Website Mapping

Edward Miller
Software Research, Inc.
901 Minnesota Street
San Francisco, CA 94107 USA

Email comments to miller@soft.com
See also the companion White Papers:
The WebSite Quality Challenge.
WebSite Testing.
WebSite Loading.

Introduction
Many properties of important WebSites are only detectable when they are seen by the user from a browser. For example, on a WebSite with dynamically generated pages which are produced from several different servers and final-assembled just before delivery to the client browser, some parts might actually be missing or unavailable. The server may not see this, but the end user -- looking at the generated page from a browser -- certainly sees this as a problem.

Also, how long a page takes to download -- a factor that is affected by server speed, the speed of the web, and the speed of the connection from the web to the client browser -- might check out OK from the servers' point of view, but might result in a too-slow page delivered to the user, who then clicks away to another site.

Theses are just two of many reasons that point to the need to have detailed analysis of pages -- or of collections of pages, i.e. whole WebSites -- analyzed from the perspective of a user, i.e. from a browser.

Why Analyze From A Browser?
The main reason for trying to look at WebSites from a browser is that this is the only accurate way to analyze what users actually see. If the WebSite is composed of static pages, i.e. pages that have fixed HTML content and don't vary over time, then server side analyses can be fairly effective. There are very good utilities, e.g. WebXREF, that serve very well in these cases.

But this option disappears in most of the sites where what you see are dynamically generated pages produced by a variety of means based on requests from a user interactively with the WebSite. These account for most of the important WebSites where eCommerce is a main focus. Such WebSites are where simple timing and availability errors have serious consequences.

But the main reason for trying to look at WebSites systematically entirely from the browser is accuracy. A browser-based analysis really does deal with the users' perspective. *What you see is what you get!* If an analysis from a browser turns up a problem there's no ambiguity: it really is a problem!

What Should We Search For?
If you can do searches from a browser [see below], then the natural question to ask is, what do you want to know?

Here are some browser-side questions that having easy and quick and accurate answers to could be very important:

- Do the pages on the site download and render quickly enough?
- Are there broken images on the pages as they are rendered?
- Does the page contain links that go to unavailable resources?
- Are there obvious typos or misspellings on the page?
- Are there JavaScript mistakes on the page?
- How old are the pages?

Note that this list does not include "HTML Errors" because in most cases the browsers overcome most such errors by simply ignoring incorrect HTML. HTML correction is a different topic entirely and, it may be important to emphasize, a high fraction of perfectly OK WebSite pages score very low on the "HTML Perfection" scale!

Site Analysis Requirements
An automated Site Analysis engine has to meet some basic components:

- A way of deciding where a search is to being, how it is to run, and when to stop it.
- A way to record (for completeness checking) which pages really were visited after the first page.
- Some way to decide what to do with pages are they are selected.
- A method for reporting what's found when tests made of pages show some problem.

This kind if a search engine based within a browser is actually a kind of spider program because it would start at some point and then create and internal worklist based on what it has just found as it recursively descends through a series of links. But for the goals of automatic WebSite analysis you really don't want the search to drive all over the web. You really want the search focused like this:

- From a specified starting WebSite pages,
- For all pages that are linked to that page (below that page),
- And continuing until some limit is hit [see below].

In other words, you want your analyses to be constrained and controlled searches of a WebSite or a sub-Website, where you can easily control the search criteria. You don't want your search to run forever, but you do want the search to be over a large enough span of pages so that the results of doing the analysis are interesting and valuable.

Site Analysis Engine Architecture
Here are the general requirements on the Site Analysis Engine that support the above goals:

- *Search Size Specification.* From a user specified starting page, i.e. where the browser is currently positioned. To all links on that page out to a specified depth of search. With and without visiting pages that are not part of the current WebSite under test. Less than a specified total maximum number of pages to be visited. Less han a specified maximum amount of search and/or download time.

- *Search Type Specification.* You should be able to specify both the types of links to search, and the types of protocols to search. Link type specifications should be by type of file, e.g. *.HTML or *.phtml or *.jpg, or by type of protocol, e.g. HTTP or HTTPS or FPT.

- *Search Inclusions/Exclusions.* You should be able to provide a list of specific URL to exclude during a search (e.g. you might do this to prevent going to a page that logs your analysis session out after having logged in. You should also be able to indicate URLs that you wish to add to the search tables if they happen to be encountered (e.g. other sub-Websites).

- *Search Modes.* You should have the option of seeing everything happen in the current browser (this will require it to have focus), or you should have the option to run in the background. (There may be some performance savings possible if less than full analysis is done in background mode analyses.)
- *Cache Control.* With no cache at all, with cache cleared before starting the search, or with cache fully operational.

Reporting
Certain standard reports are always generated by the site analysis process.

SiteMap Report
This SiteMap report is the record of the pages visited that also shows the way in which the search engine came to visit the page. This report is generated as a result of the process of doing the search. Even if there are not filters running the SiteMap report is an important record of what ULRs were and were not reached.

Two kinds of reports that are of particular interest:

- **Full Report.** This reports shows every page (by its URL) and for that page the set of pages that depend on it (i.e. have links on it). There would be no need to show all the details of each page, but for completeness all of the pages below a page would need to be shown.
- **Irredundant Report.** This reports shows every page (by its URL) and for that page shows only the pages that were actually visited. You could expect that this would be a much smaller list, particularly for a WebSite that has a lot of connectivity.

Filter Reports
The outputs of the filters need to be very simple, easy to read, and generated in real time, as the search is going on (This gives the user the information as soon as possible and prevents generating reports that contain too much information.) Given the ability of the site analysis process to systematically visit every page in a WebSite -- subject to the limits imposed by the user -- it is pretty easy to imagine the kinds of reports it would be interesting to generate from page by page analysis as each page is presented.

- **Pages Loading Slower Than Report.** This uses the browser timing capability measure the actual download time of the page. Downloading time can be expected to be correlated with page size, but not necessarily. The best use of the data is to include in the final list only those pages that are slower than a threshold, and then to sort these pages by URL in reverse time order.

- **Pages Larger Than Report.** If a page is too big it is a candidate for revision. Page size can be expected to be correlated with downloading time, but not necessarily. The report should show pages total bytecount only if the size exceeds a user-specified thresholds. The reported pages should be sorted in decreasing size order.

- **Pages Older Than Report.** If pages are old, they may represent out of date information. For pages that are dynamically generated, of course, the page age would be essentially zero. But older pages on a site may be a signal that something important may be wrong.

- **Broken or Unavailable pages Report.** From the browser's point of view a page could be broken or unavailable for a wide range of reasons. It could be, of course, actually missing (a type 404 error). Or it could be from a server that is temporarily unavailable or cannot be reached right now. All pages that the search shows as unavailable should be marked with an indication of what caused the failure.

- **Off-site Pages Report.** Many websites reference a lot of off-website pages, and it may be useful to know which links are to offsite pages. This filter lists them in the order of discovery.

- **Pages Patching Search Criteria Report.** This may be the most powerful kind of page by page analysis to have available. Every page would be searched, at varying levels of detail, for a match (or a non-match) on a specified string or strings. The report would show the pages that match (or don't match) the search criteria in the order in which the pages are visited.

One possible application of this features is as a security check mechanism. To confirm non-manipulation of pages the server could place a particular code on every page in a hidden field. The site analysis engine could search for pages that do NOT have this code -- thus revealing any pages that are not "original".

The level of detail of the scanning of individual needs to be controlled. At one end of the spectrum you might want to look at everything in the page, including JavaScript and HTML and hidden fields -- everying. On the other hand you may be concerned only about what the user sees on the page -- the visible text. Another option might be to search only in the META text fields on the page.