Graphical Web Mining Agent for Class Teaching Enhancement

P. Madiraju*, Y. -Q. Zhang, S. Owen, R. Sunderraman and Y. Zhu
Department of Computer Science
Georgia State University
Atlanta, GA 30302-4110
{cscpnmx, yzhang, sowen, raj, yzhu}@cs.gsu.edu

ABSTRACT

When a user visits a website, behind the scenes the user leaves his/her impressions, usage patterns and also access patterns in the web servers log file. This paper presents the design and implementation of a Graphical Web usage mining agent for analyzing web log files. In our approach we create our own log file and also provide visual analysis of the web site activity. The application has been experimented for analyzing students activity on course website and the results obtained aided in the enhancement of class teaching. The results helped in improving the course website. External links pertinent to the course material were added to the course website after analysing students behavior. For instance, after visiting the course website, if a student searches for some material on the Internet and finds the material useful, we add such links to the course website, obviating the need for other students to search for the same material. We experimented with two different technologies: Common Gateway Interface- Practical Extraction and Report Language (CGI-Perl) and Java Servlets with back end Oracle 9i database. We have compared both the applications in terms of speed and efficiency and the experiment results are shown.

Keywords: e-Learning, Web mining, Intelligent Agents, Web logs, e-Education, Digital library.

---

* Contact Author

1. INTRODUCTION

The World Wide Web is a popular and interactive medium to disseminate information today. The web is huge, diverse and dynamic and thus raises the issue that we are drowning in information and having information overload. Most of the information is presented on the web page. Design of a web site centers around organizing information on each page and the hypertext links between the pages that seems most natural to the site users for their browsing. For small sites an individual web designer's intuition might be sufficient. However, as the complexity of the sites increases, we need to understand user's access behaviors and appropriately design our web sites. Web usage mining, which is the process of applying data mining techniques on the web usage data (typically web server log files), is a powerful technique for analyzing log files [1].

This paper presents a web application for analyzing the log files. Analyzing log files has important applications in the following areas:

Web site design: re-organization of link structure or the content of the pages to reflect actual usage.

Business/Marketing decision support: determination of common behaviors of users who perform certain actions such as buying or selling merchandise

Class Teaching Enhancement: determination of students activity on course website. If a student cannot find material on a topic, appropriate links are added by analyzing student's activity.

Personalization: customization of page views based on information gained about each user. This can include dynamic pricing of goods based on the users interests, offering deals on the products dynamically for each user.

Usability Studies: Determination of the web site interface quality. On a certain web page, if most of the users do not spend much time and if they always change to other pages, that gives an indication that the interface may not be appealing to the users.

Security: Determination of "unusual" access to secure data.

Network traffic analysis: Determination of equipment requirements and data distribution in order to efficiently handle site traffic.

There has been a lot of research taking place in the field of web mining. Many applications have been developed before like, SurfAid & SpeedTracer from IBM, Bazaar Analyzer, etc. Some of the other popular tools in this area are: http-analyze [2] and access watch [3]. Http-analyze is a tool, which gives statistic reports of the web site usage. However, it can be used only with Netscape servers and Apache server. Access patterns of web users are extracted and analyzed [4]. They define a term called session (a single episode of interaction between a web server and a user) and they identify such sessions in the servers log file and finally cluster them using a hierarchical clustering method. The analysis and experiments were carried on the server log file.

The rest of the paper is organized as follows. Section 2 discusses the main concepts of data mining and web mining. The section further presents the taxonomy of web mining and then gives an understanding of the web server log files. Section 3 gives the implementation of the web applications. Section 4 presents the design of the application using both CGI and servlet technologies. Section 5 presents the performance analysis of using both the techniques of servlets and CGI for implementing the application and gives some future directions in which the application and the paper can be improved.

2. WEBMINING: SPECIAL CASE OF DATAMINING

2.1 Data Mining

Data mining is defined as the extraction of hidden predictive and descriptive information from large databases. It could also be defined as the non-trivial extraction of implicit, previously unknown, and potentially useful information from data [5]. The capabilities of data mining include automatic predictions of trends and behaviors, and automatic discovery of previously unknown patterns.

Primary goals of data mining are

(1) Prediction: Involves using some variables or fields in the database to predict unknown or future values of other variables of interest. Here we might use statistical analysis to predict future values, and (2) Description: Focuses on finding human-interpretable patterns describing the data. This is the most crucial aspect of data mining. "Analytical approaches that search data sets on the basis of known patterns are not data mining, but the main focus of data mining application is to discover hidden patterns [6]". But still most of the applications that we see are mainly oriented towards solving a known problem.

Once the goals of the data mining application that we are developing are identified. The starting point of the application would be Data modeling. "Data Modeling is done in order to translate raw data structure in to a format that would be used for data mining [6]". This is an important step in any application. Data mining uses sophisticated statistical analysis tools and modeling techniques to discover patterns and relationships hidden in the data, the data that ordinary methods might miss. Data mining determines the patterns in the data, which is non-trivial, valid, novel, potentially useful, interesting, general and simple and understandable.

2.2 Web Mining

Web mining can be broadly defined as the discovery and analysis of useful information from the World Wide Web. Web mining can be classified broadly in to web content mining and web usage mining (see Figure 1). Since the content of a text document (typically the information that is kept in the world wide web such as the HTML) presents no machine-readable semantic. Some approaches have suggested restructuring the document content in a representation that could be well understood and could be easily programmed by machines. There are two groups of web content mining strategies: Those that directly mine the content of documents and those that improve on the content search of other tools like search engines. In recent years these factors have prompted researchers to develop more intelligent tools for information retrieval, such as intelligent Web agents, as well as to extend database and data mining techniques to provide a higher level of organization for semi-structured data available on the Web.

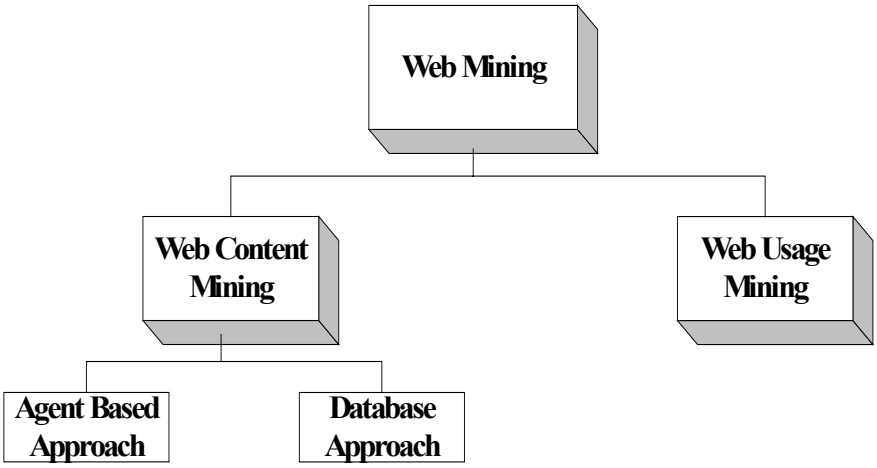

Figure 1: Taxonomy of Web Mining

2.3 Agent based approach

The agent-based approach to Web mining involves the development of sophisticated AI systems that can act autonomously or semi-autonomously on behalf of a particular user, to discover and organize Web-based information. Generally, the agent-based Web mining systems can be placed into the following three categories:

1. Intelligent search agents: several intelligent Web agents have been developed that search for relevant information using characteristics of a particular domain (and possibly a user profile) to organize and interpret the discovered information. For example, agents such as ParaSite [7] rely either on pre-specified and domain specific information about particular types of documents, or on hard coded models of the information sources to retrieve and interpret documents. Other agents, such as ILA (Internet Learning Agent) [1], attempt to interact with and learn the structure of unfamiliar information sources. ILA learns models of various information sources and translates these into its own internal concept hierarchy.

2. Information filtering/categorization: A number of Web agents use various information retrieval techniques and characteristics of open hypertext Web documents to automatically retrieve, filter, and categorize them. For example, HyPursuit [8] uses semantic information embedded in link structures as well as document content to create cluster hierarchies of hypertext documents, and structure an information space.

3. Personalized web agents: another category of Web agents includes those that obtain or learn user preferences and discover web information sources that correspond to these preferences, and possibly those of other individuals with similar interests (using collaborative filtering). For example: WebWatcher [9]. Web agents such as Letizia[10] tracks the user behavior on the web and attempts to anticipate items of interest by doing concurrent, autonomous exploration of links from the user's current position.

2.4 Database approach

The database approaches to Web mining have generally focused on techniques for integrating and organizing the heterogeneous and semi-structured data on the Web into more structured and high-level collections of resources, such as in relational databases, and using standard database querying mechanisms and data mining techniques to access and analyze this information.

Several researchers have proposed a multilevel database approach to organizing Web-based information. The main idea behind these proposals is that the lowest level of the database contains primitive semi-structured information stored in various Web repositories, such as hypertext documents. At the higher level(s) Meta data or generalizations are extracted from lower levels and organized in structured collections such as relational or object-oriented databases.

There have been many Web-base query systems and languages developed recently that attempt to utilize standard database query languages such as SQL, structural information about Web documents, and even natural language processing for accommodating the types of queries that are used in World Wide

Web searches. A few examples of these Web-base query systems are here. W3QL [11]: combines structure queries, based on the organization of hypertext documents, and content queries, based on information retrieval techniques

2.5 Web Usage Mining

Web Usage Mining is the application of data mining techniques to Web click stream data in order to extract usage patterns. As Web sites continue to grow in size and complexity, the results of Web Usage Mining have become critical for a number of applications such as Web site design, business and marketing decision support, personalization, usability studies, and network traffic analysis. The two major challenges involved in Web Usage Mining are pre processing the raw data to provide an accurate picture of how a site is being used, and filtering the results of the various data mining algorithms in order to present only the rules and patterns that are potentially interesting. Web servers record and accumulate data about user interactions whenever requests for resources are received. Analyzing the web access logs of different web sites can help understand the user behavior and the web structure, thereby improving the design of the databases.

Similar to [12], we can decompose web mining in to these tasks :

1. Resource finding: the task of retreiving intended web documents.
2. Information selection and pre-processing: automatically selecting and pre-processing specific information from retreived web resources.
3. Generalization: automatically discovering general patterns at individual web sites as well as across multiple websites
4. Analysis: validation and/or interpretation of mined patterns

By resource finding, we mean the process of retreiving data that is either online or offline from the text sources available on the web such as electronic newsletters, electronic news groups and also the manual selection of web documents.The Information selection and pr-processing process would be any

transformation process of the orginal data. Typically, data mining techniques are used to make generalization. We should also note that humans play an important role in the information or knowledge discovery process on the web, since the web is an interactive medium.

Web server log files are generated, then potentially useful information from the log files is extracted, and finally the information is presented visually in the form of graphs and charts to the user. The last step is to do visual analysis of the collected information from the log files.

2.6 Understanding Web Server Log Files

Consider the example of a log file from a popular web server : Apache. Upon a default installation of Apache, two log files are created. These files are called *access_log* (access.log on Windows) and *error_log* (error.log on Windows). These files can be found in /usr/local/apache/logs. On Windows, the logs will be in the logs subdirectory of wherever you installed Apache.

access_log is, as the name suggests, the log of all accesses to your server. Typical entries in this file look like:

131.96.244.18 - - [28/Mar/2001:14:47:37 -0400] "GET / HTTP/1.0" 200 654

Lets look at each section of this entry.

Address or DNS

131.96.244.18

This is the address of the computer making the HTTP request. The server records the IP and then, if configured, will lookup the Domain Name Server (DNS). However, with all the dynamically assigned IP addresses these days, we don't learn as much as we would expect from the domain name. In this case the visitor's domain name is:

gsu-206475.cs.gsu.edu

RFC931 (or identification)

-

Rarely used, the field was designed to identify the requestor. If this information is not recorded, a hyphen (-) holds the column in the log.

Auth user

-

That is the location where you're supposed to get the identity of the visitor. That's not just their login name, but also their email address, or other unique identifier. This information is supposed to be returned by identd, or directly by the browser. And in the old days, back when Netscape 0.9 was the dominant browser, we would usually have email addresses in this spot. However, it did not take long for unsavoury marketing types to think that it would be a good idea to collect those email addresses and send them unsolicited email (spam). So, before very long, this feature was removed from just about every browser on the market. So we will almost never find information in this field.

Time Stamp

28/Mar/2001:14:47:37 -0400

The date, time, and offset from Greenwich Mean Time (GMT x 100) are recorded for each hit. The date and time format is: DD/Mon/YYYY HH:MM:SS. The example above shows that the transaction was recorded at 02:47 pm on March 28, 2001 at a location 4 hours behind GMT.

Method Resource Protocol

GET / HTTP/1.0

In the example above, the Method is GET. The other most common methods will be POST and HEAD. There are a number of other valid methods, but those three are what we will see most of the time.

The Resource is the actual document, or URL, that was requested from the server. In this example, the client requested ``/'', which is the root, or front page, of the server. In most configurations, this

corresponds to the file index.html in the DocumentRoot directory, but could be something else, depending on the server configuration.

The Protocol is usually going to be HTTP, followed by a version number. The version number will be either 1.0 or 1.1, with most of the records being 1.0. HTTP/1.0 was the earlier version of this protocol, and 1.1 was the more recent version. However, most web clients still use version 1.0

Status Code

200

There are four classes of codes

1. Success (200 series)
2. Redirect (300 series)
3. Failure (400 series)
4. Server Error (500 series)

A status code of 200 means the transaction was successful. Common 300-series codes are 302, for a redirect from http://www.mydomain.com to http://www.mydomain.com/, and 304 for a conditional GET. This occurs when the server checks if the version of the file or graphic already in cache is still the current version and directs the browser to use the cached version. The most common failure codes are 401 (failed authentication), 403 (forbidden request to a restricted subdirectory), and the dreaded 404 (file not found) messages. Severe errors are red flags for the server administrator.

Transfer Volume

654

For GET HTTP transactions, the last field is the number of bytes transferred. For other commands this field will be a hyphen (-) or a zero (0).

Extended log file

131.96.244.18 - -[28/Mar/2001:14:47:37 –0400] "GET/HTTP/1.0" 200 654

http://tinman.cs.gsu.edu/~raj/ Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)

The extended log file records some extra information that might be provide some useful statistics about the person making the request and also about the referring URL. The blod faced words are the extra letters that you see in addition to the regular log file.

Referrer URL

http://tinman.cs.gsu.edu/~raj/

The referrer URL indicates the page where the visitor was located when making the next request. If you were looking at just the referrer log, not integrated into the transfer log, it would be made up of just two fields. The left field is the starting URL and the right field is where the reader went from the URL. Transfers within your site would also show in the access log. For example, movement from one page to another within a web site might show in the referrer log as:

http://tinman.cs.gsu.edu/ -> /tinman.cs.gsu.edu/~raj/

User Agent

Mozilla/4.0 (compatible; MSIE 5.5;Windows NT 5.0)

The user agent is information about the browser, version, and operating system of the reader. The general format is: Browser name/version (operating system). The confusion comes from the word "Mozilla," which is the original code name for Netscape. Now almost all browsers compatible with Netscape use the Mozilla code.

3. IMPLEMENTATION OF THE APPLICATION

In order to create a web application, HTML documents need to be created and published. A World Wide Web (HTTP) Server is required through which the HTML documents and other files can be

published. Regarding the server, we could either set up our own HTTP server on a personal computer, or if we have an account with a server, which has the required specifications, it will work.

3.1 Web Server

The web browser and the server collaborate in what is called a client-server system. The web browser acts as a client, obtaining information from the server over a network. The terminology simply describes the relationship between the two programs: The client asks and the server provides. A closer look at the client-server system shows that the web browser is a program that runs on the client computer system, and the server is a program that runs on a server computer system. The computers are connected by the internet. The web browser and web server work on top of the client and server computer and network. This terminology is sometimes muddled. The client server relationship applies to the computers and to the software(the browser and the server software). The term client sometimes refers to the client computer, and sometimes to the browser, and sometimes to the combination of the two. The term web-server sometimes means the computer, sometimes the webserver software and sometimes the combination.

For this paper the HTML pages are generated from the CGI program. There is no static HTML page, even the first login page is generated from the CGI program.

The tinman.cs.gsu.edu server has Apache installed and the CGI program runs on the Apache. Apache server is scalable and suits very well for unix operating system. The server has been set up on Unix operating system.

3.2 Web Application

Login

The application is secured with a user id and a password. The application is restricted to authorized users only. The application contains sensitive information, such as knowing the IP addresses of the users visiting the site and may also contain some privacy issues. The user enters his or her Login and Password, which are first validated before the users can access other pages. This Login Page is the

default page whenever some one tries to use the application. The same page that is displayed when the user's click LogOut. There is a HTML FORM field , named as Target and whenever Target is '' or 'LogOut' the Login Page is displayed. After entering the Login Id and password, the file AnalyzeLog.cgi gets the value of Login and Password in the variables $FORM{'username'} and $FORM{'password'} and these are validated against the username and password , that we initialize in the application. The users are given a link "Please try again", where the users can click the link and it takes them to the Login Page again.

Main Page

After the user name and password is authenticated, the application takes the users to the main page. The main page is constructed of three parts. Each part is developed by the invocation of a CGI-Perl method. Consider the three parts of the main page.

Header

A method *HTML_Header* contains the HTML display of the header of the page. The header page is always called, whenever, a new page is called. The idea of using the same header page is : the header part is always the same for the entire application.

```
sub HTML_Header {
return <<"END_OF_HTML";
<CENTER>
<FONT face = verdana SIZE=+1><FONT COLOR = blue>Web Mining:Digging into Web Log
Files<BR></FONT></FONT><BR>
  <A HREF="$This_Script_Address">Main Menu</A>
          - Back to <A HREF="$link_url">$link_title</A>
</CENTER>
<BR>
END_OF_HTML
```

```
}
```

After calling the above method, the header part of the whole page is displayed. The body part of the page consists of the whole application. Here we display to the user , how many hits we have currently and the user has a choice to view only the specified number of hits in a data base format or in a nice graphical format. Figure 2 shows these parts of the Main Page. A method *Footer* contains the HTML display of the footer of the page. The footer page is always called, whenever, a new page is called. The idea of using the same footer page is : the footer part is always the same for the entire application. The code for displaying the footer is shown below and when we call this method from the same script, then the footer of the page is displayed. The sample code is shown below.

```
sub Footer {
return <<"END_OF_HTML";
<H5 ALIGN="center">
<A HREF="$This_Script_Address">Main Menu</A>
-<AHREF="Http://tinman.cs.gsu.edu/cgi-bin/cscpnmx/AnalyzeLog.cgi?Target=LogOut">Log
Out</A>
<HR SIZE="1" NOSHADE WIDTH="50%">
</H5>
</BODY>
</HTML>
END_OF_HTML
}
```

Resulting screens after main page

From the main page, let's say a particular user has selected, to view the results in a Data base format and selects the option "Sort all by visitor". Let's consider that the user selects to view the most recent 30 hits. The resulting output screen has this information :

Flow chart of the visitors is shown. Visits are shown with newer hits at the top, and older hits towards the bottom, with timestamps taken from the time of first visit. Successive visits by the same user are grouped together, so that we can view the path taken through the site. A sample output is shown in Figure 3.

On the other hand let us say from the main page, a user selects to view " All the Hyper Links followed from this site " in a graphical format. The ouput is shown in figure 4.

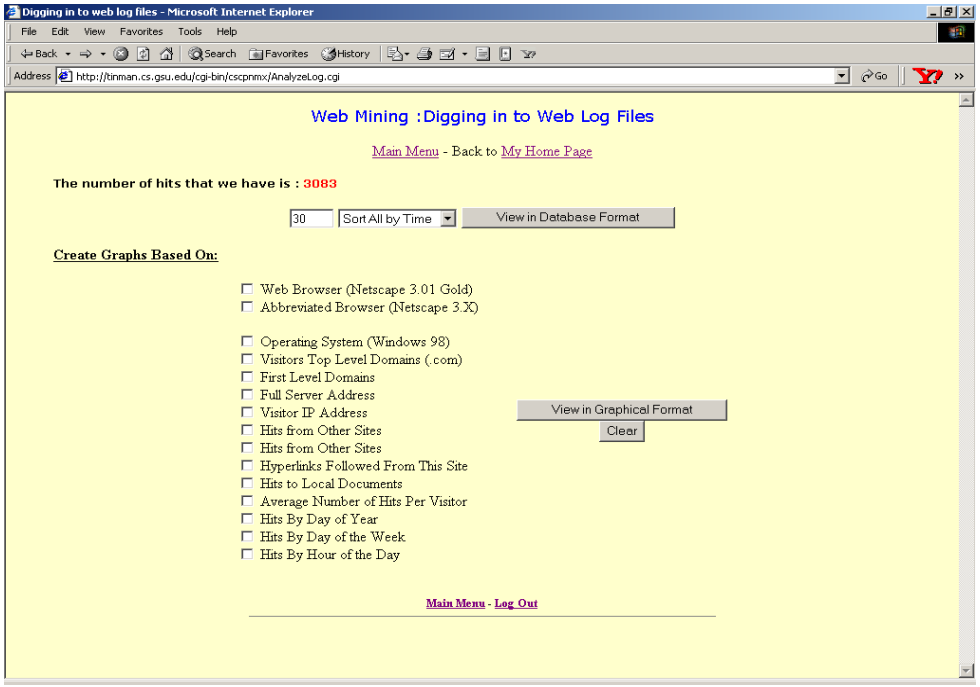

Figure 2: Main Page

Interesting Observations

The system was tested on a classroom taught course at Georgia State University (Csc 2310-Introduction to Programming Languages-I (Java)). The system was deployed on to the course website located at : http://tinman.cs.gsu.edu/~cscpnmx/csc2310.htm and the application has started to run on the course web page from mid march 2001. We found some interesting results. Most of the outside links that were followed were to Dr. Kings Jpb download site: http://knking.com/books/java/jpb/ with 16 % of all

the links followed and 13% of the links were followed to java package index site which is: www.javasoft.com/products/jdk/1.1/docs/api/packages.html

Similarly we had some other interesting results too. However, in order to fully explore these things, we need to use good pattern matching algorithms. Our main intention was to develop an application and look at the analysis in a good graphical and a data base format. For example from the application we could tell that until march 29,2001 20.37 % of the hits were made on Mondays and 20.14 % of the hits were on Wednesdays. The course was taught on Mondays and Wednesdays 7:00-8:15 P.M and this explains the reason why we had most of the hits on Mondays and Wednesdays. Similarly, we can have a very large number of interesting observations, but we shall stop here and look at the basic design of the application.

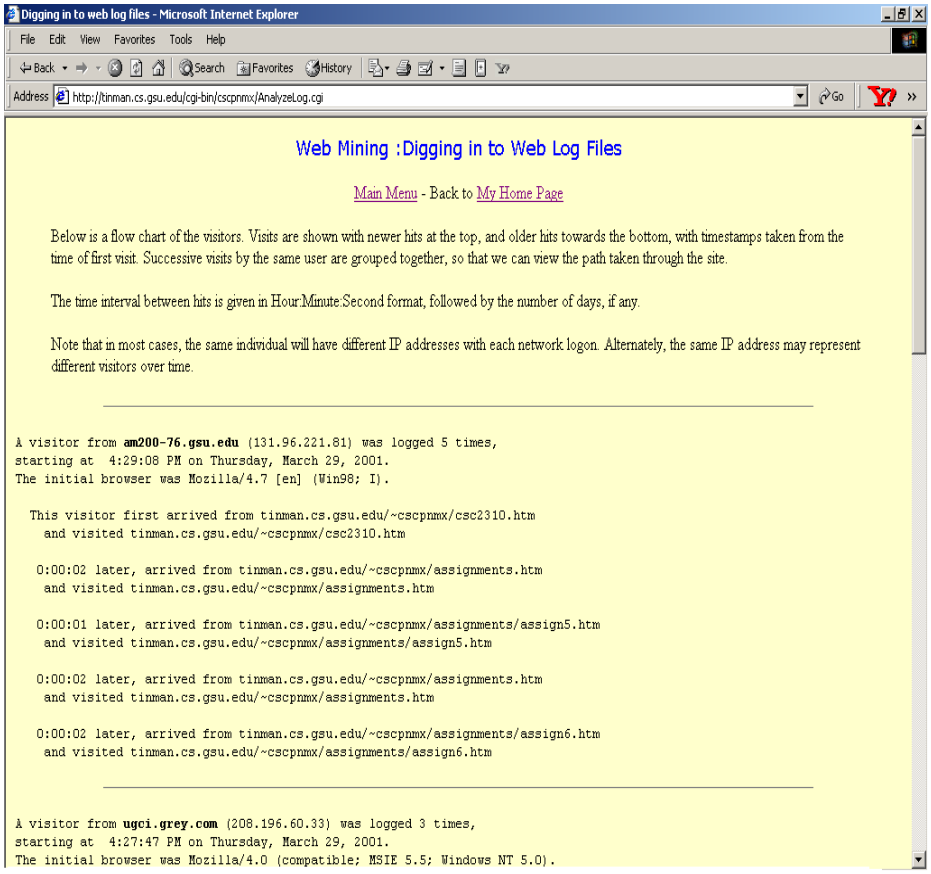

Figure 3: Resulting Page for Visitor Flow (database format)

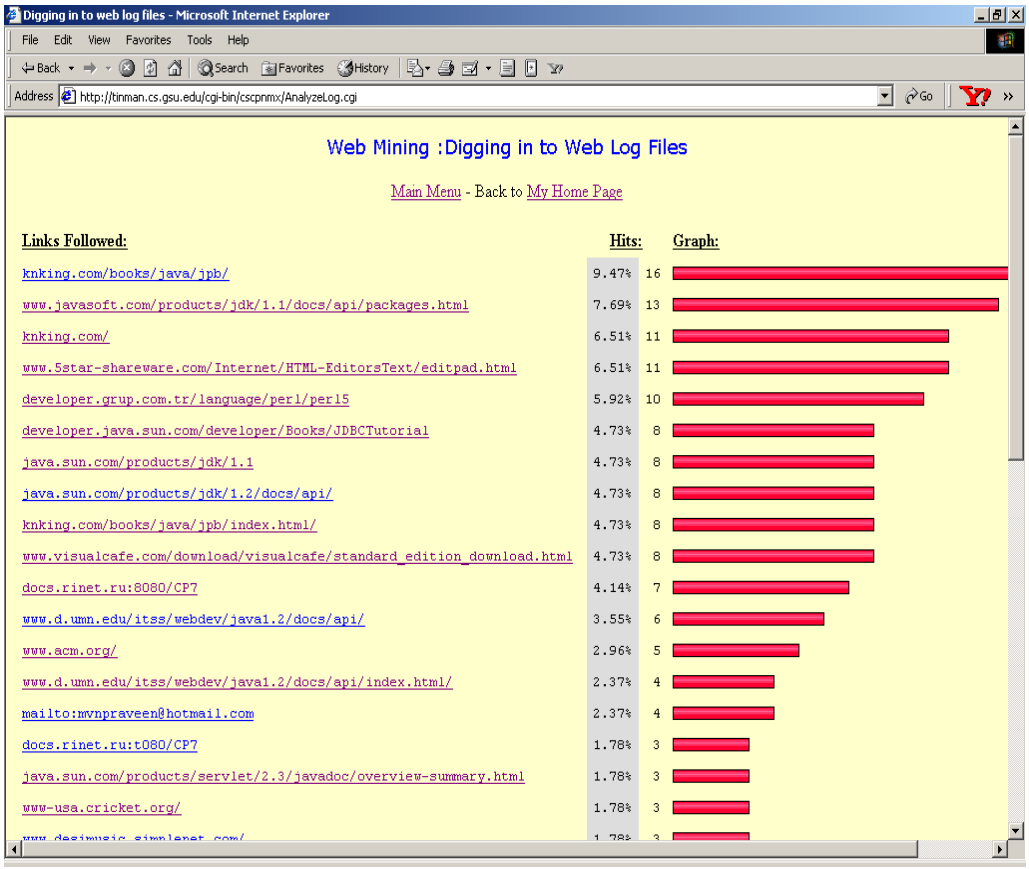

Figure 4: Resulting Page for Links Followed (graphical format)

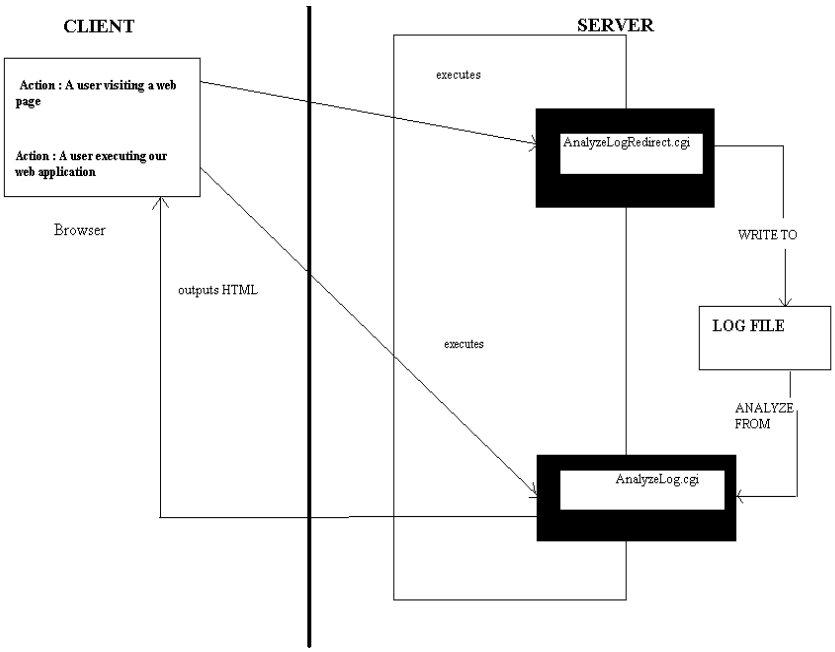

Figure 5: Basic design of the Web usage mining agent

4. DESIGN OF THE WEB USAGE MINING AGENT

4.1 Design of Web Usage Mining Agent using CGI

The basic design of the application is better explained in Figure 5. These are the following steps that are involved in the design of the application :

1. A CGI script that is capable of writing to a log file whenever, a user visits some link or web page or redirects to another page from our site
2. Executing the above script whenever, a user visits a link or webpage or redirects to another page
3. Another CGI script that analyzes the log file and reports the results to the user as desired.
4. Designing the HTML pages and calling the appropriate methods of the above script to show the output.

Figure 5 shows the complete architecture of the Web usage mining agent. The above four steps are explained brielfy here :

Step 1

AnalyzeLogRedirect.cgi is the name of the CGI program that is able to write the log file. All the information for the log file is contained in the Environment Variables. So using the environment variables the CGI program writes to the user defined log file.

Step 2

Inorder to execute the CGI script we place commands like this :

```
<IMG SRC="/cgi-bin/cscpnmx/AnalyzeLogRedirect.cgi?six.gif">
```

This is a HTML image tag and when the web page is loaded, the server tries to display the image six.gif and executes the AnalyzeLogRedirect.cgi program. However, the six.gif doesnot contain anything, so the user does not see any image on the web page, but the CGI program gets executed, which is what we want. For URL redirects, we would write statements as :

```
<ahref=
"/cgi-bin/cscpnmx/AnalyzeLogRedirect.cgi?http://knking.com/books/java/jpb/index.html">
Download Files In jpb package </a>
```

The user is taken to the link : *http://knking.com/books/java/jpb/index.html* when the user clicks on "*Download Files In jpb package*" and before doing this we execute AnalyzeLogRedirect.cgi script and we log the corresponding action in to the web site.

However, instead of using this technique we could also have used SSI(Server Side Include) commands, but some of the servers do not give access to execute SSI commands. Giving access to use SSI commands could be a security threat . so we have not used SSI commands for executing our programs.

Step 3

AnalyzeLog.cgi is the CGI program that analyzes the various information that we have previously stored in the above step. We use Perl's file processing and string processing capabilities and extract all that information the user needs.

Step 4

This is the last step, where the results are sent back appropiately to the browser.

4.2 Design of the Web Usage Mining Agent using Servlets

4.2.1 Database Modelling

The access log records are stored in to the Oracle database. For this purpose, we have used oracle9i. Figure 6 shows the E-R diagram of our database.

Figure 7 shows the relational tables corresponding to the E-R diagram. The USER table stores all the information regarding the users accessing the web page. The columns of the USER table are: UserId, OrigIp, Browser and Os. A unique UserId is created for every access to the web page. Hence UserId is the primary key for this table. This is implemented using a sequence number in the oracle database. OrigIp denotes the originating IP address of the user accessing the web page. Browser is the browser with which he/she is accessing the web page. Os is the operating system of the client. The WEBPAGE tables stores the web page URL. The only column in this table would be the URL's present in the web page. This should also include any outside links that may be present on the web page. When a USER accesses a WEBPAGE, the user's access information is stored in the ACCESS table. ACCESS table has three columns: UserId, Url, Date. Date gives the date on which a user accessed url. The ACCESS table is the most important table due to the fact that it gives information regarding the time spent by each user on a particular table. From these set of tables, we will be in a position to determine the hit ratio, time spent and Hit probability.

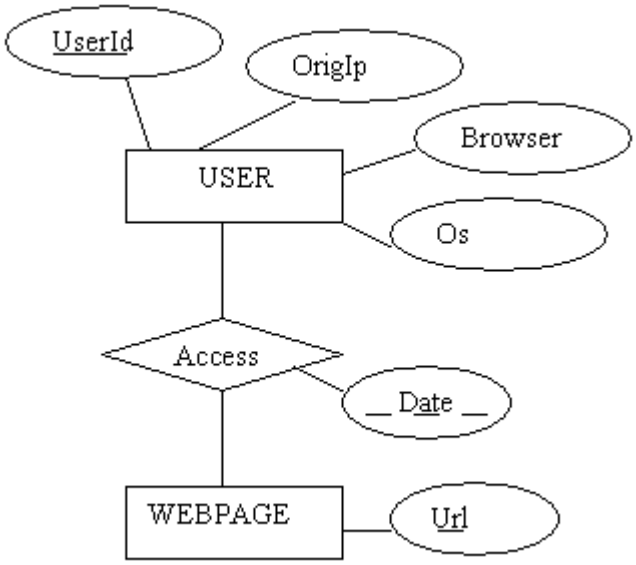

Figure 6: E-R diagram of Web Usage Mining agent

4.2.2 Basic Design of the Application

The basic design of the application is better explained in Figure 8. These are the following steps that are involved in the design of the application :

1. A Java servlet that is capable of writing to the database whenever, a user visits some link or web page or redirects to another page from our site
2. Executing the above servlet whenever, a user visits a link or webpage or redirects to another page
3. Another java servlet that queries the database tables(USER,ACCESS, WEBPAGE) and reports the results to the user as desired.
4. Designing the HTML pages and calling the appropriate methods in the servlet to show the output.

Figure 8 shows the complete architecture of the agent. The above four steps are explained brielfy here :

Step 1

WriteToLog.java is the name of the java servlet that is able to write to the database tables. All the information to be stored in the database tables are obtained by using HttpServletRequest Interface.

Step 2

Inorder to execute the java servlet we place commands like this :

<IMG SRC="http://yamacraw.cs.gsu.edu:7777/cscpnmx/servlets/WriteToLog?dummy.gif">

This is a HTML image tag and when the web page is loaded, the server tries to display the image

USER

UserId	OrigIp	Browser	Os
1	131.96.242.198	MSIE 4.01	Win98
2	205.188.196.22	MSIE 5.5	Win95
3	208.196.60.33	MSIE 5.5	Win00

ACCESS

UserId	Url	Date
1	http://tinman.cs.gsu.edu/~cscpnmx/csc2310.htm	10:00:00-03-20-2002
2	http://tinman.cs.gsu.edu/~cscpnmx/assignments.htm	10:00:02-03-20-2002
3	http://tinman.cs.gsu.edu/~cscpnmx/assignments/assign5.htm	10:00:04-03-20-2002
4	http://tinman.cs.gsu.edu/~cscpnmx/csc2310.htm	10:00:20-03-20-2002
5	http://tinman.cs.gsu.edu/~cscpnmx/lectures.html	10:00:22-03-20-2002
6	http://tinman.cs.gsu.edu/~cscpnmx/lectures/chapter5.html	10:00:26-03-20-2002
7	http://tinman.cs.gsu.edu/~cscpnmx/exams.html	10:00:40-03-20-2002
8	http://tinman.cs.gsu.edu/~cscpnmx/quizzes.html	10:00:44-03-20-2002

WEBPAGE

Url
http://tinman.cs.gsu.edu/~cscpnmx/csc2310.htm
http://tinman.cs.gsu.edu/~cscpnmx/assignments.htm
http://tinman.cs.gsu.edu/~cscpnmx/assignments/assign5.htm
http://tinman.cs.gsu.edu/~cscpnmx/lectures.html
http://tinman.cs.gsu.edu/~cscpnmx/lectures/chapter5.html

Figure 7 : Relational tables for figure 6

dummy.gif and executes the WriteToLog servlet. However, the dummy.gif doesnot contain anything, so the user does not see any image on the web page, but the java servlet program gets executed, which is what we want. For URL redirects, we would write statements as :

```
<ahref=
"http://yamacraw.cs.gsu.edu:7777/cscpnmx/servlets/WriteToLog?http://knking.com/books/java/jpb/index.html">
Download Files In jpb package </a>
```

The user is taken to the link : *http://knking.com/books/java/jpb/index.html* when the user clicks on "Download Files In jpb package" and before doing this we execute WriteToLog java servlet and we log the corresponding action in to the web site.

Step 3

AnalyzeLog.java is the servlet program that analyzes the various information that we have previously stored in the database. We execute queries from the java servlet program using jdbc. Using jdbc we make a connection to the database, send the queries to the database and result of the query is sent back to the servlet program.

Step 4

This is the last step, where the results are sent back appropiately to the browser.

5. PERFORMANCE EVALUATIONS AND CONCLUSION

Web usage mining has become an important area of research as evidenced by the growing number of research projects and commercial enterprises engaged in analyzing the log files and reporting patterns and useful information about the user behavior. The ability to observe potential customers as they browse through a virtual store promises to raise business intelligence at new level and at the same time increases concerns on consumer privacy.

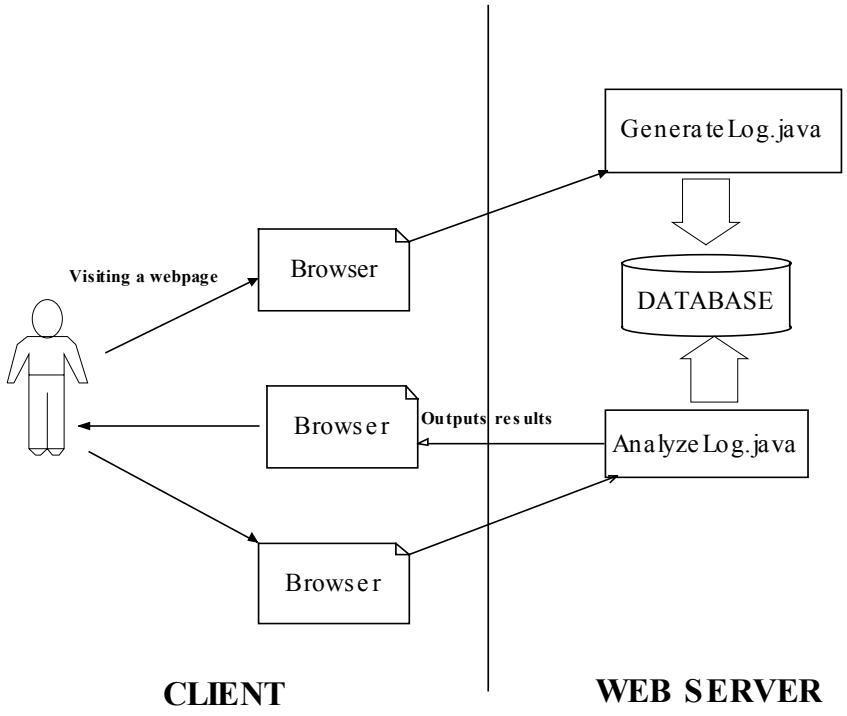

Figure 8: Agent architecture

The application finds tremendous use in Web Site design, Class room teaching(e-Learning), Business/Marketing support, Personalization, Usability studies, Security and Network traffic analysis.

The Web usage mining agent is able to serve our purpose of generating graphical charts and analyzing site activity. In the process of the development, we have experimented with two different technologies. The first technology we used was CGI-Perl, Apache web server and for simple client side validations java script. The second technology we used was Java Servlets, with Oracle 9i as back-end database. We have compared the technical merits of using both the applications. Figure 9 shows the comparison of using both the technologies. The comparison factor is response time. Response Time is the time taken between when the submit button was clicked and when the results were shown on the web page. The results were obtained and as seen in the figure for large amounts of log file, servlet technology with an underlying database has lesser response time when compared with CGI technology using a flat file. The greatest advantage of using CGI-Perl is that it is cost effective and easy to develop. Perl

particularly offers great advantages and is very efficient for an application with a medium sized log files. However, servlet with a database support can support very large amount of log files. Servlet technology offers more security features than the conventional CGI technology.

As part of future work, we can further enhance the graphical charts with 3D visualizations. We can combine multiple statistics in a single view. For example, we can combine time related information with web link hits. Once the log file is analyzed or pre-processed, we can also extend the web application to discover patterns using techniques such as association rules, clusters of similar pages or users, or sequential patterns. There are many algorithms that have been developed based on these techniques and we can directly use those algorithms. Any web application will become ineffective, if not continuously improved and developed, just as the internet as whole is continuously developing.

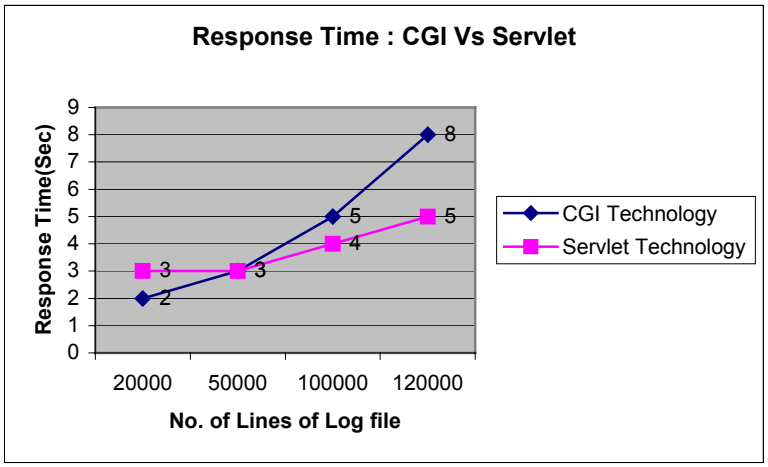

Figure 9: Response Time of CGI vs Servlet

6. ACKNOWLEDGMENTS

The authors would like to appreciate the support by NSF under Grant IIS-9980130 and the ACM SIGGRAPH Education Committee, and thank Dr. Krishnan Balakrishnan for his comments.

REFERENCES

[1] Madiraju, P. and Zhang, Y(2002). Web Usage Data Mining Agent, Proc. of SPIE: Data Mining and Knowledge Discovery: Theory, Tools, and Technology IV, Vol. #4730, April 2002

[2] http: //www.2kweb.net/software/stats/http-analyze/

[3] http://accesswatch.com/

[4] Yonagjian Fu, Kanwalpreet Sandhu and Ming-Yi Shih(1999). Clustering of Web Users Based on Access Patterns, International Workshop on Web Usage Analysis and User Profiling (WEBKDD'99), San Diego, CA, 1999.

[5] Frawley, W, Piatetsky-Shapiro, G and Matheus, C(1992). Knowledge Discovery in Databases: An Overview, AI Magazine, fall 1992, pgs 213-228.

[6] Westphal Christopher and Teresa Blaxton(1998). Data mining solutions: methods and tools for solving real world problems. John Wiley & Sons, Inc. New York, NY, USA

[7] Perkowitz, M and Etzioni O(1995). Category translation: learning to understand information on the internet, In Proc. 15th International Joint Conference on AI, pages 930--936,Montral, Canada, 1995.

[8] Spertus, E(1997). Parasite: Mining structural information on the web, In Proc. of 6th International World Wide Web Conference, 1997.

[9] Armstrong ,R. ,Freitag, D, Joachims, T and Mitchell, T(1995). Webwatcher: A learning apprentice for the world wide web, In Proc. AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments. 1995.

[10] Henry Lieberman Letizia(1995) : An agent that assists web browsing, International Joint Conference on Artificial Intelligence, Montreal, August 1995.

[11] Konopnicki, D and Shmueli, O(1995). W3qs: A query system for the world wide web, In Proc. of the 21th VLDB Conference, pages 54--65, Zurich, 1995.

[12] Etzioni, O(1996). “The world wide web: Quagmire or gold mine”, Communications of the ACM, 39(11), pages 65-68,1996.