The Portrait of a Common HTML Web Page

Ryan Levering
SUNY at Binghamton
ryan.levering@binghamton.edu

Michal Cutler
SUNY at Binghamton
cutler@binghamton.edu

ABSTRACT

Web pages are not purely text, nor are they solely HTML. This paper surveys HTML web pages; not only on textual content, but with an emphasis on higher order visual features and supplementary technology. Using a crawler with an in-house developed rendering engine, data on a pseudo-random sample of web pages is collected. First, several basic attributes are collected to verify the collection process and confirm certain assumptions on web page text. Next, we take a look at the distribution of different types of page content (text, images, plug-in objects, and forms) in terms of rendered visual area. Those different types of content are broken down into a detailed view of the ways in which the content is used. This includes a look at the prevalence and usage of scripts and styles. We conclude that more complex page elements play a significant and underestimated role in the visually attractive, media rich, and highly interactive web pages that are currently being added to the World Wide Web.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia; H.3.1 [**Information Storage and Retrieval**] Content Analysis and Indexing; I.7.0 [**Document and Text Processing**]: General

General Terms

Measurement

Keywords

HTML, visual, feature, script, JavaScript, style, CSS, survey, World Wide Web

1. INTRODUCTION

As new and more sophisticated commercial Web based applications are being developed, the usefulness of analyzing the rendered web page in addition to its HTML text and tags is becoming increasingly apparent. Using layout features of rendered pages facilitates the automatic building of wrappers that can interact successfully with the web page (see [21]), helps in finding semantically important content (see [19]), and enables mobile adaptation of web pages in order to reduce the document to small screens in a semantically coherent way (see [3]).

DocEng'06, October 10–13, 2006, Amsterdam, The Netherlands.

This progress may indicate a trend in the way in which HTML is analyzed. However, this would not be an easy transition, because the field of text analysis is more documented and non-textual analysis (including image and layout analysis) is a higher dimensional, less defined problem. Furthermore, most subfields in information retrieval seem to largely ignore the issue without seeing much impact on their results.

That being said, next to intuitive researcher impressions on the web, there does not appear to have been a survey on the underlying approach of all of this separate research. We set about building a system that would survey the Web to be able to measure the makeup of an average page, specifically focusing on its non-textual content and visual layout. While these average page statistics might not be useful on a per-feature level, they would probably help identify trends in document technology to guide future research.

Most surveys of web pages in the past have focused on text, link analysis, and rates of change on the World Wide Web. In [7], Fetterly et al. look at textual content changes on HTML documents over time, ignoring markup altogether. In [8], Gibson et al. survey templates occurring in a DOM representation, both in terms of their position on the page and their presence on the web over time. The position of the template is measured one-dimensionally against the textual length of the page. Finally, in [16], Ntoulas et al. examine document features primarily to find spam identifying text-based features that fall out of the normal distribution of a Web document.

A modern web page is made up of several client-side technologies. The glue of it all is HTML [10], which is what the World Wide Web was based on and what most Web users are familiar with. It is a markup language that was initially created to provide semantic text annotation, but migrated in common usage to be more of a visual layout language.

Cascading style sheets (CSS) [4] were created by the W3C in an attempt to separate the visual representation from the underlying data representation of HTML or other markup languages. Style sheets can dramatically alter the presentation in a client browser, affecting visibility, impact, and position.

Scripting, of which JavaScript [6] is the most widely used, has the ability to perform computation on the client side to provide a richer document viewer experience. Browsers provide access to both interface controls of the browser and the underlying document structure. This enables scripts in the extreme case to dramatically alter the content, experience, and display of the page.

On top of these core technologies are several higher level media types that have become commonplace. Java Applets [11], ActiveX controls [1], and Macromedia Flash [14] objects are all pre-compiled, embedded, possibly interactive pieces of software that run within a set area of a web page.

The goal of this paper is to come to some understanding of how all of those technologies fit together to create the end visual document. This paper will first provide a description of the framework for data collection and analysis. Then, it will provide several sections of results that detail page composition along several dimensions. Finally, it will make some broad conclusions and research directions that the data may indicate. We hope that the information gathered in this survey on how the various technologies included in HTML pages are commonly used will be helpful to designers of challenging web-specific solutions.

2. BUILDING THE FRAME

2.1 Pseudo-Random Crawler

The data in this survey was collected with a developed crawler, with the goal being to gain a pseudo-random sample of HTML web pages. We examined several sampling methods, as there is no established way of sampling the web without crawling the entire space [2]. The random crawl method is the naïve option, but this suffers from locality issues dependent on where the crawl starts. We chose to use random seed pages from search engine indexes and then use a limited random crawl from that point. This decision was an experimental one, and we make no effort to mathematically defend the sampling methods.

We used several methods of seed generation via search engines, outlined in Table 1. The first was the Yahoo! random page CGI [20], which redirects to a random URL from their index. Like most indexes, the Yahoo! index contains a higher percentage of well-connected, popular documents. To offset this, we then did a breadth-first search of twenty-five documents from every seed document. On average, this tended to glean more internal, non-popular documents which appeared to improve the coverage.

The second method was to use the Open Directory Project (ODP) database [17] to randomly select seed documents and then used a six document random depth-first search (or random crawl). This method was based loosely on the random walk model, variants of which are found in several places, including [15]. This sample was also unique in that it was more multi-lingual than the other methods.

For the final method, we used two random English words as a query to the Google search engine using their own API [9] and then use the resulting best ten documents as seeds for the crawl. We followed with a breadth first search, as well.

Table 1. Pseudo-random Samples

#	Randomized Source	Crawl Strategy	Docs
1	Yahoo! Random	Breadth (25)	9,000
2	ODP Random	Depth (6)	10,000
3	Google Random Query	Breadth (25)	2,500

The three methods produced slightly different samples of web documents. The main difference is in the length of the collected documents. Method 3, the Google query method, produces rarer, text-heavy documents because of the random nature of the query and the higher likelihood that a longer document has an obscure query term. The first method covers the top level of websites fairly well and therefore has shorter, richer documents. The second method (our preferred one) splits the difference and we believe does the best job of capturing both types of documents. Because of the variety in crawling strategies, no claim is made that the datasets are an accurate representation of the utilized search indices.

We compared the word count distributions with [16] where Ntoulas at al. use a much larger sample. Our Yahoo! and ODP sampling methods had similar quasi-normal distribution (Yahoo! being lower and ODP being higher), whereas the Google sampling method had a much larger mean. The survey will focus mainly on the ODP collection method, giving the other only as a data reference for several of the higher-level graphs.

As seen in Table 1, our sample sizes are fairly small as web surveys go. This is mainly because the processing on each document requires a prohibitively long time to acquire a sample of a greater magnitude. However, the analysis done in this survey was primarily statistical aggregate functions which tended to converge fairly quickly within several hundred documents.

None of these methods produces a statistically true random population of the entire Web. In particular, they are biased toward English documents (even ODP, which is multi-lingual) and popular pages. However, while this survey was an attempt to give an unbiased document sampling, we believe that a skew toward popular documents is acceptable. A more rigorous sampling methodology is left for future work.

2.2 Document Processing

Being as the goal was to get some idea of all aspects of the document, relatively complex processing was done on each document. The document was transformed from raw HTML source to rendered output in a series of stages, using a in-house developed rendering engine, called WebSeer. This engine, while not as accurate as current mainstream browsers, has some flexibility in its ability to analyze intermediate stages of HTML rendering and the effect that different technologies have on the final product.

A total of five models were used in the rendering and analysis: the original HTML input stream, a tree-like Document Object Model (DOM) representation, a script-interpreted DOM representation, a generic visual block model, and a rendered image model. Measured features are drawn from a particular model, combination of models, or from one of the script/style interpreters.

Documents that used frames for layout (4% of total documents) were not considered. The documents were rendered to a 1000 pixel width, which is close to a common viewing size. This actually became a very important choice when examining the area projections of the separate content types.

3. PAINTING THE PORTRAIT

In the following sections, we outline the results of analysis on the gathered datasets. In general, the results consist of feature mean values and frequencies of occurrence. With all of the statistics, we provide a confidence interval (CI) for a 95% confidence level[1].

[1] This assumes that the features have a normal distribution in the population, which is not entirely true; many of the sample distributions appear to be skewed log-normal. However, even in cases where these figures are not statistically accurate, they still give a general idea as to the distribution of the feature.

3.1 Basic Attributes

In order to provide some verification of our data set gathering process and compare against previous work, we collected basic statistics of naïve HTML features. The HTML document sample that we processed contained on average 281 (CI=14) HTML tags. They also contained on average 41 (CI=6) links of which 10 (CI=6) were outside of the domain, 2.7 (CI=.24) were to other subdomains, and 29 (CI=1) were the same domain or document. The extra-domain linkage data, as shown in the wide confidence interval for that feature, was very polarized as expected [12]. Hub type pages had clearly high numbers of external links whereas most documents had zero or one. As this paper is very document-centric and not about link analysis, that data will not be detailed.

Tables are used fairly extensively as layout devices. Only about 15.1% (CI=0.7%) of documents did not use a table at all. Those tended to be media driven, with a higher percentage of image/object content. Of the documents that did use tables, they averaged maximum table depths of 2.95 (CI=.04), which indicates that nesting of tables to achieve the desired layout is a norm. At the high end, a percentage of pages had depths of up to 8 tables, even after removing outliers. We suspect these outliers were the result of WYSIWG editors and/or templates.

Textually, the average document contains 474 (CI=27) words, after removing one very spammed outlier. As mentioned, our distribution of word count frequencies correlated with previous work on web page word frequencies done in [16]; only the mean was different between the different data sets. Most documents contain less than 300 words with a smaller number of very large documents biasing the mean.

3.2 Relative Content Areas

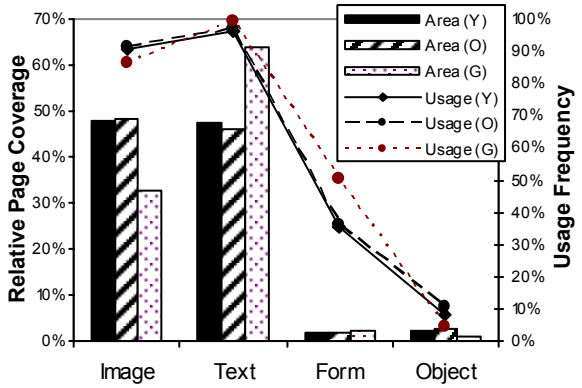

Figure 1. Content Usage and Coverage Average

In this study, rendered relative areas of four major content types in a web page (images, text, form, and media objects) were compared. To collect these numbers, the area of every individual element of the particular type was summed, disregarding any borders, margins, or padding assigned by HTML/CSS. Then this number was normalized against the rendered document content area. Because of this, the coverage for all of the elements of a particular document is 100%, which means the averages of the sample sum to that as well.

The axis on the left side of Figure 1 measures content area coverage, while that on the right of the figure shows frequency of use. The results there show that in our ODP and Yahoo! data sets, images and text occupy similar amounts of space on an average web page. A small number of documents (4% (CI=0.4%)) contained no text at all. In review, these documents tended either be redirection pages or just a large spliced image that was used as a starting point for a website.

The Google dataset contained a higher percentage of text, though similar usage. However, it shows the same trends as the other datasets, as in the following figure, which is the same as Figure 1 only for "above the fold" content area - area that is within the part of the page that is viewable when first loaded. We used the top 800 pixels as our page height, to correspond with the 1000 pixel width we rendered.

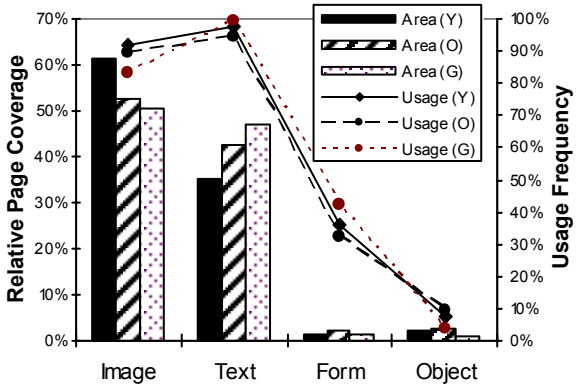

Figure 2. "Above the Fold" Usage and Coverage

The average page height for our main dataset was 1440 (CI=76) pixels, or almost twice the size of the screen height. Figure 2 demonstrates that when the average page is opened, more of what a viewer is seeing in terms of pixel area is actually image, not text. One reason is that the area of spaces between lines and paragraphs is not included in our computation of text area. Another reason for this trend is the text y-axis distribution which is examined more in depth in a section below.

3.3 Document Content Analysis

To get more detailed information on the usage of the four identified content types, we broke down their usage across the dataset and within each specific document. The first graph for each content type in the subsections below indicates a percentage coverage histogram with each bin on the x-axis holding the number of documents that have that percentage of a particular content type. In this way, it gives a better view of the usage of a particular content type across the whole dataset.

The other graphs are average projections of the content area onto a specific axis. One dimension for each document was divided into 10 ranges that contained the sum area of a specific content type within that range. This allows us to examine how often a particular content type is used in a certain relative space within the document.

The areas that were calculated in this study are actual content area, not perceived content area. Therefore, a form that typically has fields with labels will contain both text area and form area, even though to a viewer, it appears to be "a form". This also means that blank space, which is usually important to the meaning of the

document and may change the viewer's perception of the content, is not taken into account in any of the area computation.

3.3.1 Image Profile

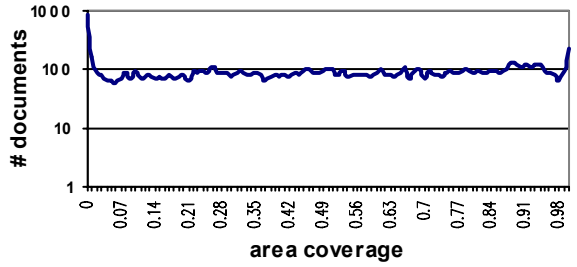

Figure 3. Image Coverage Frequencies

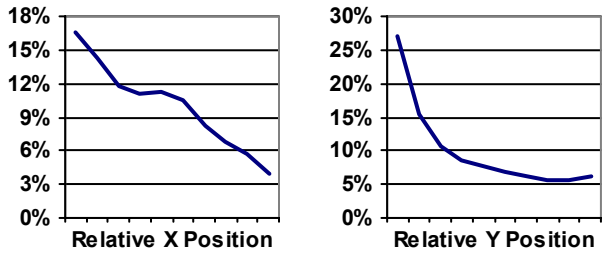

Figure 4. Image area by dimension

As shown in Figure 3, images are used at all different area coverage levels, from the entire contents of the screen to small isolated graphics. The average image area we found was 49,144 (CI=5,097) pixels, which if square would make it roughly 221x221, which is a fair sized web graphic.

Images, as portrayed in the figures above, have become an important content type on the average web page. While this may be skewed by the use of images as rendered forms of text, it does not change the way that search engines currently view the page contents. This may indicate that image analysis research will be increasingly important in the future of web analysis. This includes OCR technologies to recognize text from images, though it is possible this practice will fade as CSS control over text display becomes more commonplace.

Table 2. Image Usage Frequencies

Image Encoding	Average	Frequency
GIF	17.5 (CI=.7)	77.9% (CI=0.8%)
JPEG	3.2 (CI=.1)	55.8% (CI=1.0%)
PNG	0.36 (CI=.06)	7.2% (CI=0.5%)
BMP	0.05 (CI=.02)	0.8% (CI=0.2%)

The most frequently encountered image encoding was GIF, used on most web pages and used often within those pages. GIF is a file format that uses color frequency encoding, so is most frequently used for images with a smaller palette, i.e. non-photographic. Next to that was JPEG, which was found on over half, but did not have as high an average usage. This is most likely because of the role that JPEG, as a more photograph-specific encoding, plays on web pages. PNG files, which have the advantage of being able to encode using multiple schemes, were not as common. Bitmap images were even rarer. It seems clear that GIF and JPEG files have found their niches on web pages.

3.3.2 Text Profile

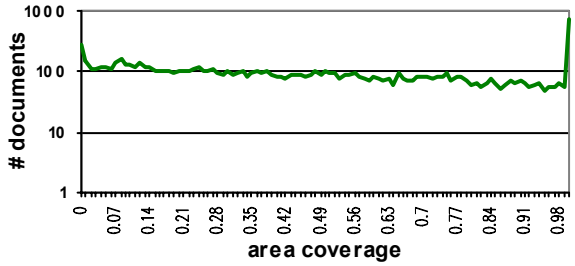

Figure 5: Text Coverage Frequencies

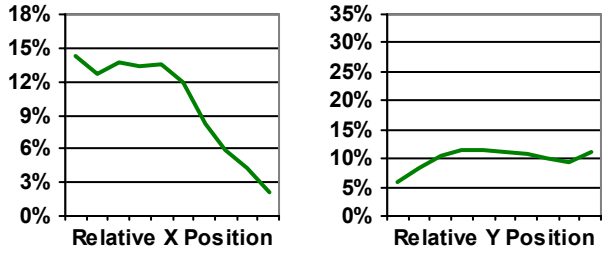

Figure 6. Text area by dimension

The text area frequency graph is very similar to the image graph, with an even distribution of text coverage. The only major difference is that there are many documents with completely text, which complements the case of many documents without images.

The text dimensional projection is interesting in that it differs greatly from the other content types. In particular, the text content tends to be more skewed towards the left side of the screen than the other content types. It is also more evenly distributed along the vertical dimensions of the page than the other content types which tend to occur at the top of the page. Even more interesting, text is not very frequently found at the very top of the document area, or at the far left. This corroborates the commonly used motif of the document header, which is often not text based.

3.3.3 Form Element Profile

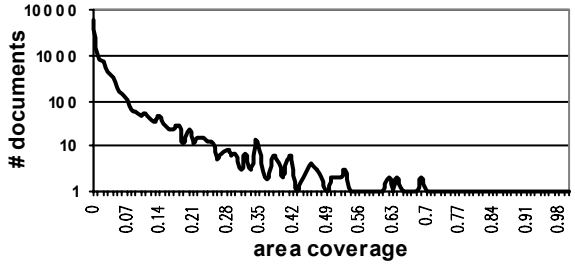

Figure 7. Form Element Coverage Frequencies

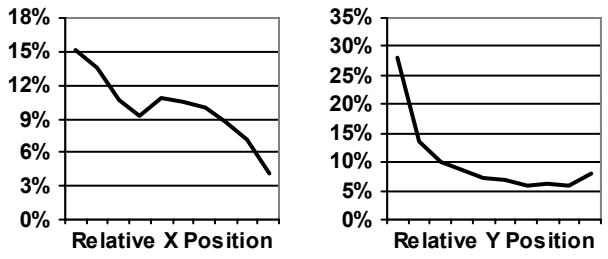

Figure 8. Form area by dimension

If scripting is used for client-side interaction, forms are used for server-side interaction, which is often biased toward data submission. They offer a means for user inputted data to be transferred to a program on the server, where a task can be performed or navigation can be dynamically chosen.

The ability to navigate a form and interact with it is of great concern in the creation of automatic meta-search engines, which aggregate the results from several search engines. Forms also are the key to the access of the deep web – information contained in databases that are only accessible via dynamically generated form responses.

First of all, we looked at where the information filled in forms is typically processed (submitted). Whether a form submits to a local link, an external link, or its own HTTP address may say something about the purpose of the form. Surprisingly, we found that it was almost as likely to find a form on the 39.9% (CI=0.9%) of pages that contain forms that submit to an external domain (19.4% (CI=0.8%)) as to the same domain (25.3% (CI=0.8%)). This could be due to external services for ad clicks or search engine redirection.

Next, we examined the visual makeup of the form. HTML defines several standard interaction objects for forms: check boxes, radio buttons, drop-down lists, text areas, text fields, and buttons. The average form is small, typically only having 1 to 2 visual elements. The popularity of several smaller form types (like search boxes) could explain this average.

Table 3. Form Element Usage Frequencies

Form Element Type	Frequency
Text Field	34.4% (CI=0.9%)
Button	23.1% (CI=0.8%)
Combo Boxes	13.1% (CI=0.7%)
Text Areas	4.2% (CI=0.4%)
Check Boxes	3.1% (CI=0.3%)

The most popular form element is the text field, which is used for most text entry within a form. Next, the form button, be it for submitting the form or resetting the fields, was found on almost as many pages. Our analysis engine separated form buttons from images that had a similar role, which represents the disparity in needing a button to submit the remaining form text fields. The only other element of note was the combo box, which has the only other frequency over ten percent. The other elements (radio buttons, check boxes, text areas) were less common.

3.3.4 Plug-in Object Profile

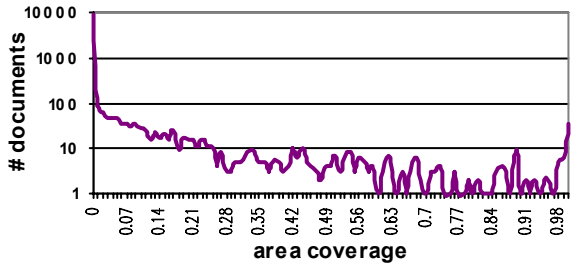

Figure 9. Plug-in Coverage Frequencies

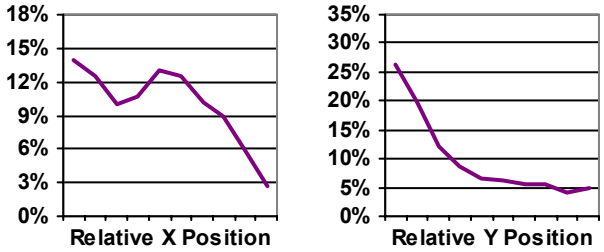

Figure 10. Plug-in area by dimension

As HTML was not originally intended for media-rich applications, several supplementary technologies have developed to provide users with a richer interface within the familiarity of a web browser. We chose to highlight two technologies in our survey which are the most popular.

The Flash technology, which is currently managed by Adobe, is a popular format that combines vector graphics with animation and interaction. Java Applets allow use of the full-featured Java programming language, and are most often used for more complex client applications. Additional plug-in types include ActiveX, which is a proprietary Microsoft browser extension technology and several types of popular non-interactive audio/video embedding.

Of the plug-in objects that were found within HTML documents, the large majority were Flash content. Java applets were found in a smaller percentage that about equaled the remaining non-classified objects.

Table 4. Plug-in Type Frequency

Plug-in Type	Frequency
Flash Shockwave	7.6% (CI=0.5%)
Other (often ActiveX)	1.3% (CI=0.2%)
Java Applet	1.2% (CI=0.2%)

In general, these higher level objects have not been studied in great depth (see [5] for an idea on Flash). However, while this study did not go into great depth in plug-in analysis, we did note that the area profiles tend to be closest to those of images, indicating that objects are often used in place of images, possibly to make use of smaller file sized, animated graphics that several of these technologies make possible.

3.4 Content-Supporting Technologies

3.4.1 Style Profile

This part of the study was designed to measure the prevalence of Cascading Style Sheet usage to modify the layout of HTML and to try to characterize how the styles are affecting the presentation. Styles consist of visual properties that are associated with HTML tags by a type of pattern matching. These visual properties can be stored in external files, in a tag within the HTML document, or directly within the specific tag to which the style should be applied.

For the purposes of determining the effect that a particular style will have on the HTML presentation, several categories were created that capture some of the basic tendencies of particular sets of styles:

1) Text – affects the low level text representation, including font, decoration, boldness, and color
2) Space – determines how the "empty" space on the screen looks, allowing margins and borders with colors/styles, backgrounds
3) Layout – changes the layout algorithm for an element, be it the traditional alignment (left, center, right) or more complex floating/absolute positioning
4) Sound – W3C has a number of tags defined for aural output, though browsers do not support them

The statistics collected measure "applied styles" – the styles that are actually applied to HTML tags within the document. This means that the possibility exists for s*t tags, where s is the number of styles defined and t is the number of HTML tags, given that every style matches universally. This measurement has the benefit of ignoring styles in style sheets that are never applied to the page contents. No user style sheets (client-side styles) are considered in the analysis.

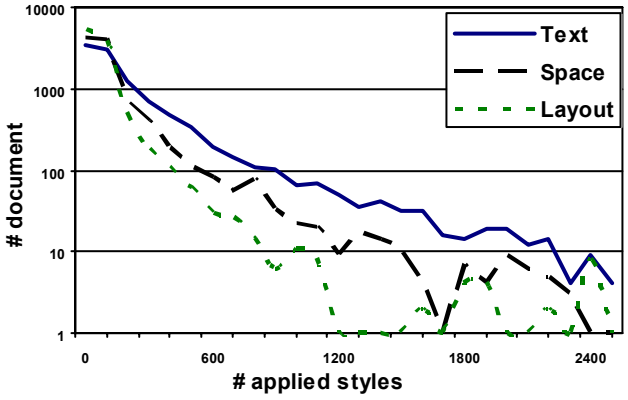

Figure 11. Number of Styles by Type

Layout styles show a fairly high frequency of minor usage but then drop off rapidly. We theorize this is because most layout styles are applied on a higher level to document structure. Text styles on the other hand, have a much greater range of frequencies, used abundantly in some cases and sparsely in others. This is most likely because of the fine-grained nature of the text tags that match more universal font styles. Sound styles were verified to have practically no presence in our data set and were not pictured in the figure.

With regards to the source of applied styles, 51.7% (CI=1.0%) of documents use inline styles, 46.5% (CI=1.0%) use external style sheets, and 27.3% (CI=0.9%) use internal style sheets. These percentages may be misleading. Although inline styles are used in approximately half of the documents, they actually account for only a small percentage of the styles applied to a particular tag.

3.4.2 Script Profile

Scripting on web pages can be very powerful and has the potential to modify any aspect of the DOM model, including modifying presentation in combination with styles. In particular, scripts can modify the textual contents of the document before it is displayed to the user, effectively changing the meaning of the document.

The scripts in the documents were interpreted using the Rhino interpreter from Netscape [18] and a developed DOM adapter and listener model to determine the effects the script had on the user interface. The DOM model was written to mimic a Netscape family browser, with the appropriate API methods and properties that the browser family makes available. It was not fully implemented, due to time constraints, so some unorthodox scripts could potentially be a source of miscalculation in the analysis. However, the large majority of common functionality was implemented.

The first test was to see how prominently scripts featured in the initial interpretation of the document. Scripts are very common on an average web page. 67% (CI=0.9%) of our sample contained JavaScript. The average page interprets 31.1 (CI=1.5) script statements and 15.1 (CI=.7) function declarations during its load. It is believed that most of the statements are variable assignments that are later used for script/form events.

Most important to us was the ability of a script to alter the page contents, so we attempted to see how often content was modified by an average script. Our data showed that a significant percentage of the documents used some method of scripted document writing to modify contents before initial display. The writing was often dynamically created advertising, but was also used in some cases to define a large part of the page structure.

Several other more common script usages are detailed in Table 1. The time based scripts are time-looped functions that are executed to poll UI, animate graphics, pull data, etc. The results of these functions were not included in analysis.

Table 5. Loading Script Actions

Script Action	Frequency
Document writing	30.5% (CI=0.9%)
Start Timer-based Script	4.9% (CI=0.4%)
Window Open (Popup)	2.1% (CI=0.3%)
Automatic Redirect	1.9% (CI=0.3%)

To get an idea of the level of interaction on a web page, the unique mouse/keyboard events were triggered and their effects recorded. The most common event was to open a popup window, which is more sympathetic to the original goal of the popup window. The automatic redirects are usually browser navigations based on form contents or dynamic script choice. It would be interesting to include these redirects in the link analysis of the document, which is not currently done. Many of the alerts were most likely from

validations that failed when the forms were left empty by WebSeer.

Table 6. Interactive Script Actions

Script Action	Frequency
Window Open	10.9% (CI=0.6%)
Automatic Redirect	6.3% (CI=0.5%)
Start Timer-based Script	3.2% (CI=0.3%)
Alert (Dialog Box)	2.8% (CI=0.3%)

We also measured the usage of AJAX – a script technology that opens an asynchronous channel to a server to transfer data after the document has already been loaded. This allows the document to be more fluid in it's interactions with the user. However, it was not frequent on the average web page, neither on page load nor on user event. Our entire primary sample only contained twenty instances where it was utilized. It seems to still be limited to large, professionally maintained sites and applications.

4. CONCLUSION

While the initial Web pages were primarily text based HTML, as shown in our survey, this is not currently the case. Styles and scripts are powerful technologies that can significantly modify the rendered page. These modifications can be positive, making the page more visually attractive, its contents more current, and more personalized to address users interests. At the same time these technologies may be used in a negative manner, for example by spammers. Forms enable e-commerce, different search tasks, and access to the deep web. They differ in their layout and usage of different element types. Finally images and other media such as animation make web pages more attractive, user friendly and meaningful. This survey shows the prevalence of images on a level with text, the commonality of scripts and styles, the variability of the interactions used in forms, and hints at the various roles of plug-ins.

The impacts of these findings affect most areas of analysis-oriented information goals on the web. They stress the role of image analysis for better page content understanding. For classification tasks, especially ones like genre classification that are multi-dimensional and heavily related to user perception, we believe using higher-order features will be necessary for higher levels of accuracy. Also, we note that HTML document content is dynamic and using a fully interpreted version of the text output would be necessary to avoid inaccuracies in certain documents. Finally, we believe that a stronger user-centric view of documents, through use of higher-level document features, will have the necessary flexibility for the future, dynamic nature of the web.

Our future work will concentrate on increasing our sampling size and methodology in an attempt at having a better representation of the World Wide Web. Also, we plan on broadening our analysis and data gathering capabilities, to include: frame-based documents, plug-in and image analysis, and use-based analysis of scripting technologies. In addition, we intend to apply these higher-order features to research areas mentioned above to practically demonstrate their merit.

5. REFERENCES

[1] ActiveX Technology. http://msdn.microsoft.com/library/default.asp?url=/workshop/components/activex/intro.asp

[2] Bharat, K. and Broder, A. A technique for measuring the relative size and overlap of public Web search engines. *Comput. Netw. ISDN Syst.* 30, 1-7 (Apr. 1998), 379-388.

[3] Chen, J., Zhou, B., Shi, J., Zhang, H.-J. and Qiu, F. Function-Based Object Model Towards Website Adaptation, in *the proceedings of the 10th World Wide Web conference (WWW10)*, Budapest, Hungary, May 2001.

[4] CSS Specification. http://www.w3.org/TR/CSS21/.

[5] Ding, D., Yang, J., Li, Q., Wang, L. Wenyin, L. Towards a Flash Search Engine Based on Expressive Semantics, in *the proceedings of the 13th World Wide Web Conference (WWW2004)*, New York, New York, May 2004.

[6] ECMAScript Specification. http://www.ecma-international.org/publications/files/ecma-st/ECMA-262.pdf.

[7] Fetterly, D., Manasse, M., Najork, M., Wiener, J. A large-scale study of the evolution of web pages, in *the proceedings of the 12th World Wide Web Conference (WWW2003)*, Budapest, Hungary, May 2003.

[8] Gibson, D., Punera, K., Tomkins, A. The Volume and Evolution of Web Page Templates, in *the proceedings of the 14th World Wide Web Conference (WWW2005)*, Chiba, Japan, May 2005.

[9] Google Search API. http://www.google.com/apis/.

[10] HTML Specification. http://www.w3.org/TR/html4/.

[11] Java Applets. http://java.sun.com/applets/.

[12] Kleinberg, J. M. Authoritative sources in a hyperlinked environment. *J. ACM* 46, 5 (Sep. 1999).

[13] Kovacevic, M., Diligenti, M., Gori, M. and Milutinovic, V. Recognition of Common Areas in a Web Page Using Visual Information: a possible application in a page classification, in *the proceedings of 2002 IEEE International Conference on Data Mining (ICDM'02)*, Maebashi City, Japan, December, 2002.

[14] Adobe Flash. http://www.adobe.com/products/flashplayer/.

[15] Henzinger, M., Heydon, A., Mitzenmacher, M., Najork, M. Measuring Index Quality Using Random Walks on the Web, in *the proceedings of the 8th World Wide Web conference (WWW8)*, Toronto, Canada, May 1999.

[16] Ntoulas, A., Najork, M, Manasses, M., Fetterly, D. Detecting Spam Web Pages through Content Analysis, in *the proceedings of the 15th World Wide Web Conference (WWW2006)*, Edinburgh, Scotland, May 2006.

[17] Open Directory Project database. http://rdf.dmoz.org/.

[18] Rhino JavaScript interpreter. http://www.mozilla.org/rhino/

[19] Song, R., Liu, H., Wen, J., Ma, W. Learning Block Importance Models for Web Pages, in *the proceedings of the 13th World Wide Web Conference (WWW2004)*, New York, New York, May 2004.

[20] Yahoo! random CGI. http://random.yahoo.com/bin/ryl.

[21] Zhao, H., Meng, W., Wu, Z., Raghavan, V., Yu, C. Fully Automatic Wrapper Generation for Search Engines, in *the proceedings of the 14th World Wide Web Conference (WWW2005)*, Chiba, Japan, May 2005.