remains an unguided missile.”

In another one of his papers (Ehrmann 1997¹¹), Ehrmann states that one of the goals of the Flashlight program is to examine choices about learning made by students in order to “illuminate the relationship between investment in technological infrastructure and improvement (or lack thereof) in outcomes.” In addition, the way we gauge a successful education is slowly but continuously evolving (i.e. shifting emphasis from the grades one receives while in school to how one performs on the job in the future). Therefore it is important to focus on “the practice that tend to produce good learning rather than on the task of directly measuring changes in learning during a period when the learning objectives were changing qualitatively.”

A third Ehrmann paper, *What Outcome Assessment Misses* (Ehrmann 1998¹²), further argues for the necessity of “attending to means, not just ends.” Since learning is most directly the result of what students do, studying what they did, as opposed to what we hope or fear they did, yields useful information. People come into an educational program with a range of needs and capabilities. Ehrmann says that “accidents and coincidences happen” and students are “creative in different ways.” Hence from the same course, different people learn different things as a result of their encounter with a learning opportunity. Without some insight into what people actually did in the program to complement whatever outcome data we can gather, he asks, how can we decide what to do next to improve the outcomes?

Lydia Plowman, a Senior Research Fellow in the School of Cognitive & Computing Sciences at the University of Sussex, wrote an interesting article titled *Narrative*,

¹⁰ Abelson, Hal, speech given at 35th Anniversary for the Laboratory of Computer Science; Cambridge, MA. April 13, 1999

¹¹ <http://www.tltgroup.org/programs/elephant.html>

¹² <http://www.tltgroup.org/programs/outcomes.html>

Interactivity and the Secret World of Multimedia (Plowman¹³). She has been observing young people using CD-Roms in school, and in her article she explores what narrative might mean for multimedia texts in general. This ties her research with ours, since a web-based learning environment, like a CD-Rom, also includes plenty of multimedia text.

One of the key benefits of interactive media is seen as being the lack of imposed structure, giving much greater freedom of control to the user. However, related advantages such as learner control, non-linear format, multiple media, frequent transition between media, all contribute to disruption of the narrative. Plowman states that interactive multimedia “offers the illusion of a more democratic or open medium by making explicit the role of its user in determining meaning and constructing narrative.” However, in reality CD-Roms are fixed media and the user can add nothing and take away nothing – they are “read only.” A hypercourse such as PIVoT differs somewhat in this respect: students do have the opportunity to enter new data into the web site via quizzes, live chats with TA’s, etc. However, the new information does not alter the composition of the web site dynamically (student input will be assessed and changes made gradually by the content providers for improvement). Thus the set of available navigation options remain the same regardless of the student’s behavior.

At the foci of interactivity, learners decide what to do: choose from a multiple of options about where to go next, answer a question, summon help, repeat the section they have just seen, or exit. Whatever the choice, the flow is disrupted by an individual reflection (in making the choice) and also often by a change of medium as well. Plowman claims that although the user “has generally made an explicit decision to choose a particular option, it can be a leap into the unknown.” But she also notes that the loss of linear structure is “welcomed by some theorists, who maintain that the associative structure of interactive media and hypertext more closely model our cognitive processes.”

¹³ <http://www.komm.ruc.dk/interkomm/98-1/plowman.htm>

Professors are able to accommodate diversions, interruptions and repetitions because they can subtly revise the lecture as they go. They can often sense when people are getting lost or need clarification and deal with it as needed. But interactive multimedia cannot do this yet, at least not with the current technology in intelligent software agents. Granted, the unknown can be exciting and arouses the students' curiosity to explore further. But it can also be frustrating and often the students fail to find the optimal path for their learning process. Therefore, she suggests that teachers offer a strong external narrative context which relies on the use of multimedia tool as an information resource. This notion is not exactly along the same lines as our objectives, however, since we view PIVoT as something which promotes independent learning as well as being a repository to supplement classroom education.

2.3 Visualization

For all the reasons discussed above, it is imperative for us to study the kind of decisions students make when using technology to aid learning. Once we have gathered the needed data, however, the challenge then becomes how do we best *present* them? Visual display of quantitative and qualitative information has long been an interesting area of research, especially since its outcome would benefit just about every discipline. Edward Tufte, the founder of Graphic Press and a Professor of Political Science and Statistics at Yale University, is considered one of the foremost authorities in this field. The rest of this sub-section draws extensively from his work, and tries to apply the material towards our design for an effective presentation of students' usage data of the PIVoT web site.

In *The Visual Display of Quantitative Information* (Tufte 1983¹⁴), Tufte states that “excellence in statistical graphics consists of complex ideas communicated with clarity, precision, and efficiency.”(p13) He produces a list of basic guidelines, which are as follows:

- Show the data

¹⁴ Tufte, Edward, *The Visual Display of Quantitative Information*. Cheshire, Connecticut: Graphics Press, 1983

- Induce the viewer to think about the substance rather than about methodology, graphic design, the technology of graphic production, or something else
- Avoid distorting what the data have to say
- Present many numbers in a small space
- Make large data sets coherent
- Encourage the eye to compare different pieces of data
- Reveal the data at several levels of detail, from a broad overview to the fine structure
- Serve a reasonably clear purpose: description, exploration, tabulation, or decoration
- Be closely integrated with the statistical and verbal descriptions of a data set

One of the many types of graphs he brings up and scrutinizes over in this book is a time-series plot. It is the most frequently used form of graphic design, with one dimension “marching along to the regular rhythm of seconds, minutes, hours, days, weeks, months, years, centuries, or millennia.” The natural ordering of the time scale gives this design strength and efficiency of interpretation “found in no other graphic arrangement.” Tufte claims it is best for big data sets with real variability, since simple linear changes can usually be better summarized in one or two numbers (p29). In our case, the large variation of nodes/files encountered in a student’s navigation trajectory makes it a suitable candidate for the time-series plot.

In the late 16th century, J.H. Lambert and William Playfair both experimented with graphics of multivariate arrangements that were not dependent on direct analogy to the physical world. This allowed any variable quantity to be placed in relationship to any other variable quantity, measured for the same unit of observation. About 40 percent of published graphics in modern scientific literature have a relational form, with two or more variables (none of which are latitude, longitude, or time). The graphic encourages the viewer to assess the possible causal relationship between the plotted variables. It “confronts causal theories that X causes Y with empirical evidence as to the actual relationship between X and Y.” (p47) One of the barest forms of relational graphic – the scatterplot – is an effective way to show possible correlation between different types of media usage and student performance.

Interestingly, data can be embedded in the data measure itself (where the data measure is the graphical element that actually locates or plots the data, i.e. the dots of a scatterplot). An example of this would be the stem-and-leaf plot, which constructs the distribution of a variable with numbers themselves. (p140) We can make use of this notion in several ways: by linking each of the dots in our scatterplots to the profile of the student it represents; by making the nodes in our time-series plot links to the actual files; and by coloring these nodes to indicate the media type.

Tufte points out that “words and pictures belong together.” (p180) In addition to labeling the axis and the legend, it is nearly always helpful to write messages on the plotting field to explain the data. One just needs to make sure that the words are devoid of “elaborately encoded shadings, cross-hatchings, and colors... run from left to right, the usual direction for reading occidental languages... (and) type is clear, precise, modest..” Following these guidelines, we add more text to our graphics. This would be especially useful in the plot for navigation trajectory, where the user would not be able to deduce which specific file each node actually stands for without explicitly viewing the title.

Proportion and orientation are two subtle points often overlooked in graphics. Tufte asserts that graphics should tend toward the horizontal for several reasons: the human eye is naturally practiced in detecting deviations from the horizon, thus horizontally stretched time-series are more accessible to the eye; it is easier to write and to read words that read from left to write on a horizontal plotting-field; and since the causal variable usually goes on the horizontal, a longer axis helps to elaborate it in more detail. (p186) The most favored horizontal-vertical ratio, in fact, is 1.618 to 1.

In another one of his books, *Envisioning Information* (Tufte 1990¹⁵), Tufte discusses the importance of the micro/macro composition in graphics. Panorama, vista, and prospect deliver to viewers the freedom of choice that derives from an overview, “a capacity to compare and sort through detail.” (p38) Micro-information, on the other hand, “provides

¹⁵ Tufte, Edward, *Envisioning Information*. Cheshire, Connecticut: Graphics Press, 1990

a credible refuge where the pace of visualization is condensed, slowed, and personalized.” Thus micro/macro designs can report immense detail, organizing complexity through multiple hierarchical layers of contextual reading. This principle motivates us to display a “top-view” of aggregated group usage as well as a “zoomed-in” view of individual navigation path.

Tufte addresses the important issue of graphical quantitative reasoning with a simple question: “Compared to what?” (p67) Small multiple designs are effective solutions because they “answer directly by visually enforcing comparisons of changes, of the differences among objects, of the scope of alternatives.” Since the information slices are positioned within eyespan, viewers have uninterrupted visual reasoning and can easily make comparisons. The constancy of the design “puts the emphasis on changes in data, not changes in data frames.” For this reason we split the display area into six sections and use the same scatterplot framework to display correlation between media type usage and student performance. We hope this allows the user to make easy comparisons across the different media types.

3. Design

In the previous sections, we described why it is important to analyze the ways people use a web-based learning environment. The various aspects which need to be evaluated, however, are so numerous that it would be quite difficult to incorporate all or even most of them into one comprehensive tool. To keep within the scope of a M.Eng project, we decided that *Spelunker's Trail* would address the following:

- The *aggregated* usage of each of the media objects, distinctly compared against a common parameter – the students class performance. By “aggregated” we mean the data is collected from a group of students (i.e. a class or a recitation section) and accumulated for some period of time (i.e. over a term or even over the entire history of the course).
- The *individual* navigation activity for a specific student. This data is collected from an isolated session, during which a particular student has spent a continuous length of time (i.e. 10-15 minutes or even an hour or two) browsing through the web site.

The two features above were chosen because together they give a good general understanding of how a multimedia-enhanced web site is used. The first allows a top-level view of the overall user-preference for the different media objects; as well as possibly provide insight on the relative “merit” of these objects (since they are contrasted with the students’ grades). The second gives a lower-level view of how one single student might utilized the web site and more insight into the learning process. The two of them combined provide a simple and comprehensive analysis of usage data, both on a macro- and micro-level. Thus it follows the Tufte principle of revealing the data at several levels of detail.

The rest of this section goes over the two types of data in more detail and how we chose to present them in our tool. We discuss the features and their functionality, the display models employed and the reasons for them. We also describe some of the alternatives explored and the trade-off.

3.1 Aggregated Usage Data

The first feature, where aggregated usage of each of the different media objects are shown against students' class performance, calls for a design which displays the relationship between two variables. A scatterplot is one of the simplest of such designs. In order to allow comparison, we show all six plots on the same screen (since most desktop monitors are large enough for the user to view all of them without having to scroll). The x-axis consistently shows the students' grades, while the y-axis represents the various parameters we want to look at: total hours of usage of PIVoT; the number of visits to Frequently Asked Questions(FAQ), to on-line text(Text), or to self-quiz(Quiz); as well as the number of downloads of Java Applets (Applets) or video clips (Video).

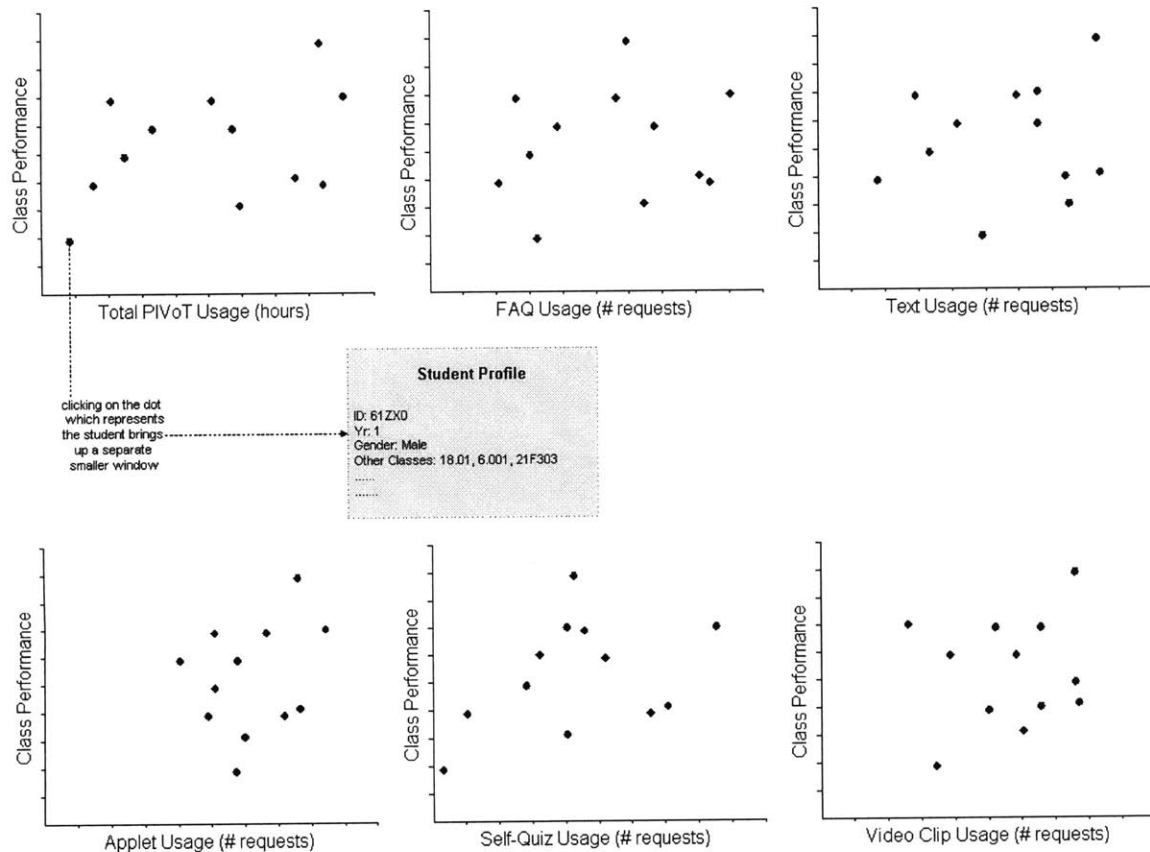


Figure 1 -- WholeClass

Other popular means of showing a large collection of data, such as a bar graph or a pie chart, are not suitable in this particular case. The main reason is that while they are good for displaying one-dimensional data, they are not able to show the *relationship*

between two sets of data – the students’ grades and the particular usage parameter we are examining.

In addition, it is undesirable to show all the parameters on the same graph (which is what a bar graph or pie chart would require), because the quantitative natures vary and the result will be misleading. For example, it may be fairly common for the student to visit many pages of the text before going to a self-quiz page, or to view many video clips of the tutorial before looking at an applet. However, this does not necessarily mean that the text or the video are better methods for delivering course material, since they serve different purposes from the self-quiz and the applet. It is also for this reason that we choose to display the six parameters on six separate scatter plots, each using a different scaling system for the y-axis.

3.1.1 Student Profiles

On each of these scatter plots, a single dot stands for an individual student in the class. These dots are actually hyperlinks, and clicking on one of them opens up a smaller window, which shows a profile of the associated student. The profile may contain, for example, information about the student’s age, gender, prior and current course work, and the type of computer and network connection they have. Thus the evaluator can move from a “macro-view” to a “micro-view” in just one click, learning more about the student represented by a distinct position on the scatter plot.

The purpose of this feature is to explore other factors which affect the way students use PIVoT or how they perform in class. We hope it will reveal “information of association,” such as a link between the type of students who tend to get higher grades in the class and, say, their previous course work (e.g. those who have completed multi-variable calculus consistently get do better on tests, even though this course is not a pre-requisite). Furthermore, we may also discover relationship between the student demographic profiles and their web-learning habits, such as the tendency for male students to download a lot more applets than the female students do.

In addition to providing insight for the education evaluator , this feature is also valuable to the content providers. Since they can keep track of the spectrum of computers and network connections available to the students, they can be sure to not implement functions that would require machines with capabilities beyond those to which most students have access. Therefore, the collection of student profiles should also be accessible without having to go through this evaluation tool.

It is important to note that the real identities of the students are kept secret, either via the “double blind”¹⁶ method or some other approach. Student information should be absolutely confidential because a student’s measured usage of PIVoT and other tools associated with it are not meant *per se* to influence the student’s grade in any way. In other words, usage of PIVoT is not like the often heard phrase, “class participation counts towards your grade.” Mere usage of PIVoT as seen by this monitoring software will not by itself affect the student’s grades. The goal of the whole project is to enhance the learning experience and encourage exploration of the course material outside of the classroom; to factor its usage into a student’s grade induces an artificial incentive, which would go against our original principles.

3.2 Individual Navigation Trajectory

The above section describes the merit of analyzing aggregated usage data of the different media objects. It also gives value to the provision of a direct link between the scatter plot data and the student profiles. Now we want to look beyond these and examine some other components of usage information, such as the navigation path of a typical user session. We can think of this kind of a path as a sequence of nodes, where each node is a media object visited by the student during a particular session.

¹⁶ See Implementation section

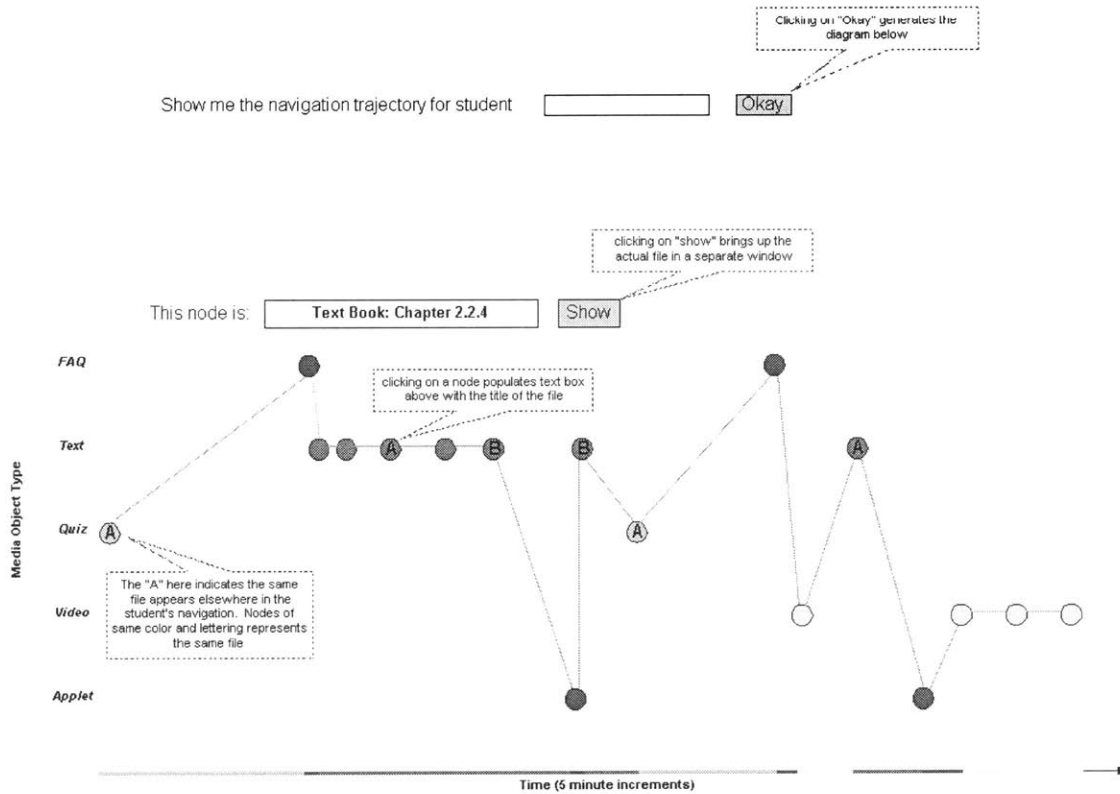


Figure 2 -- OneStudent

The following is a small subset of all the knowledge we might be able to obtain from looking at such a path:

- sequencing of the different objects (how does the student navigate through the web site; e.g. does the student like to take the self-quiz after reading a section of the text, or does she prefer to start with the quiz and consult the text as needed?)
- recycling (whether the students return to previously visited nodes; e.g. does the student tend to revisit a particular page or view a particular video tutorial repeatedly? Does she use them as a gateway to jump to other sections of the system?);
- length of a typical session (how long does the student spend inside this web-based learning environment; e.g. does she spend a longer period going over the material before tests, before problem set due dates, etc?)
- complexity of a typical path (what is the depth and breadth of a the navigation trajectory; e.g. does the student focus on one section of the course at a time, probing depth first, or does she jump between different sections?)

A path is shown as a sequential graph, with time along the horizontal base, increasing from left to right. The five media object types are stacked on top of each other along the vertical axis. The nodes are small filled circles, color-coded according to the media object. Their coordinates are represented by <time of visit, media object type>. The circle may contain a letter, in which case it means this particular node appears elsewhere in the session (i.e. the same node was previously visited/will be revisited).

On top of the display area, there is a list of the titles of all the *unique* nodes visited in this session (i.e. each title is listed only once, regardless of the number of times the associated node had been visited). These titles are links in the form of text, clicking on one opens up a window to bring up the corresponding media object. In addition, all the nodes in the graph which maps to that title are highlighted. (*this feature is not shown in the screenshot in Figure 2*)

Between the list and the graph, there is a text field with a “Go” button next to it. When the user clicks on a node in the graph, this field is populated with that node’s title. Then if the user clicks the “Go” button, a separate window brings up that particular media object, as described above.

Lastly, the horizontal axis is colored in segments, according to the media object type which occupied that segment of time. For example, if the student was looking at a quiz page between the 5th and the 23rd minute, then that portion of the time axis will have the same color as nodes which represent quiz pages.

This model makes it easy for an evaluator to see about behaviors such as some of these mentioned above. Sequencing is immediately apparent by observing the various colors *and* heights of the nodes. Recycling is easily discovered by looking for nodes with the same lettering among those of the same media object type (i.e. searching along horizontally). Duration of the session is the x-coordinate of the exit node, which appears at the rightmost of the graph. Path complexity is not nearly as simple to determine, but

one is able to at the title list of the nodes which the student has visited. This gives a good idea of the areas that were explored and during which times (since the various nodes in the time-series which corresponds to the clicked title are highlighted).

A navigation trajectory holds a wealth of information, much of it may not be obvious to us right away. Thus, we want to *mine* the set of data so that future investigations can use different parameters to retrieve needed information. To accomplish this, we need to keep the collection of the visited nodes, as well as the sequence, in a data structure. This issue is one which concerns both design and implementation (see the implementation section)

3.2.1 Sequential vs. Other Display Models

Although the idea of displaying a user's navigation history seems uncomplicated, finding an effective display model turns out to be a challenge. What we need is a simple, intuitive, and visually attractive mechanism. We want to reveal as much information as possible, but to follow Tufte's principles, we do not want to clutter the display area with anything that is excessively decorative and overwhelms the data itself. We also prefer something with easy implementation up front and relatively effortless maintenance in the long run. This means the system should scale nicely to adapt to dynamic changes in data (i.e. the addition/removal of media objects within the system, the varying lengths of session duration, etc)

We started with several options: regular connected graph, text display, sequential graph, and tree display mapped to a hyperbolic plane. Some of them are more commonly used than others; each has its own advantages and disadvantages when placed within the context of our needs. The rest of this section briefly describes each, explains why we chose the sequential display model, and discusses the trade-off.

- **Connected network:** this model allows bi-directional pointing between the nodes, and any node in the collection can point to any other node. This means each unique one needs to be displayed only once, regardless of the number of times it was actually visited. This kind of graph is effective in showing recycling – a node which has

several arrows pointing to it is intuitively recognized as having been visited multiple times. But the display quickly becomes messy when the number of interconnection between nodes increases. For example, if the student is in the habit of using the “Back” button to get back to previously viewed pages, he creates a lot of extra arrows and it becomes hard to distinguish between different ones. Furthermore, we would need to come up with a scheme for determining where each node should be placed on the graph, so that we optimize the given space. This task is non-trivial, especially since the total number of nodes visited varies by student.

- Textual display: this is probably the easiest model to implement. It can have several variations and our design actually includes one version -- the list of node titles above the graph. By itself, textual display is generally uninteresting and not visually engaging (it would be like reading one phrase after another). In addition, we cannot easily manipulate it to effectively demonstrate “navigation elements” such as recycling or session lengths. However, it can be a valuable supplement to a more abstract display model. At a glance, it states what the graphics represent without further user action.
- Hyperbolic mapping: this display model was developed by Lamping and Rao of Xerox Palo Alto Research Center in 1994.¹⁷ The concept is so new that it is perhaps the least popular of the ones we describe here. By mapping a hyperbolic plane onto the unit disk and using pointer dragging to shift focus, it is effective for visualizing and manipulating large hierarchies. However, it is not quite so suitable for a world wide web hierarchy because the system tends to be very flat, with few generations but many children at each one. Therefore, we unfortunately cannot take advantage of the fisheye technique employed by this model.
- Sequential graph: the basic model simply displays the data along some kind of a timeline. In our design, we added a dimension by also showing the various media

¹⁷ http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/jl_bdy.htm

object types. Unlike the connected network, a sequential graph scales very well – additional links or nodes are simply added in the order they occur. Thus, the display stays neat and easy to read. The implementation is relatively simple as well, the position and appearance of nodes are systematically determined by the time of visit and the object type.

Unlike the connected network, the same object visited more than once are represented by distinct nodes on the sequential graph. While this compromises the intuitive perception of recycling, it eliminates the clutter that can easily grow out of control. We think it makes for a worthy trade-off, especially since we make use of another effective mean of indicating recycling (the labeling of nodes).

Another strength of the sequential graph is the obvious and persistent concept of time. It naturally incorporates time as a parameter, unlike the other models where we would need to insert an additional indicator. A related factor is the clear presentation of the “flow” of the path: the unidirectional model does not require the dizzying task of having to follow the correct ones among many arrows in a network.

Our tool would be more powerful if we can display the entire system of objects as a background, then highlight the path taken by the individual student. This would provide a consistent backdrop for all the different trajectories and emphasize the contrast among different navigation patterns. Unfortunately, such an endeavor appears unrealistic. Even with innovative solutions such as the hyperbolic mapping described above, the files are so inextricably interconnected and the hierarchy so flat and wide that there is not an elegant way of displaying the web content in its entirety. The challenge is further exacerbated by the fact that there will constantly be more files added to the collection as the content providers continue to enrich the hypercourse.

4. Implementation

We are in the process of developing a prototype of this tool using Java applets. The prototype is a proof-of-concept upon which a more robust model can be built later. Due to time and resource limitations, we focused on the most important portions of the design and a few features were not implemented. In addition, since some elements of design depend on uncompleted parts of the PIVoT site, we came up with the needed data in order to illustrate the functionality. Eventually these data will be generated by the PIVoT web server.

In the rest of this section, we first outline the front-end system by describing the user experience. We explain the implementation of the graphical user interface (GUI) and user-generated actions. Along the way we provide concrete methods for adding the features which we did not build. Then we describe the back-end portion *for the prototype* (where we had to handcraft the data at this time), and suggest approaches for the eventual integration with the PIVoT server (which is how the final product should be set up). In addition, we address the issue of confidentiality of the student profile and how it can be accomplished.

4.1 User Experience

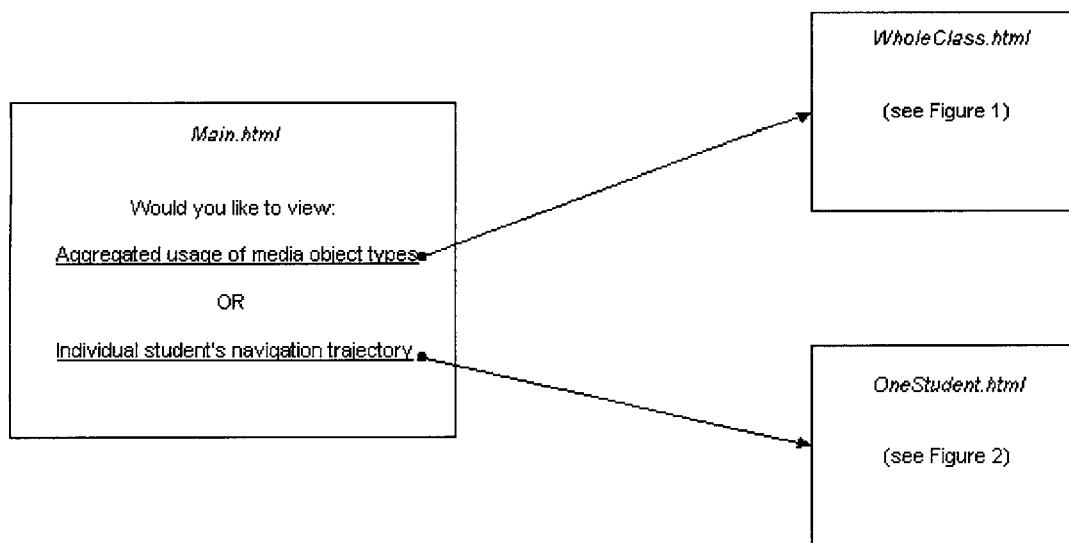


Figure 3 – Flow of User Experience

Our tool is built as a Java software package. The user accesses it by visiting a World Wide Web site. The main page of entrance(**main.html**) has a menu, which allows the user to select one of two choices via the radio button: display aggregated usage data for an entire class, or display individual navigation data for a single session. Based on user input, a JavaScript embedded on this HTML page points the browser to the appropriate subsequent page.

4.1.1 Viewing Class Usage

When the user clicks on the top button in **main.html**, the script takes their browser to a page which displays the six scatter plots (**WholeClass.html**). The display portion is generated by a Java Applet (**WholeClass.java**). This Java file's main functionality is to execute the following three actions:

- Read data from a text file¹⁸ and create an array of Student objects (**Student.java**), where each object represents a student of the class(or some kind of group entity).

In addition to displaying the usage information, we also want to *keep* this information in a data structure for further exploration in the future. This is done by maintaining an array of Student objects. A Student is a Java class which has the following variables:

1. ID: something to “tag” the student with, not his real MIT ID but one assigned to him based on it.
2. perf: how the student is doing in the class; the simplest way may be to just use the average of all his test or homework scores, but there are certainly more sophisticated and accurate gauges. In fact, this can eventually be turned into an array to track a multiple of factors which indicate performance.
3. hrs: the number of hours this student has spent using PIVoT this semester (or some other length of time)

¹⁸ This text file contains cumulative usage data for each student of the class; our Java file simply parses each line as a text string.

4. `faq`: the number of requests for FAQ files that was sent from the student's browser
5. `vid`: the number of requests for video files
6. `app`: the number of requests for the Java applet files
7. `text`: the number of requests for the on-line text files
8. `quiz`: the number of requests for the self-quiz files

This class also has methods to return the value of each of the variables above.

- Split the display area in to six equal portions and display the scatter plots of student performance vs. total usage hours, FAQ hits, video clip requests, applet downloads, text page hits, and quiz page hits.
- When the user clicks on one of the dots on any of the scatter plots, get the coordinate of that dot and search for a corresponding entry in the array; bring up another window to display the profile of the associated student

4.1.2 Student Profile, Revisited

We implemented an example of this functionality by simply using the dot as a link to a static HTML page that contained the profile. However, on a large scale, a better approach would be to dynamically generate such HTML pages directly from the class database, which is constantly being updated as the term progresses. Ideally, administrative tasks for the class are processed in a “paperless” environment. Thus, we get this kind of information for free when the TA enters it into the student record for class book-keeping.

As discussed in the Design section above, it is very important to keep the student profiles completely confidential. A popular way of achieving this is the “double blind” method, where the real data is encrypted with a secret key only known to a very few (say, the head TA), then the result is scrambled yet again to map to a completely unrecognizable set. This pretty much ensures no one but the proper authorities will have access to the

information in question. In this particular case, the real data would be the students' MIT ID, and the result of the double blinding are the IDs which we assign as one of the variables of the Student object.

4.1.3 Viewing Individual Navigation Trajectory

When the user clicks on the lower radio button, the script takes her browser to a page (**OneStudent.html**) which allows her to view a particular student's navigation trajectory for a single session. The display is generated by another Java applet (**OneStudent.java**) and splits the screen into three areas: the top is a text field for the user to enter the student's ID, the middle is a text field which shows the node selected and the option to open the associated file; the bottom is the sequential graph of the path. This file essentially performs the following:

- split screen and draw the top (for obtaining user input) and the middle (left empty for now)
- upon the pressing of the "Go" button from the input portion, take the content of the text field and append ".txt" to the end
- search for a file of this name (which ends in .txt)
- read data from this text file and create an array of Node objects (**Node.java**), where each Node represents a file requested by the student's browser.

Just as we wish to keep record of the student usage information for the whole class, we also want to save information for a completed path. So we keep an array of Node objects, where each object has the following variables:

1. title: a string to indicate the name of the file
2. fn: a string which is the actual name of the file
3. type: an integer which represents the media object type

4. *x*: a double which is the x-coordinate of the node in the sequential graph (based on time of request)
5. *y*: another double which is the y-coordinate of the node (based on media object type)
6. *letter*: a character which, if it is not empty, indicates the node has been visited previously or will be revisited. It would be the identical to that of other nodes which represent the same file

As with the Student object, Node has methods to return the values of the above variables. In addition, it has a method to open up the file that has *this.fn* as its filename. This method is called by OneStudent.java when the user clicks on the “Show” button.

Creating a Node is more complicated than creating a Student, since the variables *type*, *x*, and *y* need to be calculated (although the calculations are simple), and *letter* requires comparison to all the previous nodes titles (which means n^2 time).

- Plot the sequential graph, which is an instantiation of a TimeSeries class (**TimeSeries.java**). This class extends the Canvas class, and in addition to painting the graphics, it handles the user-action of clicking on a node. It has a method which gets the coordinates, searches for the corresponding entry in the array, and populates the middle text field with the correct title. It has another method to return the Node object clicked.
- When the user clicks on a node, all these Java objects go to work: the TimeSeries object finds out which Node it is and shows the title in the text field; if the user clicks on “Show,” the OneStudent object looks at this Node object and calls its method for opening the actual file. Then a separate window is launched to show this file in the appropriate mode – browser for HTML page, applet viewer for Java applet, or Real Player for video clip.

Note here that we simply crafted data for what we believe represents a typical session of using PIVoT. Realistically, however, the data may be very different. And we may want to eventually keep a collection of completed sessions, which would allow the user to choose among a multiple of them from the same student.

4.2 Data Files

The data we used for this prototype were hard-coded into text files. This allows easy parsing for building the data structures we need. Ultimately these files should be generated from the web server for the hypercourse. The process would involve going through the entries in the access log, noting the IP address, the requested file name, and the timestamp of the request, then writing this information to a text file in the format that is readable by our program. A server-side script should be able to accomplish such a task efficiently.

Some kind of naming convention is required to distinguish between all the media objects. When viewing the access log of a web server, requests for Applets and video clips are usually obvious because the requested file name has a distinct extension. However, the other media objects are all HTML files, thus we would need to have a system which tags the file in order to avoid confusion.

The data structures created by this tool (i.e. the array of Student objects for a class's population or the array of Node objects for a session's path) can eventually be stored into databases. It would be a good idea to take advantage of data mining technologies since, as mentioned previously, we may not see all the information which is useful to us right away.

5. Discussion/Future Work

Throughout this paper we have been using PIVoT as an example to illustrate the functionality of the evaluation tool. However, this tool can be used to analyze the usage of any web site. Since the design calls for appropriately formatted text files as its “database,” it is sufficiently modular for us to not be concerned with the system architecture of the hypercourse. All we need is for the web server which hosts the course site to run a script that parses the access log and generate text files with the relevant information in an appropriate format. Then the data can be read and displayed properly. Of course to truly target a multi-media environment, the user should be allowed to specify which media object types are relevant (i.e. a music class may not have Java Applets, instead there might be wave files, etc).

As with any web-based software, we encountered some problems in technology limitations. One of which has to do with the fact that the server-side access log usually only notes the *initial* file request from the client’s browser. After that, the file is cached locally and subsequent visits are invisible to the server until the browser does a cache clean-up, where the file is removed from the user’s machine. This means we cannot get a truly accurate picture of how frequently the nodes are visited. Unfortunately there does not seem to be a simple way of getting around such a problem at this time. What *could* work, however, is to embed a Java applet in every single document and have it communicate with the server each time the page is loaded.

The current prototype shows one navigation path for each student, which is sufficient as proof of concept. Realistically, however, there will (hopefully) be many sessions per student per semester. Thus when the user enters a student ID, he should be presented with a list of sessions from which he can choose one to view. The concept of a “session” itself is actually ill defined in our case, at least from the server-side. One method of determination is to look at the time lapsed between requests from the same machine – if it is beyond some threshold, say, 10 hours, then we can assume that the student’s browser had left PIVoT. This number is rather arbitrary, and regardless of what we choose as

such a “break-point,” we cannot be certain that the student indeed exited the hypercourse. But what matters is that we see a good portion of the student’s exploration trajectory; it is not quite necessary to view in its entirety.

We want to keep the students’ real identities secret because we do not want to create an artificial incentive to use (or not use) PIVoT. Our goal is to passively observe how they naturally perceive and utilize this new concept of learning. There can be many undesirable effects if identities are not anonymous. For example, when a student believes his usage is being monitored, he may log in much more frequently to give the impression of being diligent. Or, on the other hand, he may avoid logging on at all so that his performance will not be tracked.

As class administration continue to evolve into a virtually “paperless” process, student profile information for this tool can be obtained with little additional effort: as the T.A.’s enters grades into the class database, the data should automatically update the student profiles as well. In this sense the tool is quite low-maintenance, however the course staff and faculty would certainly want to pay attention to the analysis of the student usage results and update the hypercourse content accordingly.

6. Conclusion/Summary

The goal of this project is to develop a tool which can be used to effectively display student usage information of a web-based learning environment. Its aim is to allow an educational evaluator to easily look at these information and make assessments on the merit of high-tech, multi-media learning, as well as to improve the structure and content of the hypercourse. We hope the two chosen areas – total accumulated usage of the various media types and individual navigation paths – are presented effectively and are helpful in achieving this purpose.