

Reverse Engineering of Web Pages with Page Monkey

Joseph Scoccimaro, Spencer Rugaber

1. Introduction

Imagine that you visited a web page and wondered how the author achieved a particular effect in the document. You then proceed to look at the source and are bewildered by the complexity of the information that is shown to you. You think that there must be a better way to get meaningful information from the web page that is understandable. Behold, Page Monkey provides a simple solution to reverse engineer a web page. Page Monkey will present meaningful and understandable information about a web page to you.

Web pages have become quite rich in their expressivity. One often wishes to know how the author achieved a certain effect, but the HTML source code, possibly combined with a scripting language and style sheets, makes the process of understanding a page daunting. Page Monkey is a tool whereby a user can select different types of analyses to perform on a web page. Page Monkey show properties of nested structures, such as frames and tables, style aspects, and active aspects such as links and scripts. The information of a page can be obtained from the Document Object Model (DOM) as specified by W3C and accessed through various APIs. After getting access to the HTML structure, Page Monkey needs to infer higher level abstractions from the information. The key property of our research is to determine the meaning for reverse engineering of a web page. In order to formulate this definition, we need to look at the key principles of web page design. By studying the principles of web page design, we hope to learn what types of information are useful for presenting to the user of our system.

During the process of defining the meaning of reverse engineering a web page, we interviewed an expert to elicit important information about the design of a web page. To begin the interview we asked the expert to tell us his experience level, principles of good page design, and principles of bad page design. The expert has six years experience in designing web pages and web development. Our expert is experienced in layout design, graphic layout, HCI, HTML, and other web page related languages. The expert told us that the number one principle of good web page design for him is usability of the page. He states that if a page is not usable by the user then it becomes useless. Other good design principles include good navigation and a striking design. Good navigation allows a user to find what they are looking for in an easy manner. A striking design is needed to draw attention to your site so that people will visit the page repeatedly. Some bad design principles are poor navigation, non-uniform navigation across pages, and non-uniform design across pages. Poor navigation makes it difficult for the user to find information they are seeking. Non-uniform navigation and design across many pages bewilders the user and makes it difficult to navigate through a web site [19].

For the second portion of the interview, we had the expert look at four different web sites and answer some questions. The questions were about the layout, information density, and navigation of these sites. In analyzing the answers to the questions, we pulled out general information about the important aspects of web design. One of the common themes we saw in the expert's responses was that a page should be designed to meet the user's expectations. For instance, CNN's web site is tailored to users that wish to view important news going on in the world. In order to determine if the web page has

to high of an information density, one must first understand the domain of the web site. A news web site like CNN is expected to have a high density of information for their users. The expert noted on all the sites visited that links should be given descriptive titles so that the user may determine if they wish to follow that link. This helps in improving the navigation of a web page. One key note we obtained from our analysis of the expert's responses was that a page should provide easy access to important information or capabilities. We had the expert visit Newegg.com, which is an online retail store, and the expert mentioned how it was a good design to have the search capability readily available to the user on all pages in the site. The expert found few bad design choices in the sites that were examined. One bad design the expert pointed out were that some of the pages contained too many 3rd party advertisements that took up valuable screen real-estate which is better used for user content [19]. After examining the expert's responses, we formulated a definition of what it means to reverse engineer a web page.

The definition of reverse engineering of a web page is very broad in nature because of the many components related to designing a page. To successfully reverse engineer a web page one must analyze the presentation, content, navigation, usability, and enhancements (JavaScript, etc) of a document. To properly define reverse engineering of a web page, we needed to derive a concept that would tie all the different aspects of a web page together in a coherent manner. One aspect these concepts have in common is there meaning to the consumer of the web page or a person wishing to duplicate a certain effect. In our definition of reverse engineering, one must provide key information to gain insight to the typical user experience of that web page while being able to present how the author of the web page achieved particular effects.

1.2 Related Work

In the process of determining the functionality of Page Monkey, we investigated other approaches to the reverse engineering of web pages. During our research process, we were unable to find a system that dealt with the general area of reverse engineering of a web page. The systems presented below deal with very specific instances of reverse engineering of an online document or a whole web site. Besides this drawback we found that some of the information provided by those systems is useful to us. We also felt that it is important to look at what are considered as proper guidelines for the design of web pages.

1.2.1 Reverse Engineering Systems

Fabrice Estievenart, Aurore Francois, Jean Henrard, and Jean-Luc Hainaut present a methodology to extract data from static HTML pages and migrate them to a database [5]. Their process contains a set procedure to extract the schema and data from the page. The process consists of a classification phase, cleaning phase, semantic enrichment phase, extraction phase, and conceptualization phase. The overall goal of their proposed system is to help in the maintenance of static html pages by extracting the schema and data in order to translate the static pages to dynamic pages that use a database. Page Monkey differs in the respect that our system reverse engineers a page to present higher level information to the user. The authors' system deals with whole web sites and does not give detailed information about a web page to the user. Both systems reverse engineer web pages in different ways and for different reasons.

Jean Vanderdonckt, Laurent Bouillon, and Nathalie Souchon have developed a system called Vaquista to examine the presentation model of a web page. They use this information to migrate the presentation model to another environment. Vaquista performs a static analysis on web pages when given a file or a URL [6]. Page Monkey differs from Vaquista in that we perform an on-demand analysis while the user is looking at the web page in the browser. Our system also performs more types of analysis on a web page. Page Monkey performs analysis on presentation, style, structure, and content of a web page while Vaquista only examines the presentation model.

Vincenza Carchiolo, Alessandro Longheu, and Michele Malgeri analyze the structure and semantics of pages and sites in the purpose of highlighting their structural and semantic organization. They build a logical schema of the structural and semantic organization by attempting to reflect the author's intentions and the criteria used during development. Their system views the web page as a tree of nested HTML tags. They locate specific tags that are then labeled as primary tags if they give structural or logical organization to the web page. They organize their gathered information into tree structures that contain structural information and content analysis information in order to provide a logical and structural analysis of a web site [12]. Page Monkey differs from this system in that we are again an on-demand analysis system compared to a static analysis system such as the one as described by Carchiolo et al. Page Monkey also deals with the Document Object Model of a page compared to actually analyzing the HTML file of a web page. This provides us with a higher level of abstraction than the raw tags of the page and gives us a predefined organizational solution. Furthermore, Page Monkey provides a wider variety of different analysis techniques that this proposed system does not contain.

1.2.2 Web Design Guidelines

Charles M. Hymes and Gary M. Olson present a process for organizing the content of a web page for the user's needs. They say the most difficult problem for the user is to understand how the content is organized within a site. Their techniques help a designer represent the mental model of the user's conception of the content [3]. We believe this to be an important concept for Page Monkey because it allows us to analyze the content of a web page to see if it effectively organizes data to represent a user's mental model. This type of analysis will also allow our system to categorize a web site and get an intended audience of the page.

Cynthia M. Calongne argues that one can use HCI usability techniques to help design a good web site. The goal is to target the identified target audiences and classify them into user classes. She also mentions trying to identify what type of web site it will be, determining the content, and any constraints or boundaries to the content. The author says one should postulate the content going into each page to sure it is a good reason for the page to exist [4]. Page Monkey employs these arguments when analyzing the content of a web page. Furthermore, Page Monkey checks for HCI usability requirements such as Universal Design.

Beverly B. Zimmermann argues that information design principles can be effectively applied to the creation of web pages. The author bases her information on Edward Tufte's general principles of information design [2]. She claims that most advice for web page design is based on what style is popular at a certain time. This doesn't

provide a consistent methodology for designing web pages. The author's approach includes five different information design principles that can be applied to web page design. The five principles are micro/macro design, layering and separation of information, principle of small multiples, tying color to information, and the integration of words and images [2]. The micro/macro design principle states that small details should mix into larger patterns in order to provide an overview of information to the user while containing immense detail. We analyzed all these recommendations and incorporated them into our Page Monkey system. We however do not use the micro/macro design principle in our analysis technique because we feel it is not a necessary feature to reverse engineering a web page.

2. Taxonomy of Analysis

We have created a taxonomy of different types of analysis the system can perform. In order to create the taxonomy we looked at the features of other related projects, page design guidelines from the World Wide Web Consortium, studying the HTML specification, and the interviewing of some web page design experts [17]. The highest level of the taxonomy includes the text analysis, structure analysis, style analysis, augmentation analysis, and media analysis. Figure 1 shows the different levels and types of analysis included in the taxonomy.

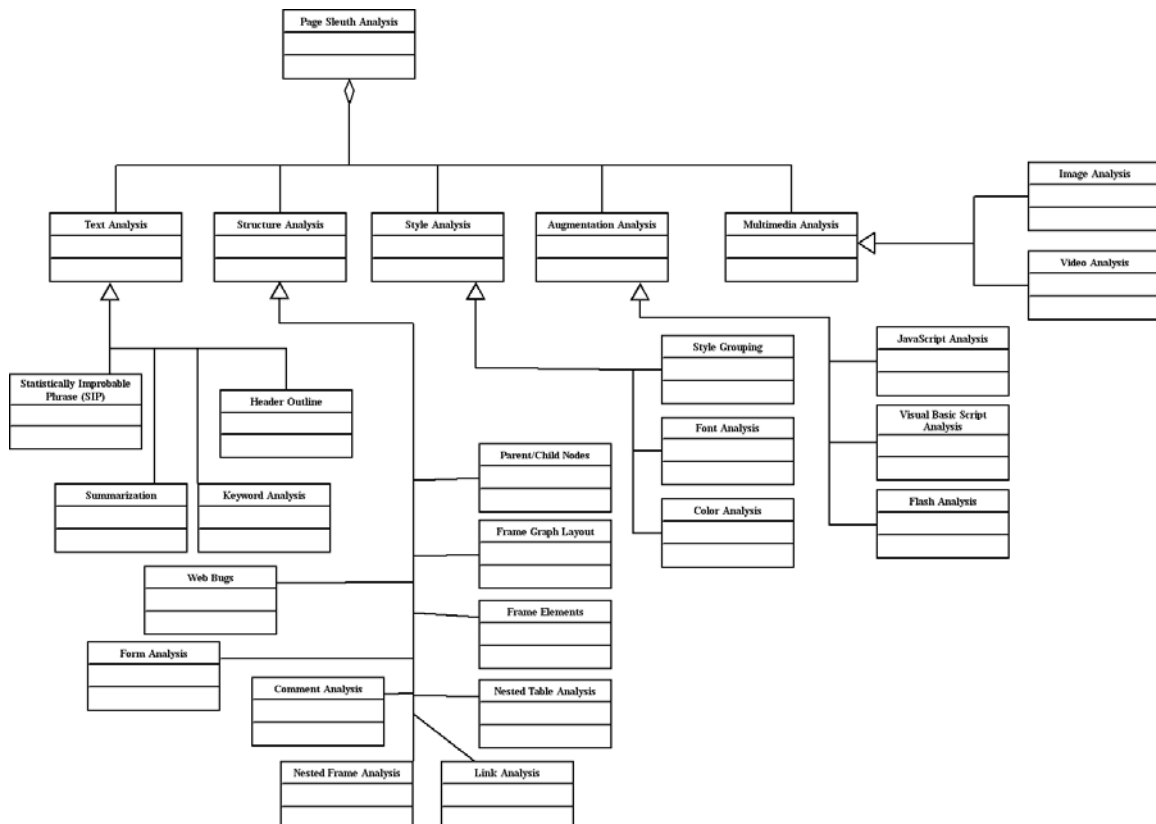


Figure 1: Diagram of Taxonomy of Analysis Techniques

2.1 Description of Analysis Techniques

Intended Audience Analysis

The analysis for intended audience seeks to find the applicable level of understanding of the reader of the web page. By obtaining this insight, a person analyzing the page can learn the difficulty of understanding and intended age level of the intended audience. To aid in this type of analysis one can employ the Flesch-Kincaid Reading Level and Flesch-Kincaid Reading Ease tests to a web page [13]. To perform Flesch-Kincaid Reading Level, one must calculate it with the following formula: $.39(\text{total words} / \text{total sentences}) + 11.8 (\text{total syllables} / \text{total words}) - 15.59$. The Flesch-Kincaid Reading Level translates the understandability of a piece of writing into the different US grade levels. To perform Flesch-Kincaid Reading Ease, one must calculate it with this formula: $206.835 - 1.015(\text{total words} / \text{total sentences}) - 84.6 (\text{total syllables} / \text{total words})$. The Flesch-Kincaid Reading Ease gives a score from zero to one-hundred as to the difficulty of the writing. A lower number means a harder to read page, while a higher number means the page is easier to read [13]. By employing these two reading analysis techniques, the system can successfully estimate the intended audience of the web page.

Table Hierarchy and Frame Hierarchy Analysis

Table Hierarchy Analysis and Frame Analysis both look at the hierarchical structure of their respective html components. The Frame analysis looks at the nesting of framesets and frame tags to determine their structure. Such information allows a person viewing a web page to see how frames are being deployed on the page. The wire frame drawing tool employs the frame analysis algorithm to produce a skeleton drawing of the frame layout of a page. It is used to give the user a visual representation of the web page that is designed with frames. The Table hierarchy analysis examines the different layers of nested tables. The table hierarchy aids the user in learning how a page is organized by using tables or how information is nested within different tables.

Link Analysis

Link Analysis examines the hyper-links embedded in a web page. It looks at how many links are internal, external to the page, external to the web site, or mailto links. The analysis also determines which links are broken and the targets of the active links. This analysis draws a representation of the link structure that shows external links by having lines leave a page and internal links by drawing lines to different points of the page. Broken links are drawn using a wavy line to show the link goes nowhere. Page Monkey implements these techniques by estimating link location obtained from the DOM of the web page. A similar analysis is to look at different image maps of the web page. These types of images contain links in certain areas of the image. The system examines them to see the relevance of the image map to the different destinations of its links.

Table of Contents

The table of contents presents an outline of the web page using all the header tags contained in the document. They are then presented based on their importance. Their importance is derived from the level of header tag. For instance, header ones are

considered to be the most important while header six is the least important of the header tags. By grouping content in this hierarchical way the system easily creates a table of contents for a web page.

Page Complexity Analysis

The page complexity analysis technique determines if it is difficult for a user to effectively understand the information of a page. These types of analysis are important in determining the burden on the user in deciphering the meaning of a web page. Similarly, the system will employ a Universal design analysis along with the page complexity analysis. The universal design of a web page means it will accommodate those with disabilities. An example of testing for universal design would be to see if a web page can support people with eye sight problems or hearing problems. These types of analysis are important for determining the usability of a web page.

Content Analysis

There are many different types of analysis the system does on the content of a web page. Page Monkey performs a Statistically Improbable Phrase (SIP) analysis which compares text in a web page and determines the best phrase to describe the web page. This phrase is then used to categorize a web page. Categorizing a web page is an important analysis technique because it shows the user the intended domain of a web page. Different categories have been created to fit pages into such as a store, news site, and others. Along with these types of analysis, Page Monkey parses nouns and noun phrases to look for interesting words in a web page. Another type of content analysis determines how crowded a page is based on the amount of white space, text density, image density, and link density. We call this type of analysis the “Grunge Factor” of a web page.

Style Analysis

Style analysis allows a person viewing a web page to see how the author performed the presentation features the page employs. The analysis examines the style sheets and style options for relevant information for the presentation of the web page. It allows the user to capture this information so that they may save it into a text editor. By saving the information the user can then use similar presentation techniques as the author of the web page it was obtained from. The World Wide Web Consortium recommends that web page developers separate presentation from structure in the web page. Presentation is defined as the attributes of different tags such as font size and family. According to the World Wide Web Consortium, one should put presentation attributes into style sheets and avoid using tag attributes [18]. Style analysis also looks to see whether or not the page developer has separated structure from presentation.

Augmentation Analysis

Augmentation analysis looks at external features that are not part of HTML that are incorporated in a web page. Such features include JavaScript, java applets, and flash content. The system analyzes such content to see its importance to the web page. One important feature of augmentation analysis is to determine if JavaScript is only employed as a mouse-over feature for links or for other functionality. This type of analysis also lets a user save the structure of these features for future reference. Once saved the user can examine the augmentation code of a page and see what parts they wish to incorporate in their own web pages.

Form Analysis

Form analysis captures the use of forms on a given page. It looks at the different types of inputs and fields employed within a form. The system also allows one to capture output from a form submission to see the interaction in the background. The form analysis also looks at the hidden fields of a form. These hidden fields contain important information for what is done to the data provided in the form when the user submits it.

Meta Data Analysis

Page Monkey examines the meta information of a web page. This information includes the title of the page, the description of the page, and what keywords represent the page. Also, the date the page was last modified is obtained using this analysis technique. Since meta information can contain any type of data in it, Page Monkey also displays any extra information if the author has included it in the web page.

Multimedia Analysis

The final proposed type of analysis is to examine the multimedia content of a web page. For example, animated images and embedded video are examined for their importance to a web page. Page Monkey looks at the different images or sound included into a page to try and get an understanding to the meaning of the web page. Multimedia must be used in a particular manner to be effective to the viewer of a web page. The system analyzes the multimedia content to see if it could overwhelm the viewer of the web page.

The following table shows the priority of analysis techniques, their difficulty, and how the technique can be implemented. The priorities start with the analysis techniques we feel should be completed first. The difficulty rankings include easy, medium, or difficult labels. The three labels were decided upon after thinking of how difficult it would be to implement the analysis technique. The question of whether it can be done lexically is to determine if the analysis technique can be done with just lexing the information and not having to infer higher level details from the low level information. If it can not be done lexically then the analysis technique needs to perform a syntactic analysis of the information.

Priority	Name	Difficulty	Can it be done lexically?	Can it be done Syntactically?	Status
1	Image Map Analysis	Medium	No	Yes	Partially Complete
2	Target Analysis	Difficult	No	Yes	Future Addition
3	Grunge Factor	Difficult	Yes	Yes	Future Addition
4	Flesch-Kincaid Reading Ease	Easy	Yes	Yes	Future Addition
5	Flesch-Kincaid Reading Level	Easy	Yes	Yes	Future Addition
6	Intended Audience	Easy	Yes	Yes	Future Addition
7	Noun and Noun Phrase Analysis (Domain Analysis)	Difficult	Yes	Yes	Future Addition
8	Categorizing a Site	Difficult	No	Yes	Future Addition
9	Universal Design Analysis	Difficult	Yes	Yes	Future Addition
10	Augmentation Analysis	Difficult	Yes	Yes	Future Addition
11	Statistically Improbable Phrase (SIP)	Difficult	Yes	Yes	Future Addition
12	Style Analysis	Medium	Yes	Yes	Partially Complete
13	Navigation Analysis	Difficult	Yes	Yes	Future Addition
14	Content of Table Analysis	Difficult	Yes	Yes	Future Addition
N/A	Table of Contents	Medium	Yes	Yes	Complete
N/A	Link analysis	Easy	Yes	Yes	Partially Complete
N/A	Wire Frame	Medium	No	Yes	Complete
N/A	Frame Analysis	Easy	Yes	Yes	Complete
N/A	Meta-Analysis	Easy	Yes	Yes	Complete
N/A	Table Hierarchy Analysis	Easy	Yes	Yes	Complete
N/A	Forms Analysis	Medium	Yes	Yes	Complete

Table 1: Categorization of Different Analysis Techniques

3. Available Technologies for Creation of Page Monkey

Page Monkey could have been developed in two different ways: as a separate application or as a “plug-in” to the web browser. The separate application offers substantially more expressive power in the form of non-internet specific features. It does however pose an inconvenience to the user because they have to download the web-page or remember the URL of the page so that the system can download the content to analyze. On the other hand, building a “plug-in” to the browser allows the user to run Page Monkey while looking at the web page or surfing the internet. This provides optimal convenience to the user and the system. By being a “plug-in,” the system does not have to download the content of the web page and can get access to the Document Object Model (DOM) in real time. For these beneficial reasons, we have decided to implement Page Monkey as an add-on to the web browser. Several different browser technologies are available to implement Page Monkey in. The technologies we researched included Visual C++ extensions to Internet Explorer, Greasemonkey add-on for Firefox, HotJava browser, NetClue browser, and Java.

Visual C++ technology gives one the capability to make extensions the the Internet Explorer browser using C++ code. This technology gives one the capability to make a full featured extension of the Internet Explorer browser. By full featured we mean that it is not limited in its ability to perform tasks that a stand alone application can perform. Visual C++ gives the programmer the capability of DOM level 2 interactions with the browser. This means the programmer can read information from the Document Object Model as well as make dynamic additions to the document. Visual C++ extensions contain ample documentation through the MSDN libraries and have a decent amount of examples on the internet that one can study. On the other hand it can be very difficult to find any extra resources for this technology that are not provided by Microsoft. Visual C++ has a difficult learning curve and is specific to only the Internet Explorer browser on Microsoft Windows. This exclusiveness limits the capacity to allow a universal usage of our system. Also, the learning curve would make it difficult to get meaningful development to be done in a short period of time.

HotJava is a browser that is written entirely in Java. It has the ability to support java applets and JavaScript. The browser contains support for DOM level 1 interactions, which allows a programmer to only read the contents of the document. The only documentation available for this technology is located on Sun Microsystem’s website at <http://sunsite.nus.sg/hotjava/hotjava.html>. The browser supports all operating systems except MacOS. This feature is alluring but the system is still in beta release mode and has not been updated recently. The lack of a stable version and lack of DOM level 2 support has lead us to choose a different technology for our system. Also, the lack of MacOS support doesn’t allow the system to accommodate Apple computer users.

NetClue is a browser developed purely in java that is capable of running on multiple devices such as computers, cell phones, and other non-pc devices. It supports JavaScript and java applets. The browser also contains DOM level 1 support for programmers to access the contents of a document. The only documentation available for this technology is on the creator’s web site. There are no outside examples of how to interface with this browser. The lack of documentation and examples has lead us to look at other technologies to use for our system. The support for many different devices is impressive and the capability to support JavaScript will possibly lead to a future port of

the system for NetClue. Another reason we decided to go with a different technology is that not many web-surfers use this browser to explore a web page. More information on this browser can be found at <http://www.netcluesoft.com/top.php3?cat=base&sub=home>.

Another option available to us was to create a stand alone java application that uses the W3C DOM package. The java API contains information on how to use this package to get information from a web page. This technology would allow us to implement any analyses we could think of. There is lots of documentation and examples of the use of the API to analyze the contents of a web page. On the other hand, this technology is not integrated with the browser and must take input from the user on which page to analyze. This poses an inconvenience to the user since they can not just analyze a page from within the web browser. We recommend that future add-on system may be written in Java.

Greasemonkey is an add-on to the Mozilla Firefox browser that allows one to integrate scripts, like JavaScript, into the client-side browsers. There are a plethora of examples and tutorials for JavaScript that ease the learning curve of this simple language. Also, JavaScript can later be added to other browsers in a type of file called a bookmarklet, which allows one to activate a JavaScript program through the bookmark interface in the browser. By using JavaScript our system can be used in any operating system and on most popular browsers. The specific implementation of the Greasemonkey script was for ease of testing since there is a simple menu addition in the Firefox browser for our script. The Greasemonkey system also allows us to add a short-cut key-press combination that can activate our system. Greasemonkey/JavaScript supports DOM level 2, which gives the programmer the capability to inject items in to the document as well as read them [15]. The mix of quick learning and DOM level 2 support have led us to choose the Greasemonkey and JavaScript combination as the technology to implement Page Monkey [15].

4. System Architecture

Page Monkey is composed of three major components: the analysis engine, the DOM, and the user interface. The analysis engine contains all the different analyses that Page Monkey performs on a web page. The DOM is important to the system because it holds all the document information Page Monkey analyzes. Page Monkey is implemented in JavaScript that acts as a plug-in to Firefox through the Greasemonkey system. A plug-in is an addition to the browser that allows one to perform extra functionality. The user interface allows the user to select various types of analysis techniques and the results are displayed in a pop-up window. When the user selects an analysis to perform, the user interface queries the browser for the specific document information and then parses it accordingly to get the results. The results are then written to the results window. Figure 2 shows how the components described above interact with each other.

The system contains very few platform dependencies. The fact that Page Monkey is written in JavaScript theoretically allows it to be run on any browser that supports JavaScript functionality. There are many different browsers across different operating systems and devices that contain support for JavaScript. This allows Page Monkey to not be dependent on only one type of operating system or device. In principle, Page Monkey may be run on a hand held computer or a JavaScript enable browser within a cell phone.

The integration into the browser provides a major convenience for the user. The user can now analyze a web page without having to open a separate application and saving the web page for it to study. Page Monkey also uses the W3C's HTML conformance checking tool to determine if a page is compliant with the HTML 4.01 standard.

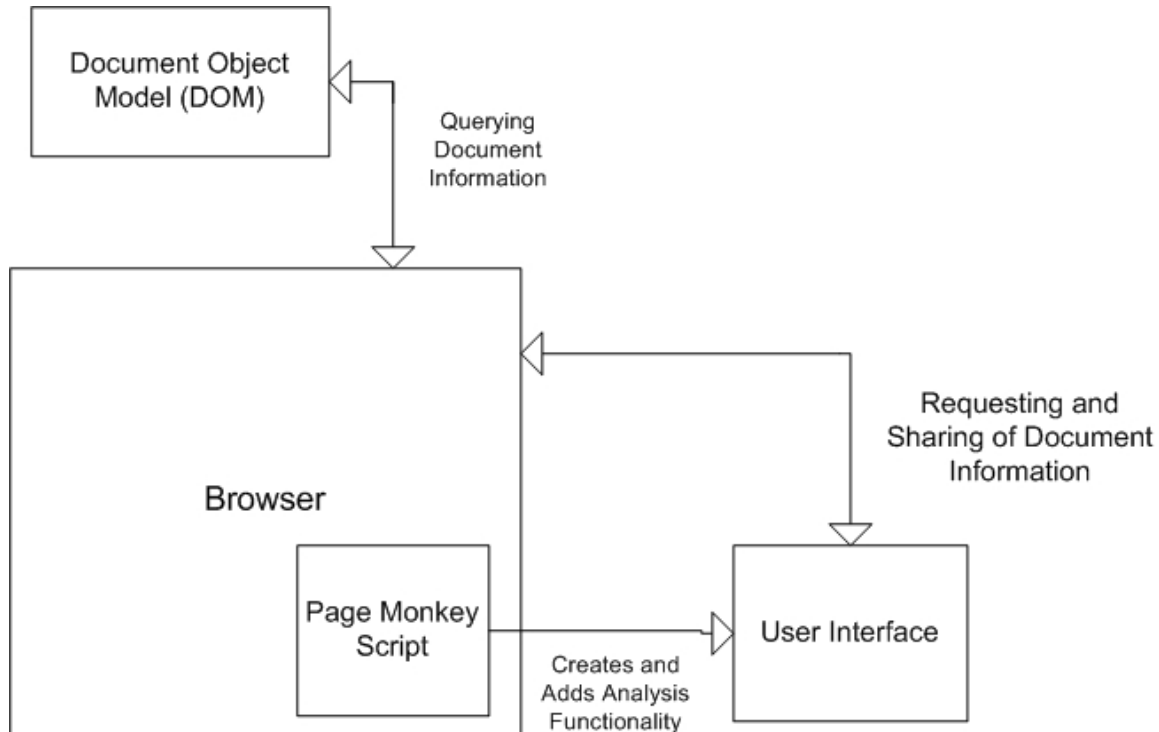


Figure 2: Page Monkey System Architecture

Figure 3 shows the sequence of events that occur during the usage of Page Monkey. First the user will issue a command to the system by choosing an analysis technique they wish to perform. Page Monkey then requests document information from the DOM API of the browser. The DOM API then returns the requested information and Page Monkey performs the analysis on the obtained document information. Page Monkey then displays the results of the analysis to the user and waits for additional commands.

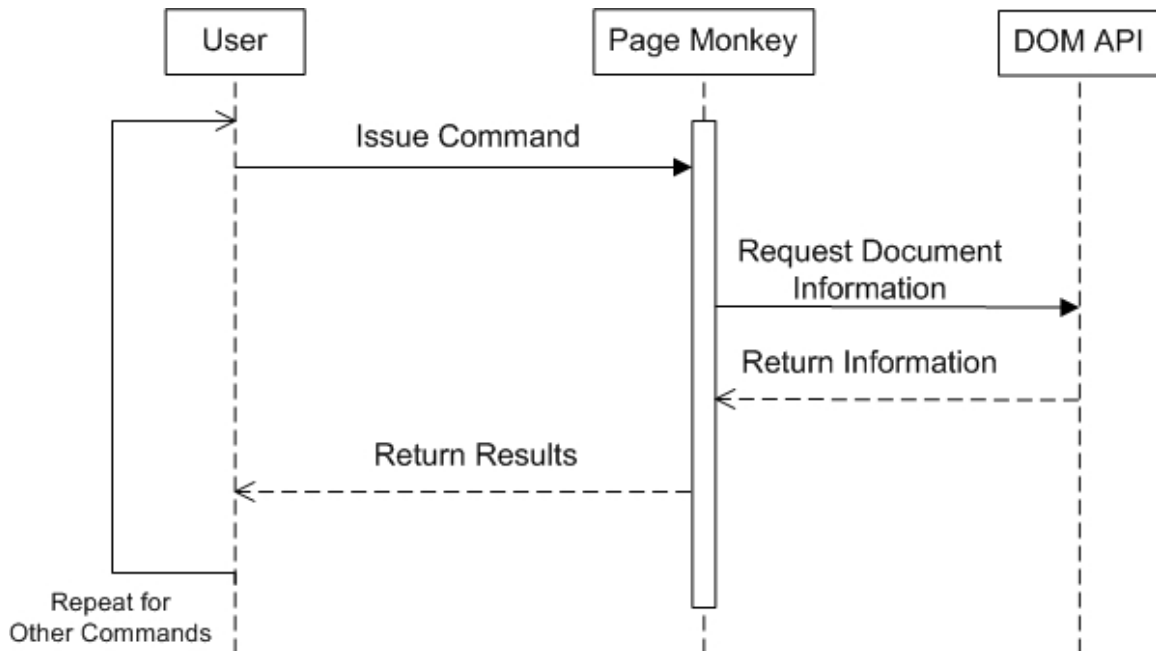


Figure 3: Page Monkey Sequence Diagram

The DOM contains different level specifications that bring about different capabilities to manipulate and interact with the document. Currently there are three different levels of the DOM. Level one provides a low-level set of fundamental interfaces that can represent any HTML document [14]. Level two provides extra enhancements not present in level one. Additions include the capability of scripts to dynamically access and update the content of a document, `getElementById` and `getElementsByTagName` methods have been added to provide easier access to the document, a generic event system has been added for scripts and programs, access to style sheet information for scripts and programs, and ability for programs and scripts to dynamically access and update to the content of HTML documents [14]. Level three adds even more options one can do to the document. Level three additions include the ability of a program or script to dynamically load and save document information. A validation component has been added to allow scripts to update content and still be valid by HTML specification standards. Also an extension of the event handler allows one to focus on keyboard input [14].

The JavaScript file follows the recommended style for writing Greasemonkey scripts. Recommendations on writing Greasemonkey scripts can be found at this web site: <http://greasemonkey.mozdev.org/authoring.html>. The script contains four distinct sections. The first section is the system initialization which adds the Page Monkey menu to the DOM of the web page and sets the appearance attributes for the menu. The second section includes the addition of the menu identifiers and the menu items. The third section contains the analysis engine of Page Monkey. Each analysis function's signature is named in a manner to correspond to which type of analysis it performs. Within each analysis function a script element is created to be appended on to the DOM of the web page. This script element contains the actual script that performs the analysis in a string representation so that it can be appended to the DOM. The final section of the script

contains the system activation code that tells Greasemonkey to initialize our system and begin accepting commands from the user.

The user interface is shown in Figure 4. To use the system, the user simply needs Greasemonkey installed and the Page Monkey script installed as well. The user can then just select an analysis technique to perform from the menu that appears above the web page. For instance, the user wishes to perform a Meta data analysis of the page they are currently viewing. In order to view the Meta data, the user selects the analysis menu and the different choices of analysis techniques appear. The user then proceeds to select the “Meta Data Analysis” menu entry. The system then does some work and displays all Meta data associated with the web page as shown in Figure 5. After this the user can close the results window when they are finished looking at the presented data.

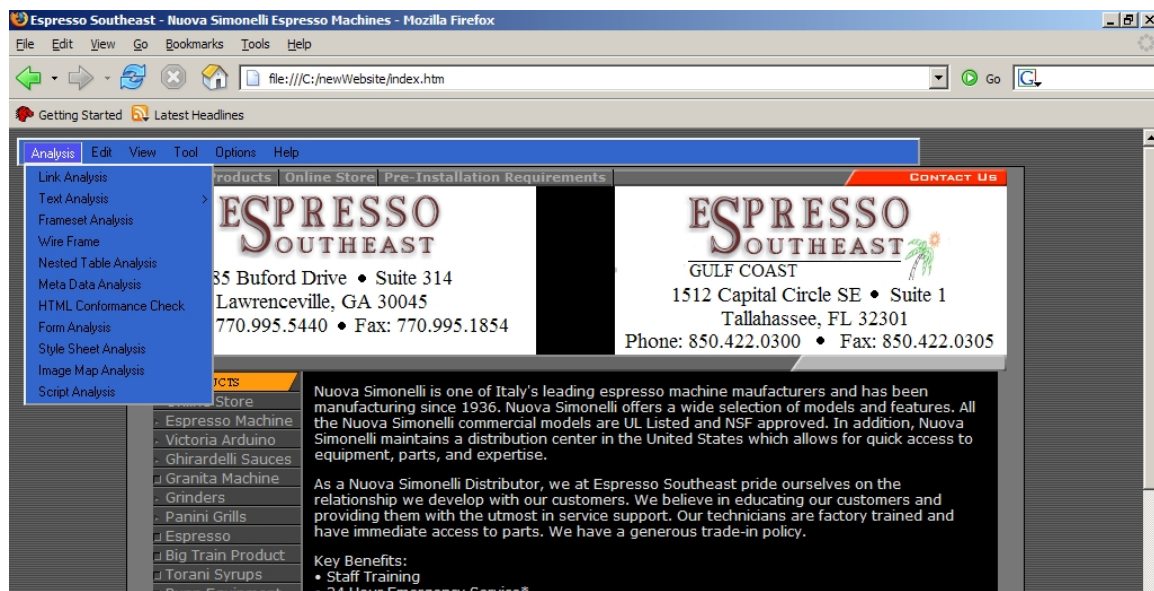


Figure 4: Picture of User Interface

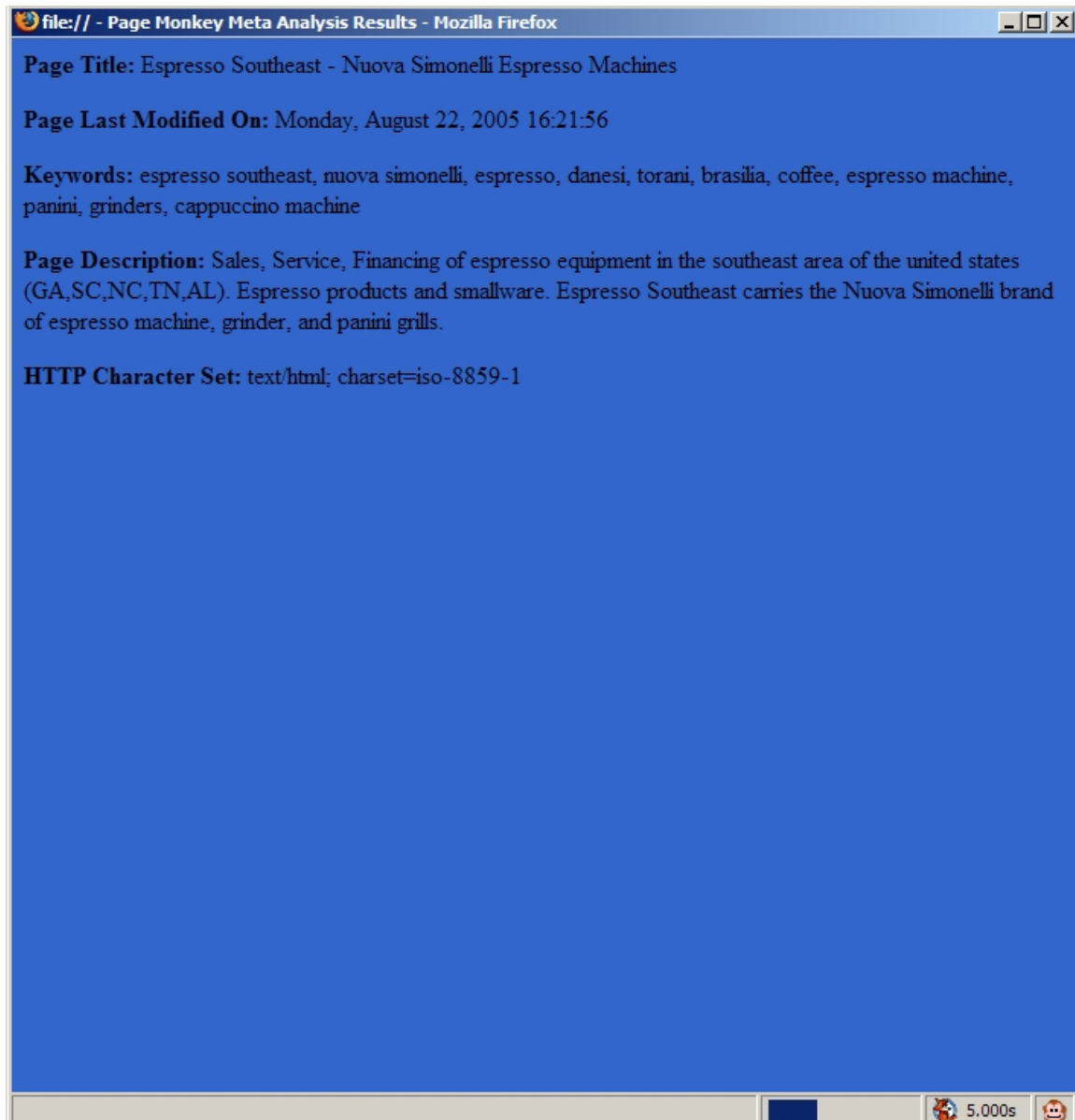


Figure 5: Picture of Analysis Result Screen

5. System Implementation Issues

JavaScript contains many limitations that make it difficult to provide a full feature system that is compatible across many platforms. JavaScript only has a very simple drawing interface for creating images. We have only found one library that for drawing simple figures such as lines, rectangles, circles, and sine curves [16]. This limited capability makes it difficult to express any type of analysis in a diagrammatically. One of the major drawbacks we have found in JavaScript is its inability to perform file input and output operations. This had hindered Page Monkey's capability to store data to files and load data from previous saved files. The lack of file I/O in JavaScript is because of the inherent security risk associated with allowing a script to access one's hard drive. This topic is mentioned as a future addition to the system. JavaScript also does not have the

ability to create network sockets for communication with other machines. This prevents us from making additions to the system that can transfer data over the network.

6. Informal User Evaluation

6.1 Tasks Users Must Perform

To perform an informal user evaluation, we tested our system on some students from the Georgia Institute of Technology. First, we gave the subject a brief introduction to the topic of our research. We then guided them through an example use of the system on Delta's home page. We let them play with the system only requiring that they perform a form analysis of Delta's home page. After they were done interacting with the system we asked them some questions.

6.1.1 Questions for Subjects of User Evaluation

- What aspects of our system are useful?
- What improvements can we make to our system?
- Are the results from the form analysis understandable and easier to understand than reading the web page source code?
- What are some improvements we can make to the results that are displayed?

6.2 Results of Informal User Evaluation

Overall the evaluators felt that Page Monkey presented the analysis results in an understandable way. All of them agreed that Page Monkey provides better feed back then looking at the source code of the web page. The evaluators provided us with many ideas for improving our system. They recommended that we make the results of an analysis be more interactive for the user. One evaluator mentioned that perhaps we could use a tree menu representing form components and the user could expand the menu to see detailed information about a form component. For link analysis, one evaluator recommended that we display the email addresses associated with a mail-to link and also to add the destination of the links for the user to look at. After performing the informal user evaluation, we feel that we are on the correct path to creating a system that can be useful for reverse engineering a web page. In the future a more intensive user evaluation will need to be performed.

7. Future Work

Future incarnations of Page Monkey will contain more functionality that the current version of the system. Future work will include the additions of the different analysis techniques outlined in this paper that are not currently completed. Currently the system only provides the lowest level abstraction of the data. The functionality for providing different layers of abstraction needs to be added to the system. Along with this addition, the system will need to have the capability to move up and down these different layers of abstraction when the user commands the system to change level. A help system needs to be added to Page Monkey in order to provide the user with useful information when they are having difficulties using the application.

JavaScript has the capability to handle mouse input and keyboard input from the user. The capability of the user to highlight a certain area of a web page and display data

about it needs to be added in a future version. Also, shortcut keys need to be added for each of the analysis techniques so that experts of the system can submit a query to Page Monkey without having to go through all the menus. Page Monkey needs to provide the capability for the user to capture any analysis results and allow them to easily copy it to a text editor or other external system for future reference.

Currently there are a few of the analysis techniques that are only partially complete. The analyses that need to be completed are image map analysis, link analysis, and style analysis. Additions to the image map analysis include displaying which image the map is intended for. An idea we had for a different presentation of the image map would be to section the areas on the image that correspond to the links in order to provide a visual presentation of the data to the user. Link Analysis needs to include the destination of each link including mail-to links in its results and which links may be broken. Furthermore, the presentation model may be changed to make the results of the link analysis more interactive for the user. Currently it just prints all information to the result screen. One possibility for an improvement was found during our informal user evaluation. The evaluator said that the system could represent the link information in a tree menu to allow the user to expand the results of any link information they wish to look at. Currently style analysis just looks at the style sheets that are in external files. It needs to be updated to handle the style tags in the head section of the web page. The style analysis also needs to determine if the author of the web page has separated structure from presentation on the page.

Once the system contains a greater portion of these additions, we need to perform an extensive user evaluation. One can perform such things by having HCI experts perform pre-defined tasks and take notes of what they find difficult. One can also question the HCI experts after they have completed the tasks to see what improvements can be made. Typical users should also be evaluated when testing the system. This way we can determine how the average user will interact with our tool. One should take notes to how difficult it is for the average user to perform these tasks. Along with testing the user interface, we should make sure our help documentation is understandable to the beginning user and that it is useful to an expert or intermediate user.

8. Conclusion

We believe we were successful in defining the meaning of reverse engineering a web page. Most systems we have found contain the reverse engineering of a specific aspect of a document. In retrospect, there is a large variety of information one can obtain from analyzing a web page. Our definition relates all the different types of information by incorporating the user experience in the results of the analysis. I believe we have created categories of information types that are useful to a user of our system. Future work may include refining the categories of analysis and perhaps adding some more. The field of reverse engineering of web pages has not received a lot of attention because of the lack of papers one can find on the subject. I believe we can pioneer some useful ideas in this field with the work that has been done.

During the process of designing analysis techniques and developing the system, I learned many different things about the internet domain. I have learned that web pages currently do not have a set standard of design to help facilitate what is a good web page. This aspect is mainly left up to the individual to decide if the web page can be considered

to have a good design. Many of the papers I have read insist that we should apply information principles and usability principles to the design of our web site. I believe the checking of these principles by our system could help establish the importance of these principles in the design of web pages. Web pages must be designed for the user and I believe that Page Monkey and the ideas set forth in this paper can give a developer guidance for user-centric web page design.

While developing the system I learned many things about JavaScript and the Greasemonkey system. As an intermediate level Java programmer, I found JavaScript fairly easy to learn in a short period of time. Many of the language features are similar to Java. When there were differences I was able to find sources online that can help clarify problems I ran in to. One tutorial site I found useful was created by the W3C and is at this URL: <http://www.w3schools.com/js/default.asp>. This tutorial contains almost all aspects of JavaScript including interactions with the DOM. Also, the `comp.lang.javascript` newsgroup is a good source for information when I was having trouble programming in JavaScript. Prior to this project, I have never heard of the Greasemonkey system. This add-on to Mozilla Firefox is a very interesting and useful tool when interacting with a web page. It allows a programmer to edit the document of a web page from their own browser by creating a personal script. This script can contain JavaScript in it to allow for advanced interactions between the programmer's script and the document. The fact that Greasemonkey is an add-on to the browser allowed us to develop our system in a light-weight manner since we did not have to deal with the interaction between the browser and our system. Greasemonkey handled all interactions between our script and the browser and allowed me to focus on writing only the JavaScript.

9. References

- [1] Betsy Beier, Misha W. Vaughan. The bull's-eye: a framework for web application user interface design guidelines, Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press, New York, NY, USA, 2003 pp. 489 - 496.
- [2] Beverly B. Zimmermann. Applying Tufte's principles of information design to creating effective Web sites, Proceedings of the 15th annual international conference on Computer documentation, ACM Press New York, NY, USA, 1997 pp. 309-317.
- [3] Charles M. Hymes , Gary M. Olson, Quick but not so dirty web design: applying empirical conceptual clustering techniques to organise hypertext content, Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques, p.159-162, August 18-20, 1997, Amsterdam, The Netherlands.
- [4] Cynthia M. Calongne, Designing for Web Site Usability, The Journal of Computing in Small Colleges, v.16 n.3, p.39-45, March 2001.
- [5] Fabrice Estievenart, Aurore Francois, Jean Henrard, Jean-Luc Hainaut. "A tool-supported method to extract data and schema from web sites," wse, vol. 00, no. , p. 3, 5th

2003.

- [6] Jean Vanderdonckt, Laurent Bouillon, and Nathalie Souchon. Flexible Reverse Engineering of Web Pages with VAQUISTA. Proceedings of the Eighth Working Conference on Reverse Engineering (WCRE'01). IEEE Computer Society, Washington, DC, USA. Eighth 2001 pp. 241.
- [7] Jose A. Borges , Israel Morales , Néstor J. Rodríguez, Guidelines for designing usable World Wide Web pages, Conference companion on Human factors in computing systems: common ground, p.277-278, April 13-18, 1996, Vancouver, British Columbia, Canada.
- [8] Laurent Bouillon, Quentin Limbourg, Jean Vanderdonckt, and Benjamin Michotte. Reverse Engineering of Web Pages based on Derivations and Transformations. 07 August 2005. 27 August 2005.
- [9] Melody Y. Ivory , Rashmi R. Sinha , Marti A. Hearst, Empirically validated web page design metrics, Proceedings of the SIGCHI conference on Human factors in computing systems, p.53-60, March 2001, Seattle, Washington, United States.
- [10] Nahum Gershon , Jacob Nielsen , Mary Czerwinski , Nick Ragouzis , David Siegel , Wayne Neale, Good Web design: essential ingredient!, CHI 98 conference summary on Human factors in computing systems, p.90-91, April 18-23, 1998, Los Angeles, California, United States.
- [11] Thomas S. Tullis. A method for evaluating Web page design concepts, CHI 98 conference summary on Human factors in computing systems, ACM Press, New York, NY, USA, 1998 pp. 323-324.
- [12] Carchiolo, V., Longheu, A., Malgeri, M. Improving Web usability by categorizing information. Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on 13-17 Oct. 2003 Page(s):146 – 152.
- [13] Flesch-Kincaid Readability Test. 16 November 2005. Wikipedia. 29 November 2005 <<http://en.wikipedia.org/wiki/Flesch-Kincaid>>.
- [14] What does each DOM Level bring? 10 July 2005. Mozilla.org. 29 November 2005 <<http://www.mozilla.org/docs/dom/reference/levels.html>>.
- [15] Testing the DOM API. 24 January 2003. Mozilla.org. 29 November 2005 <http://www.mozilla.org/docs/dom/domref/dom_intro.html>.
- [16] DHTML: Draw Line, Ellipse, Oval, Circle, Polyline, Polygon, Triangle with JavaScript. 24 October 2005. Walter Zorn. 29 November 2005 <http://www.walterzorn.com/jsgraphics/jsgraphics_e.htm>.

[17] HTML 4.01 Specification. 24 December 1999. World Wide Web Consortium. 29 November 2005 <<http://www.w3.org/TR/REC-html40/>>.

[18] Web Content Accessibility Guide. 23 November 2005. World Wide Web Consortium. 29 November 2005 <<http://www.w3.org/TR/WCAG20/>>.

[19] Gonzalez, Ben. Personal Interview. 6 December 2005.

10. Appendix A – Greasemonkey Information

Greasemonkey can be downloaded from this URL: <http://greasemonkey.mozdev.org/>. It requires the Mozilla Firefox browser version 1.0 or 1.5. The instructions for installation of this add-on are available on their web site. Once Greasemonkey is installed, a user can add our script by downloading it from our web site and then dragging the file into the browser. The user can also just click on the link to our script to load it into the browser. The user then needs to select install this script from the tools menu. The Page Monkey script will then be activated when ever the user is browsing a web page. To deactivate the script, the user can just click the little smiling monkey in the bottom right corner of the browser window. The monkey will turn grey and be frowning when the system is disabled. When using Greasemonkey all scripts must be named x.user.js where x is what ever name the author wants for the script.

11. Appendix B – Expert Interview Questions and Responses

Opening Questions:

[1] What training have you had on the design of web pages?

Answer:

Most of my training was self-taught. My Computer Science degree from Tech gave me the algorithmic thinking behind coding in general, but all my sepecific knowledge to Web coding and Web programming languages I learned through books and online Web site tutorials and APIs.

[2] What type of experience do you have in designing web pages? How long have you been designing web pages?

Answer:

I'm experienced in layout design, graphic creation, HCI implementation, database design and programing, HTML, XML, DHTML, JavaScript, JSP, PHP, CFML, ActionScript, Flash, Search Enginer Optimization (SEO), and designing & programming to meet accessbile guidelines.

I began designing Web pages as a bored Co-op student during one of my initial Co-op rotations. I've been working with Web design and development for over 6 years now.

[3] What are your principles of good design for web pages?

Answer:

Usability first and foremost. Good navigation is key. A site is useless if you can't find what you're looking for. Then you must have a striking design. Some sites that are just

data stores are fine without anything striking, but people come, find what they need, and leave. Any site that updates that doesn't necessarily have information the public "needs" must have a great visual design to draw people back. It's the "coolness" factor that brings return visitors.

[4] What are your principles of bad design for web pages?

Answer:

Poor navigation. Non-uniform navigation across different pages. Non-uniform look/feel/branding across pages in the same domain. It's very disorienting to go from one page to another in the same site, same domain, and have page look different. Having navigation change drastically from page to page adds to the confusion and makes navigating the site non-linearly very difficult.

Look at these sites and please answer the following questions.

Sites:

[1] www.cnn.com

[2] www.espn.com

[3] www.newegg.com

[4] www.cc.gatech.edu

Questions for [1] www.cnn.com:

[1] What HTML concepts do you see when looking at this page?

Answer:

Standard "News site" layout. A lot of information sectioned into headings to make finding things a bit easier.

[2] When visiting these pages which portions do you like and which do you dislike and why?

Answer:

LIKES:

The top story headline links immediately available. Quick and easy navigation. The stories with headers farther down the page are easy to check quickly and let the user decide if they want to click through for more detail.

DISLIKES:

3rd party advertising takes up screen real estate that would be better served for user content.

[3] What presentation models (such as frames, tables, iframes) do you think the page uses when first looking at them and not checking the page source?

Answer:

Tables

[4] Is the information density of the web page too high or too low? What reasons have lead you to conclude this?

Answer:

It's neither. This is a news site; a global news site at that. It's expected to have a lot of content. Visitors coming here know that so it's not as overwhelming as it would be if you were to find this much content on the homepage of a company's Web site.

[5] Does the site's navigation components (links, etc) provide the user with clear means to navigate to other pages? Please state why it is or is not difficult to navigate the page.

Answer:

Yes. The links match appropriate headings one would find in a news paper. This is a good example of remediation of the paper newspaper.

Questions for [2] www.espn.com:

[1] What HTML concepts do you see when looking at this page?

Answer:

Again, standard "News site" layout. A lot of information sectioned into headings to make finding things a bit easier.

[2] When visiting these pages which portions do you like and which do you dislike and why?

Answer:

LIKES:

All the current sports are listed as links right at the top. And direct links to Scores are available one-click from the homepage, no need to click on "Football" first, THEN click on "Scores" to see the football scores.

DISLIKES:

There's not much I don't like about ESPN.com. They do a good job with allowing 3rd party advertising without having it interfere with prime content areas.

[3] What presentation models (such as frames, tables, iframes) do you think the page uses when first looking at them and not checking the page source?

Answer:

Tables

[4] Is the information density of the web page too high or too low? What reasons have lead you to conclude this?

Answer:

It's neither. This is a news site as well; sports news, but still news. It's expected to have a lot of content. Visitors coming here know that so it's not as overwhelming as it would be if you were to find this much content on the homepage of a company's Web site.

[5] Does the site's navigation components (links, etc) provide the user with clear means to navigate to other pages? Please state why it is or is not difficult to navigate the page.

Answer:

Yes. ESPN is a sports site. Their navigation is geared towards sectioning out each sport but also providing general highlight stories across any sport.

Questions for [3] www.newegg.com:

[1] What HTML concepts do you see when looking at this page?

Answer:

Large shopping site. They offer a lot of products and have to display them all.

[2] When visiting these pages which portions do you like and which do you dislike and why?

Answer:

LIKES:

Search available immediately, a must for large shopping sites. The product categories are there for people who would rather browse than search for something specific. The dynamic expanding sub menus is a great touch as well. Let's me see what's there without having to commit to a click first. Member login at the top of the page is good. Members should not have to sift through the site to login no matter what page they're on. I also like the Stocking Stuffers link with the winter backdrop. Shows the site is maintained on a frequent basis (unless you still see this link in April).

DISLIKES:

There's a lot going on down the right-most column. Little-to-no space between any of the content squares. Makes it very difficult to follow and just adds unnecessary clutter to the screen.

[3] What presentation models (such as frames, tables, iframes) do you think the page uses when first looking at them and not checking the page source?

Answer:

Tables

[4] Is the information density of the web page too high or too low? What reasons have lead you to conclude this?

Answer:

The right-most column makes the page too dense for me.

[5] Does the site's navigation components (links, etc) provide the user with clear means to navigate to other pages? Please state why it is or is not difficult to navigate the page.

Answer:

Yes. Navigating to various categories is easy. Searching is well placed. For the most-part, main navigation does not change across pages.

...And no. The right-most column changes on each page. Sometimes it disappears completely. A Quicklinks section shows up on some pages and not on others.

Questions for [4] www.cc.gatech.edu:

[1] What HTML concepts do you see when looking at this page?

Answer:

Basic informative site about a college/school/academic program.

[2] When visiting these pages which portions do you like and which do you dislike and why?

Answer:

LIKES:

Links and submenus are apparent immediately (although this could be achieved with dynamic mouse-overs just as well). Top news stories and event are prominent and top FAQ questions regarding applying to the college are readily available one-click from the homepage.

DISLIKES:

Georgia Tech began putting forth an effort to add uniform branding and navigation across all it's departments' Web sites. CoC has yet to catch on. Linking to the CoC off Tech's homepage is a bit disappointing to go from a nice, clean, visually stunning look to something so bland. Also, Security Center and Webcam images seem like wasted space. They'd be better served as small callout icons off to the side and allow top, content realstate to house more important news.

[3] What presentation models (such as frames, tables, iframes) do you think the page uses when first looking at them and not checking the page source?

Answer:

Tables

[4] Is the information density of the web page too high or too low? What reasons have lead you to conclude this?

Answer:

It's neither. They do a good job of using white space to separate content areas and keep the content small to not overwhelm visitors. However, this also serves to make their page look very bare without any striking visual elements to accompany the layout.

[5] Does the site's navigation components (links, etc) provide the user with clear means to navigate to other pages? Please state why it is or is not difficult to navigate the page.

Answer:

The homepage links provide a quick glance at almost every option available to link to. However, once you get on subsequent pages, backwards navigation becomes difficult. A breadcrumb link trail would work well on these pages. Also, navigation changes from page to page. Some pages have submenus positioned to the right, pages like External Relations have the submenu links embedded in the text. It lacks uniform navigation throughout the site which make non-linear navigation difficult.