Vol. 11 No. 4, July 2006

The influence that JavaScript™ has on the visibility of a Website to search engines - a pilot study

M. Weideman and F. Schwenke,
e-Innovation Academy
Cape Peninsula University of Technology
Cape Town, South Africa

change font

Abstract

Introduction. In this research project, an empirical pilot study on the relationship between JavaScript™ usage and Website visibility was carried out. The main purpose was to establish whether JavaScript™-based hyperlinks attract or repel crawlers, resulting in an increase or decrease in Website visibility.
Method. A literature survey has established that there appears to be contradiction amongst claims by various authors as to whether or not crawlers can parse or interpret JavaScript™. The chosen methodology involved the creation of a Website that contains different kinds of links to other pages, where actual data files were stored. Search engine crawler visits to the page pointed to by the different kinds of links were monitored and recorded.
Analysis. This experiment took into account the fact that JavaScript™ can be embedded within the HTML of a Web page or referenced as an external '.js' file. It also considered different ways of specifying links within JavaScript™.
Results. The results obtained indicated that text links provide the highest level of opportunity for crawlers to discover and index non-homepages. In general, crawlers did not follow Javascript™-based links to Web pages blindly.
Conclusions. Most crawlers evade Javascript™ links, implying that Web pages using forms of this technology, for example in pop-up/pull-down menus, could be jeopardising their chances of achieving high search engine rankings. Certain Javascript™ links were not followed at all, which has serious implications for designers of e-Commerce Websites.

Introduction

The purpose of this research project was to investigate the influence of JavaScript™ on the visibility of Web pages to search engines. Since JavaScript™ can be added to Web pages in different ways, it is necessary to consider all these different ways and investigate how they differ in their influence on the visibility (if at all). The authors will discuss the structure of JavaScript™ content, and also how the Website author can prevent search-engine crawlers (also called robots, bots or spiders) from evaluating JavaScript™ content.

The literature suggests that JavaScript™ can be implemented to abuse search engines - a claim that will be investigated in this research project. Abuse in information retrieval is not a new topic - much has been written about ethical issues surrounding content distribution to the digital consumer (e.g., Weideman 2004a). Most search engines have set policies specifying on what basis some Web pages might be excluded from being indexed, but some do not adhere to their own policies (Mbikiwa 2006:101). Although graphical content is intrinsically invisible to crawlers, many search engines (including the major players like Google, Lycos and AltaVista) are offering specific image searches (Hock 2002).

In a pilot study the authors attempted to determine to what extent search engine crawlers are able to read or interpret pages that contain JavaScript™. The results of this research can later be used as a starting point for further work on this subject.

Literature Survey

Introduction

A number of authors have discussed and performed research on the factors affecting Website visibility. One of them, Ngindana Ngindana 2006: 45, claims that link popularity, keyword density, keyword placement and keyword prominence are important. This author also warns against the use of frames, excessive graphics, Flash home pages and lengthy JavaScript™ routines (Ngindana 2006: 46). Chambers proposes a model which lists a number of elements with positive and negative effects respectively on Website visibility: meta-tags, anchor text, link popularity, domain names (positive effect) as opposed to Flash, link spamming, frames and banner advertising (negative effect) (Chambers 2006:128). In this literature survey, the focus will be on the use and omission of JavaScript™ only.

JavaScript™ background

Netscape developed JavaScript™ in 1995 (Netscape 1998). Originally its purpose was to create a means of accessing Java Applets (which are more functionally complex than is possible in JavaScript™) and to communicate to Web servers. However, it quickly established itself as a means of enhancing the user experience on Web pages over and above achieving the original purpose.

The original intent was to swap images on a user-generated mouse event. Currently JavaScript™ usage varies from timers, through complex validation of forms, to communication to both Java Applets and Web servers. JavaScript™ has been established as a full-fledged scripting language (which in some ways is related to Java) with a large number of built-in functions and abilities (Champeon 2001).

JavaScript™ inclusion on Web pages

There are two ways to include JavaScript™ on a Web page. First, a separate .js file that contains the script can be used, or the script can be embedded in the HTML code of the Web page (Baartse 2001). Both have definite advantages and disadvantages (see Table 1). The disadvantages of a separate file are the advantages of an embedded script and vice versa.

Defining JavaScript™ as a separate file	
Advantages	**Disadvantages**
Ease of maintenance - script is isolated from unrelated HTML.	**No back references** - scripts have difficulty in referring back to HTML components.
Hidden from foreign browsers - the script is automatically hidden from browsers that do not support JavaScript™.	**Additional processing** - the interpreter loads all functions defined in the HTML header - including those in external files. It may mean that unnecessary functions are loaded.
Library support - general predefined functions can be put in external scripts and referenced later.	**Additional server access** - when the Web page is loading, it must also load the JavaScript™ file before it can be interpreted.

Table 1: Advantages and Disadvantages of defining JavaScript™ as a separate file.

Generally JavaScript™ will be programmed using a hybrid environment with both embedded and external scripts. More general functions will be placed in a separate JavaScript™ file while some of the script specifically for a page will be embedded in the HTML (Shiran & Shiran. 1998).

JavaScript™ can also be included in different places on a Web page; the only mandatory requirement is that the script must be declared before it is used. It can either be declared in the <head> tag of the HTML or it can be embedded in the <body> tag, at the point where it is needed. JavaScript™ is currently still under development. Developers who wish to provide scripts to the public must take this into account when developing Web pages and, therefore, a technique has been developed that allows developers to test the environment and execute different scripts for different environments (Shiran & Shiran 1998).

It is important to note that JavaScript™ can be added as the 'href' attributes of anchor tags. This implies that instead of using a normal 'http:' URL, a 'javascript:' URL is specified in the tag. This can have consequences for Website visibility through crawlers which investigate page content to find linked content (Shiran & Shiran 1998).

JavaScript™ links

Three basic types of JavaScript™ links have been identified. This experiment will be testing all three kinds of links:

- *document.write() links*. In this case, JavaScript™ is used to simply add text to the HTML document at load time.
- *<a href="javascript:..."> links*. Here the HTML contains an <a href> tag that calls some JavaScript™ function. This function is then responsible for opening the document and can be used to open the document in a separate window if so desired.
- *JavaScript™ menus*. These menus involve the most 'dynamic' of all JavaScript™ links. The HTML never contains any links to the documents and never has any reference to the links at all. The URLs to the documents are all embedded inside the JavaScript™ code. The script will display a menu from which the user can exercise his/her choice and open a document.

Crawler handling of JavaScript™

According to Goetsch (2003), search-engine crawlers cannot interpret JavaScript™ and, therefore, cannot interpret the content it refers to. However, use of the correct design can still make a Web page 'visible' to search-engine crawlers. Venkata (2003), Slocombe (2003a) and Brooks (2003) all agree that crawlers cannot access JavaScript™ and thus ignore it completely. They suggest that Website designers follow some rules to make their pages more visible to crawlers. Brooks (2003 & 2004) specifically states that Google will avoid parsing scripts in general and JavaScript™ in particular.

The design rules for making Web pages containing JavaScript™ visible to crawlers are:

- the Website designer should ensure that links are added between the pages in addition to the JavaScript™. This implies that the Website author does not rely on the JavaScript™ alone to provide links throughout the site;
- a sitemap with normal text links should be included as part of the Website;

- navigational links should be included as 'footers' on each page; and
- normal text links in a <noscript> tags should be included. However, this practice could trigger the spam alarm, which in turn could result in search engines removing the Website from their index completely (Gerensky-Greene 2004).

However, some authors disagree with the claim that JavaScript™ is not being interpreted by search-engine crawlers. According to one forum contributor (Chris 2003), there is no reason why a crawler cannot search through JavaScript™ code for links to other pages. The crawler does a string search for certain patterns and does not try to interpret the HTML in any way. Assuming that this is true for all crawlers, they should have no trouble in extracting hyperlinks from JavaScript™ content. It is interesting to note that this specific author is not conclusive about this claim, but only suggests that it is possible.

Authors from Google (2004) support the idea that Web page designers should be careful when using JavaScript™. They are, however, not conclusive and warn that: 'then search engine spiders may have trouble crawling your site'. It is clear that there is not unanimous agreement on these issues - this fact was the major motivation for this research project.

Crawler exclusion of JavaScript™

According to Koster (2004) there are two ways to prevent search-engine crawlers from visiting Web pages. One is to use a 'robots.txt' file, placed in the root folder of the Website. This file contains exclusion rules specifically aimed at visiting crawlers. These programs will read this file, parse the rules and visit only those pages that are not excluded by any of the rules.

The second way to exclude content from being seen by crawlers is to use the 'robots' meta-tag in the header section. This tag has the general form:

```
<meta name="robots" content="noindex">
```

The 'content' parameter can have the following three values:

- 'noindex' - do not index this page.
- 'nofollow' - do not harvest this page for links to other pages.
- 'noindex, nofollow' - both of the above.

When the 'robots.txt' file and/or the <meta> tag is used for exclusion, the Website author effectively instructs the crawler that this part of the Web page must not be indexed for search results.

It appears as if it is not a feasible solution to limit the bandwidth used by crawlers when the page in question should still be listed on the search results. If bandwidth is a problem, the Website author should rather contact the search engine directly with information regarding the problem (Thurow 2003).

JavaScript™ and crawler abuse

Crawler abuse refers to unscrupulous Web page authors attempting to convince crawlers that the content of a given Web page is different from what it really is. This can be done in a number of different ways.

The <noscript> tag can be used to include keywords that have no relevance to the content of the site. This could result in the crawler indexing the site on keywords that are never seen by any human user. Authors agree that this is an unacceptable way to promote a Website and that text in 'hidden' fields (such as the <noscript> tag), should not be adding anything but relevant information for users.

Sites with adult content are regular abusers of JavaScript™. These sites use JavaScript™ to open new windows with new content, which are sometimes hidden behind the main window, so the user does not even see them. This caused crawler developers to view the crawling of JavaScript™ links with care.

Secondly, the <noscript> tag can be populated with links to other pages on the same site.The acceptable way to use this would be to only include links to pages that are actually referenced by the JavaScript™ code. However, one can easily abuse this function by adding a number of unrelated links to the <noscript> tag.This will cause the crawler to browse though pages that are never really referenced by your page.

Another way that JavaScript™ can be used to abuse crawlers is the use of so-called 'doorway' domains. These domains are registered with content, but are seldom seen by the user, because it uses JavaScript™ to automatically forward the user to another domain. Doorway pages and a number of other practices (cloaking, spamming, etc.) are regarded as unethical (Weideman 2004a).

Web page visibility and JavaScript™

It has been claimed that over 80% of Web traffic is generated by information searches, initiated by users (Nielsen 2004). Web page visibility, if implemented correctly, will ensure that a given Web page is listed in the index of a search engine, and that it will rank well on the user screen when certain keywords are typed by a user. Weidman (2004c) established a set of relationships between Internet users, Websites and other elements, which could contribute to Website visibility. This study then proved that one of these elements, metadata, was underutilised by most Website authors. Another claim in this regard is that single-keyword searching on the Internet has a lower success rate than when a user employs more than one keyword (Weideman and Strumpfer 2004). The use of relevant keywords on a Website can make it more visible and affect the success of single- or multiple-word Internet searches.

From the discussion above, it can be determined that JavaScript™ has an influence on a Web page's visibility to search engines. In many cases search-engine crawlers will completely ignore JavaScript™ content. In other cases the crawlers may be selective about the JavaScript™ links they are willing to include while crawling. This is possible since it is easy to abuse crawlers while using JavaScript™ inclusions.

Futterman (2001) claims that the easiest way to ensure that JavaScript™ pages are search-engine friendly is to provide duplicate HTML code contained within the <noscript> tags. An added advantage is that this page will now also be accessible to older, non-JavaScript™ browsers. Slocombe (2003b) suggests that text alternatives be provided for the same purpose.

It appears as if there are many ways to solve the abuse problem and still achieve good search-engine ratings, even while using JavaScript™. The important thing is that the JavaScript™ must be used properly and that the alternative methods of achieving crawler visibility should be used in conjunction with JavaScript™. Alternatives should also be used properly and not in an abusive way.

Weideman and Kritzinger (2003) claim that meta tags as a Website visibility enhancer in general do not seem to be used much by Website authors. Few search engines seem to recognize them either. Thurow (2004) proposes that the proper use of JavaScript™ will not negatively affect a site's search-engine rating. Abusive use will also not only affect it negatively, but may even lead to complete banning from certain search engines.

The experimental Study

Purpose of the study

The purpose of the pilot study has been identified during the literature survey as being the determination of the ability of the crawlers to find, interpret and follow links contained within JavaScript™ on a Web page. It is clear from the survey that there is some confusion about the ability of search-engine crawlers to interpret links contained in JavaScript™.

This experiment was justified, since the effect that JavaScript™ will have on a Web page's ranking within a search engine, will be directly influenced by the crawler's ability to interpret links contained within JavaScript™.

Research questions

The study attempted to answer as many as possible of the following questions (Anonymous2 2004):

- Do search engine crawlers execute JavaScript™? If so, to what extent?
- Can crawlers recognize valid links in JavaScript™ code, specifically in 'document.write()' statements?
- Do crawlers handle relative and absolute links within JavaScript™ in the same manner?
- Do crawlers handle embedded JavaScript™ and external JavaScript™ in the same way?
- Does a Web page require a <a href> tag to the JavaScript™ in order to trigger the crawling?

Methods

Steps taken during the execution of the experiment

The experiment was executed by creating a test Website, registered with a number of search engines, and monitored regularly for crawler visits.

- A Website with as much as possible content was created for the experiment (see figure 1).
- The Website contained some normal text links, where no JavaScript™ was involved. Furthermore JavaScript™-type links of various formats and styles were created, in order to answer as many as possible of the questions above.
- Once the site was created, a sample of search engines was selected, and the site was manually registered on each of them. Only the home page of the site was registered in this way, to ensure that crawler visits to the non-home pages of the Website can only be triggered by a visit to the homepage.
- At this point, the search engines were given time to send their crawlers to the site. One month was initially allowed. At the end of this period, the logs from the Website were gathered and all the chosen search engines were also monitored to see which of the pages the different crawlers had accessed. The results were then compiled and presented (see Table 2).

Both the Website logs (obtained from the Internet service provider) as well as the information from the different search engines would be collected.

Selection of the search engines to use

Sullivan's lists of popular search engines and other engines were used to make the initial selection of search engines on which to register the site. From this list, all engines that require a paid registration were eliminated. This was done since payment for having Web pages listed (i.e., paid inclusion and Pay Per Click schemes) overrides natural harvesting of Web pages through crawlers. If these payment systems were included, it would have contaminated results. Closer investigation of the remaining engines showed that some search engines share a common database as their index, although the respective algorhitms would be different. From this investigation, it was clear that registration on one of these grouped engines would enable all the others in the group also to crawl the site.

Another list published by Weideman was used to extend the list of engines.The same criteria were used as for Sullivan's list, but only Weideman's 'standard' engines list was used.

The algorithm used to select the search engines was:

- select all engines available on the selected lists,
- eliminate all that require payment to register a Web page,
- from the remaining list, select only one engine for each database, and
- remove all those search engines which state that they register the submitted Web page to other search engine indices.

The last point above seems to imply that only crawlers from the selected search engines would visit the registered Web page. However, this was not an expectation during the early stages of the research, since it is certainly possible that a crawler could visit a Web page that has not explicitly been registered. A human user, for example, could visit the test Website (after having found it through one of the registered search engines), and link to it through his or her own Website. This would create the opportunity for any other crawler to visit and index the test Website.

The list of remaining search engines after this process was executed is:

Aardvark
AESOP
AllTheWeb
AltaVista
AOL
DMOZ
Google
MSN
Netscape
SearchHippo
SearchKing
TrueSearch
Yahoo!

Creating the test site

For the experiment to be successful, the five questions as well as the types of JavaScript™ needed to be incorporated. The questions could only be answered if different variations of the different link types could be tested. The following list describes all variations that needed to be tested (at least one unique document was used for each type of link identified):

- Text Links. Normal text links will be added to the homepage, to be used as base measurement. The expectation was that all the text links would be followed by the crawlers.
- Links of type <a href="javascript:...>. For these tests, the HTML code contained a <a href> tag that links it to the JavaScript™. The script then contains the link to the real document.
- JavaScript™ menus. These menus will link to documents through a JavaScript™ menu.
- Link of the type 'document.write()'. Scripts will be added to a page to dynamically add links to the HTML code.

For each of the above, three different ways of identifying the links were used:

- static strings in the script,
- simple concatenated strings, and
- strings built through special function calls.

For each of these three tests, two different types of scripts were created:

- scripts that are embedded in the HTML, and
- scripts that are in external ".js" files.

A further two options of each of the tests exists and links were created for them:

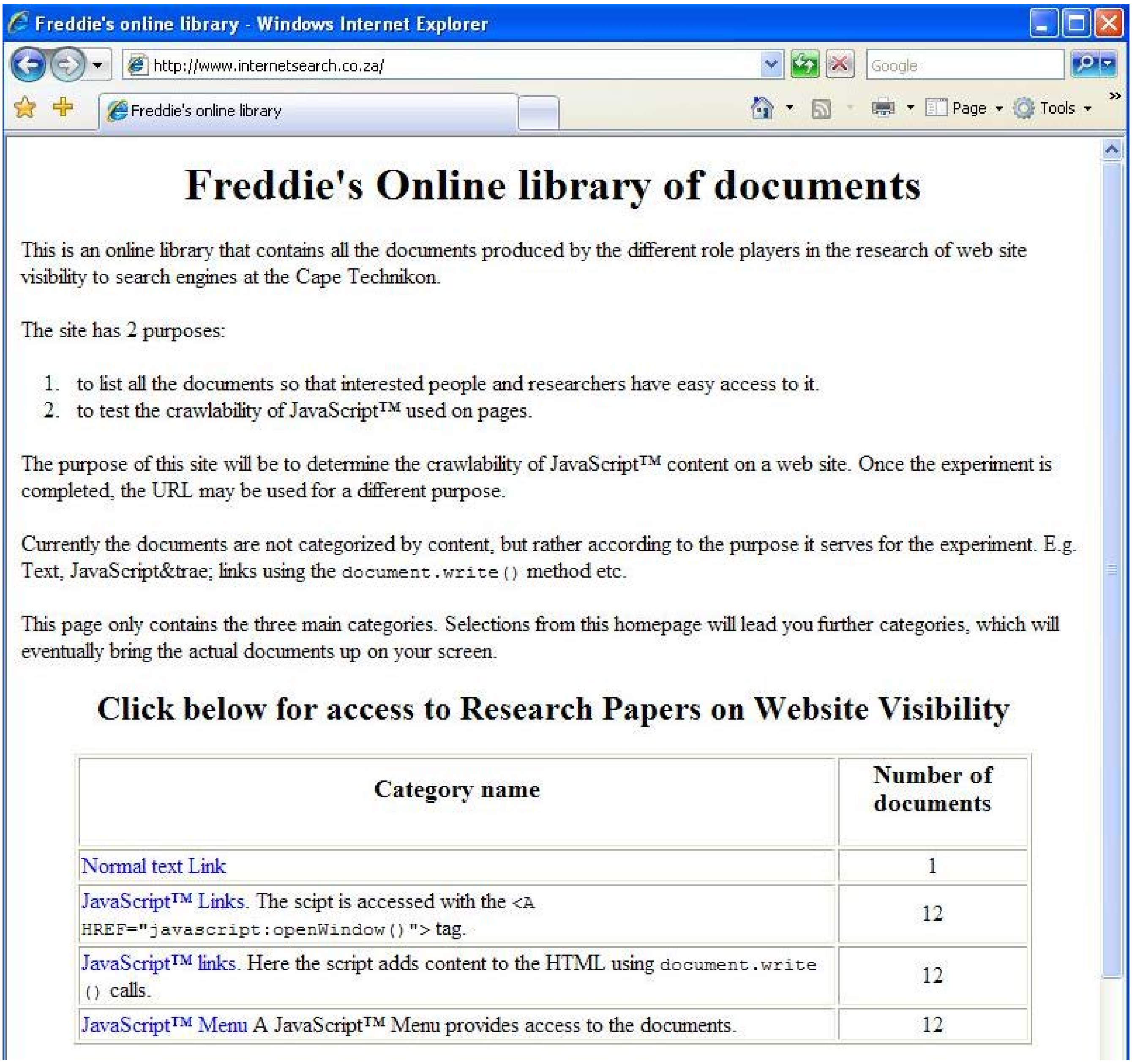
- links that contain relative paths to the documents and
- links that contains absolute paths to the documents.

This resulted in a large number of tests and documents that were needed for the experiment. A different document was to be used for each kind of link, so the number of unique documents required were calculated as follows:

```
Number of combinations of link types: (a * b * c * d) + e = 37

where a = 3 (types of JavaScript™ links)
b = 3 (possible ways to build the links)
c = 2 (types of scripts: internal & external)
d = 2 (ways the links can be identified: absolute & relative)
e = 1 (text link = 1x method)
                * = multiplication, + = addition
```

Figure 1 shows the main (home) page of the experimental Website. Note that each category will test a different kind of link.


Freddie's online library - Windows Internet Explorer
http://www.internetsearch.co.za/
Google
Freddie's online library
Page
Tools

Freddie's Online library of documents

This is an online library that contains all the documents produced by the different role players in the research of web site visibility to search engines at the Cape Technikon.

The site has 2 purposes:

1. to list all the documents so that interested people and researchers have easy access to it.
2. to test the crawlability of JavaScript™ used on pages.

The purpose of this site will be to determine the crawlability of JavaScript™ content on a web site. Once the experiment is completed, the URL may be used for a different purpose.

Currently the documents are not categorized by content, but rather according to the purpose it serves for the experiment. E.g. Text, JavaScript&trae; links using the `document.write()` method etc.

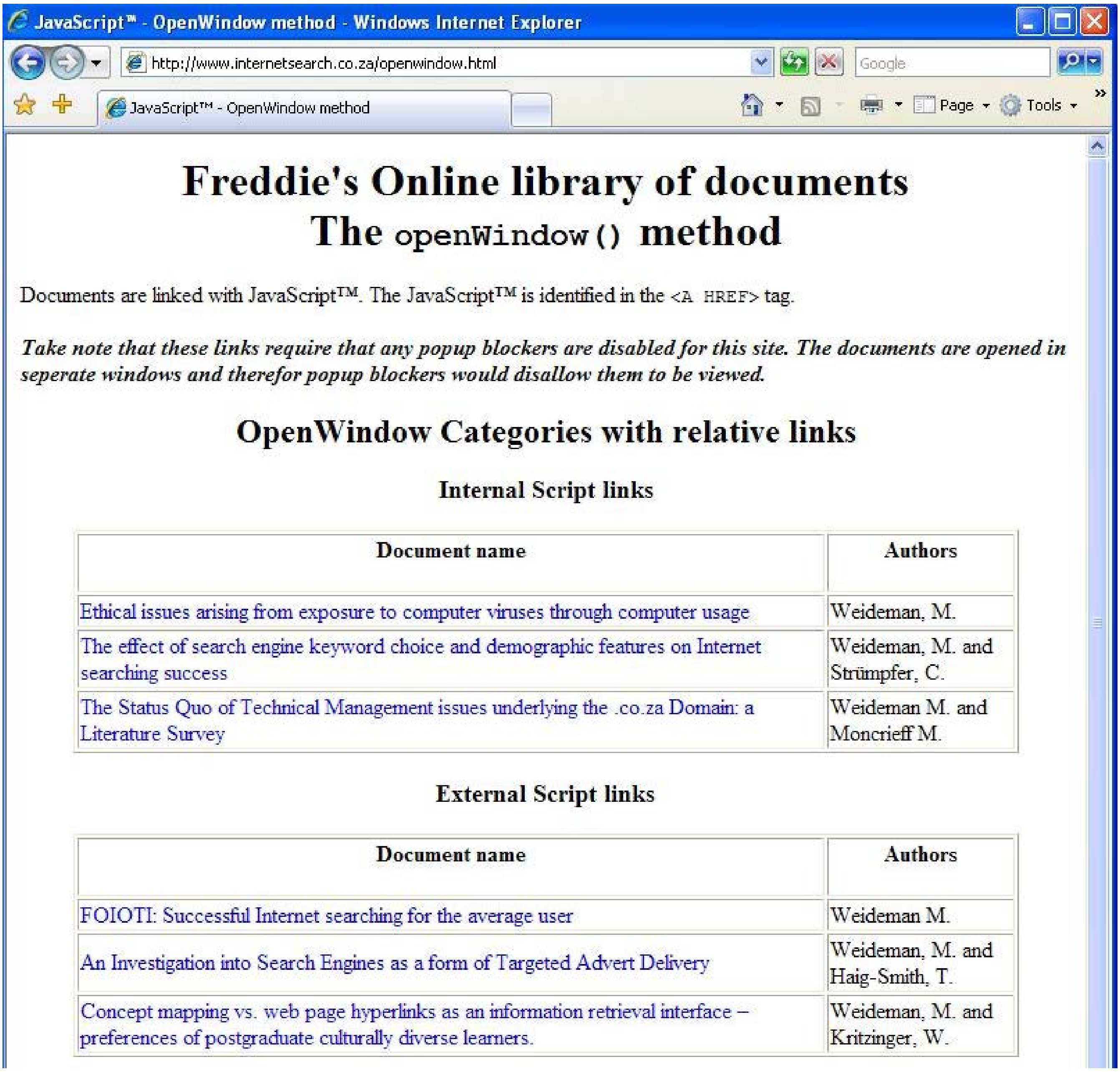
This page only contains the three main categories. Selections from this homepage will lead you further categories, which will eventually bring the actual documents up on your screen.

Click below for access to Research Papers on Website Visibility

Category name	Number of documents
Normal text Link	1
JavaScript™ Links. The scipt is accessed with the `<A HREF="javascript:openWindow()">` tag.	12
JavaScript™ links. Here the script adds content to the HTML using `document.write ()` calls.	12
JavaScript™ Menu A JavaScript™ Menu provides access to the documents.	12

Figure 1:The Home page of the experimental Web site (www.internetsearch.co.za)

Figure 2 shows the page created for using JavaScript™ in the 'open window ()' method


JavaScript™ - OpenWindow method - Windows Internet Explorer
http://www.internetsearch.co.za/openwindow.html
Google
JavaScript™ - OpenWindow method
Page
Tools

Freddie's Online library of documents
The `openWindow()` method

Documents are linked with JavaScript™. The JavaScript™ is identified in the `<A HREF>` tag.

Take note that these links require that any popup blockers are disabled for this site. The documents are opened in seperate windows and therefor popup blockers would disallow them to be viewed.

OpenWindow Categories with relative links

Internal Script links

Document name	Authors
Ethical issues arising from exposure to computer viruses through computer usage	Weideman, M.
The effect of search engine keyword choice and demographic features on Internet searching success	Weideman, M. and Strümpfer, C.
The Status Quo of Technical Management issues underlying the .co.za Domain: a Literature Survey	Weideman M. and Moncrieff M.

External Script links

Document name	Authors
FOIOTI: Successful Internet searching for the average user	Weideman M.
An Investigation into Search Engines as a form of Targeted Advert Delivery	Weideman, M. and Haig-Smith, T.
Concept mapping vs. web page hyperlinks as an information retrieval interface – preferences of postgraduate culturally diverse learners.	Weideman, M. and Kritzinger, W.

Figure 2: The page using JavaScript™ in the 'open window ()' method>

Registering the site at the search engines

The experimental home page was registered at all the selected search engines on 24 August 2004. On 14 September 2004, no results were observed in the logs, so the page was registered to the same search engines again. Shortly after this date some crawlers started visiting this site, and the log reflected in Table 2 was created.

The experimental Website

The experimental site was created as a data-generating device, not a human friendly Web page.The site content was chosen to simulate an online library of documents. Each document would be accessed by means of a unique link as described above.

The categories in the site were chosen in a way to describe the type of links that are used to access those documents. From the main home page, there are four categories:

- Text links.
- JavaScript™ links with the 'document.write()' method.
- JavaScript™ links with the 'openWindow()' method.
- JavaScript™ menu links.

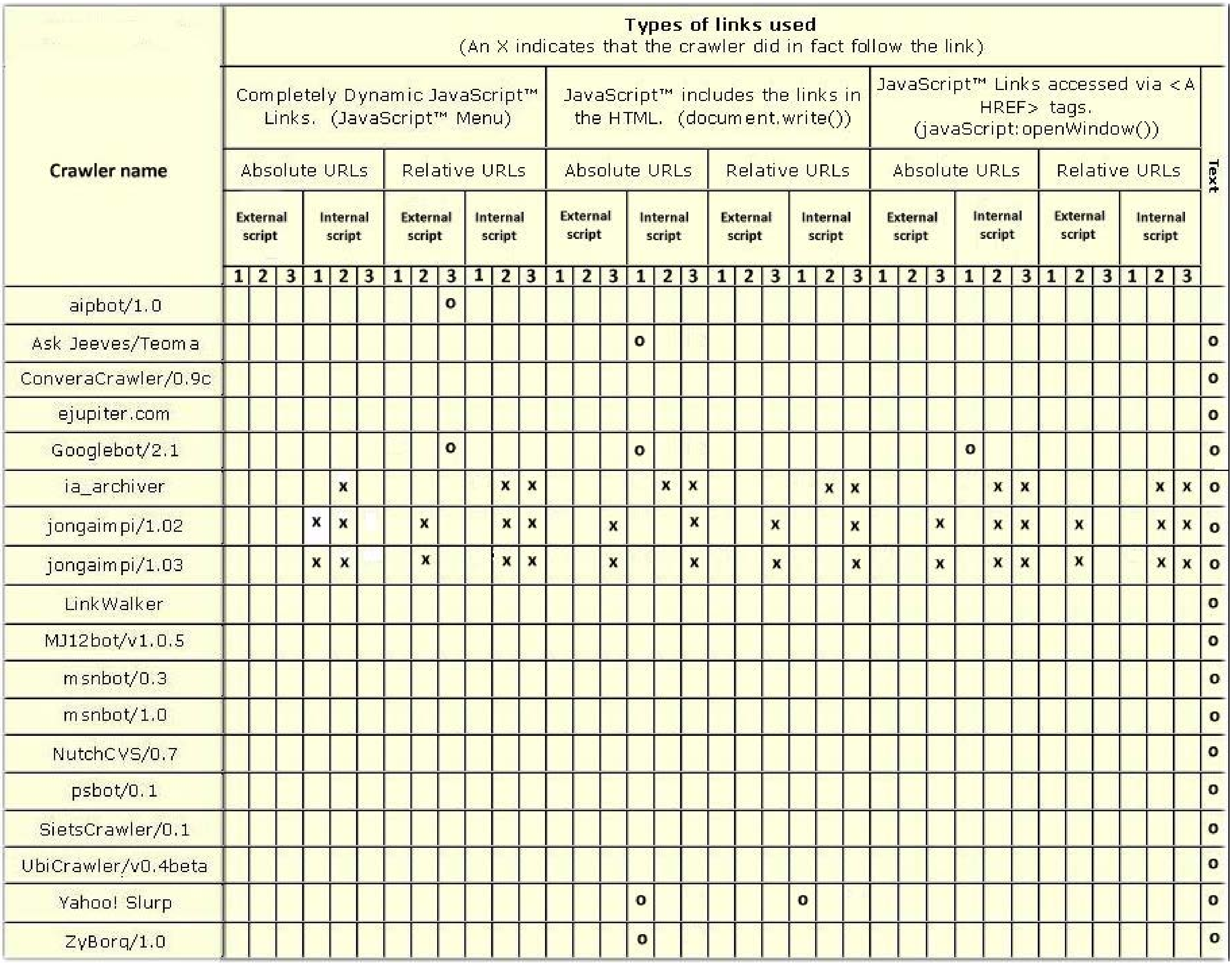
Each of these categories has its own page with the documents linked as described.

Results and interpretation

The Website was closely monitored up to and including 11 December 2005, yielding a total time span of 463 consecutive days. Every visit by a crawler to each one of the non-homepages was recorded in Table 2, where the following notations are used:

- "1" represents simple complete strings in the JavaScript™ code. Crawlers only need to parse the JavaScript™ code in order to crawl these links.
- "2" represents simple concatenated strings in the JavaScript™. Crawlers may be able to do simple string concatenations, which will allow them to crawl these links.
- "3" represents strings that were built by means of function calls. Crawlers will have to execute the JavaScript™ code to build these links.

An "O" in a box in Table 2 indicates that the crawler followed this specific type of link.An "X" indecates that the crawler attempted to follow the specific type of link but was unsuccessful for some reason.



Crawler name	Types of links used (An X indicates that the crawler did in fact follow the link)																																				
	Completely Dynamic JavaScript™ Links. (JavaScript™ Menu)												JavaScript™ includes the links in the HTML. (document.write())												JavaScript™ Links accessed via <A HREF> tags. (javaScript:openWindow())												Text
	Absolute URLs						Relative URLs						Absolute URLs						Relative URLs						Absolute URLs						Relative URLs						
	External script			Internal script			External script			Internal script			External script			Internal script			External script			Internal script			External script			Internal script			External script			Internal script			
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	
aipbot/1.0									o																												
Ask Jeeves/Teoma																o																					o
ConveraCrawler/0.9c																																					o
ejupiter.com																																					o
Googlebot/2.1									o							o												o									o
ia_archiver					x						x	x					x	x					x	x					x	x					x	x	o
jongaimpi/1.02				x	x			x			x	x			x			x			x			x			x		x	x		x			x	x	o
jongaimpi/1.03				x	x			x			x	x			x			x			x			x			x		x	x		x			x	x	o
LinkWalker																																					o
MJ12bot/v1.0.5																																					o
msnbot/0.3																																					o
msnbot/1.0																																					o
NutchCVS/0.7																																					o
psbot/0.1																																					o
SietsCrawler/0.1																																					o
UbiCrawler/v0.4beta																																					o
Yahoo! Slurp																o						o															o
ZyBorg/1.0																o																					o

Table 2: Search engine crawlers and the links that they have followed.

Sixty-five crawlers were noted as visiting the Website during the course of the experiment. Not all of them attempted to load any of the non-homepage pages. Some crawlers only attempted to load the robots.txt file and nothing more.

A detailed summary of the result, together with an interpretation of each point from Table 2 indicates the following:

- 27.7% (18) of all the visiting crawlers attempted to index some of the non-homepage documents.
  The authors are unable to explain why many of the crawlers did not attempt to index any of the documents, but a possible reason could be that some are experimental projects, which are not really interested in indexing any documents.
  Another possible reason could be that many of the crawlers are still under development. This is suggested by what seems to be the version number that forms part of the crawler's name.Many of the nn-indexing crawlers indicated that their version numbers are very low (below 1) or carry the word "beta" or "dev" as part of the number.
- The table indicates that quite a few crawlers attempted to follow JavaScript™ links, but not all were successful in doing so.
  44% (8) of the listed crawlers in the table, attempted to follow the JavaScript™ links. 62.5% (5) of them were successful in their attempts. The other three tried unsuccessfully to load documents.

This may not be conclusive, since the five successful crawlers represent only 7.6% of the total number of visiting crawlers. It can be noted that the unsuccessful crawlers all attempted type 2 and 3 links, which are less straightforward than the type 1 links. At this stage, no explanation can be given as to why this is happening.

- From the table, it can be seen that external, absolute JavaScript™ menu links were never attempted by any crawler. A possible reason could be that the URLs to these documents are so disguised in the JavaScript™ code that none of the crawlers was able to identify them as URLs.
  The logs indicate that 'ia_archiver' tried some links with only partial filenames. Closer inspection indicates that these partial filenames are coded in the HTML as single strings. This supports the suggestion of 'Chris' (2003) that crawlers may search for specific character combinations within files and interpret is as links, which it will try to follow.
- The logs also indicates that 'jongaimpi/1.02' and 'jongaimpi/1.03' attempted to load documents from the 'scripts' folder.This suggests that it would search for a combination of characters that can be interpreted as a URL in any file and does not distinguish between '.HTML', '.HTM' files and other types.The documents attempted in this manner were all external links, which would be defined in '.js' files within the 'scripts' directory.
- Some crawlers ('ia_archiver', 'jongaimpi/1.02' and 'jongaimpi/1.03') attempted to load most documents from the root directory, and ignored any directory indication in the URL, regardless of the type of link. Since they also tried only type 2 and 3 links the authors conclude that it is a result of the fact that the JavaScript™ is never executed, but rather a character pattern is found that that does not include the directory.

Conclusion

JavaScript™ seems to have found an application in the modern world that differs from its original intention. The literature survey clearly indicated that Website designers should choose the inclusion method for JavaScript™ based on a variety of factors. The literature survey further indicates that confusion appears to exist about the ability of crawlers to index JavaScript™. This provides challenging material for future research. A number of methods exist to exclude crawlers from reading JavaScript™ on Web pages.

This study produced one conclusive result, namely that text links are the preferred road for crawlers to access sub-pages on a Website. This implies that human-friendly, pop-up menus on Web pages, for example, are one of the stumbling blocks (identified in this project) that prevent the development of search-engine-friendly Websites. It is possible that different crawler algorithms would produce different results, but that fact could not be included in this research, because algorithm details are closely guarded trade secrets and, as such, were not available to be used to enhance the research results.

However, not all the research questions were answered, and more research will have to be done in order to obtain answers to all these questions.

Further research

This paper did not set out to find conclusive answers to all results regarding the influence of JavaScript™ on the visibility of a Web page to search engines. In fact, it is clear that more work would need to be done in order to be conclusive.

Possible areas of research include the following:

- Are links within JavaScript™ interpreted by search-engine crawlers?
- Is text within JavaScript™ indexed by search-engine crawlers?
- How does the inclusion of JavaScript™ on a page affect its visibility?
- How does the placement of embedded JavaScript™ affect its visibility?
- How does external JavaScript™ affect the site's visibility?

Acknowledgements

The authors would like to acknowledge the following institutions for supporting this research project: the National Research Foundation (South Africa) for funding to register the URL; Exinet (employer of one author), for providing the Webspace required to host the test Website and the Cape Peninsula University of Technology for providing overheads to make this project possible.

References

- Anonymous2. (2004). *Cookie and JavaScript* Message posted to HighRankings.com/forum archived at http://www.highrankings.com/forum/index.php?showtopic=3062
- Baartse, M., Conway, S., David, J., Li, S., McFarlane, N., Palmer, S.B., *et al.* (2001). *Professional JavaScript.* Indianapolis, IN: Wrox Press. [Chapter 4 of first edition available online] Retrieved 21 August, 2006 from http://www.Webreference.com/programming/javascript/professional/chap4/1/
- Brooks, T.A. (2003). *Web search: how the Web has changed information retrieval.* *Information Research*, **8**(3) paper 154. Retrieved 15 June 2004 from http://informationr.net/ir/8-3/paper154.html
- Brooks, T.A. (2004). *The nature of meaning in the age of Google.* *Information Research*, **9**(3) paper 180. Retrieved 15 June 2004 from http://informationr.net/ir/9-3/paper180.html
- Chambers, R. (2006). *Search engine strategies: a model to improve Website visibility for SMME Websites*, Unpublished master's thesis, Cape Peninsula University of Technology, Cape Town, South Africa.
- Champeon, S. (2001). *JavaScript: How did we get here?* Retrieved 19 May 2004 from

http://www.oreillynet.com/pub/a/javascript/2001/04/06/js_history.html

- Chris. (2003, March 24). Why a search engine crawler is not at all like Lynx. Message posted to http://www.searchguild.com (Search engine optimization (SEO) forums). Retrieved 8 June, 2006 from http://www.searchguild.com/tpage283-0.html
- Futterman, D. (2001). Making content findable. *Online*, **25**(3), 36-40.
- Gerensky-Greene, M. (2004). How to optimize a Web site using JavaScript menu. Retrieved 10 August, 2006 from http://tinyurl.com/jzjut
- Goetsch, D. (2003). *Search engine robots - how they work, what they do (Part I)*. Petaluma, CA: SearchInnovation.com. Retrieved 05 August, 2006 from http://www.searchinnovation.com/search-engine-robots-1.asp
- Google. (2006). Google information for Webmasters: Webmaster guidelines. Mountain View, CA: Google Inc. Retrieved 11 August, 2006 from http://books.google.com/webmasters/guidelines.html
- Hock, R. (2002). A new era of search engines: not just Web pages anymore. *Online*, **26**(5), 20-27.
- Koster, M. (n.d.) *The Web robots pages.* Retrieved 28 April, 2004 from http://www.robotstxt.org/wc/robots.html
- Mbikiwa, F.N. (2006). Search engine exclusion policies: implications on indexing e-commerce Websites. Unpublished master's thesis, Cape Peninsula University of Technology, Cape Town, South Africa.
- Netscape. (1998). *Netscape rallies content community around leading Web development technologies of today and tomorrow - JavaScript, XML and RDF*. Retrieved 12 April, 2006 from http://wp.netscape.com/newsref/pr/newsrelease599.html?cp=nws04flh2
- Nielsen, J. (2004). *When search engines become answer engines* Retrieved 10 April 2006 from http://www.useit.com/alertbox/20040816.html
- Ngindana, M.W. (2006). *Visibility of e-commerce Websites to search engines: a comparison between text-based and graphic-based hyperlinks.* Unpublished master's thesis, Cape Peninsula University of Technology, Cape Town, South Africa.
- Shiran, Y. & Shiran, T. (1998). *Learn advanced JavaScript programming.* Retrieved 20 May 2004 from http://www.ods.com.ua/win/eng/web-tech/js/
- Slocombe, M. (2004). Boost your site's search engine ranking. Retrieved 12 June, 2006 from http://www.urban75.org/tech/search_engine_index.html. (Originally published in *Internet Magazine*, May, 2004.
- Slocombe, M. (2003b). *Construct a foolproof navigation system* . Retrieved 20 August, 2006 from http://www.urban75.org/tech/navigation.html
- Thurow, S. (2003). *Search engine visibility*. Indianapolis, IN: New Riders.
- Thurow, S. (2004). *Webpronews - making the Web... smarter!* Retrieved 21 August, 2006 from http://tinyurl.com/6fnmx
- Venkata, R. (2003). *Seven Web site tips to attract search engine crawlers.* Retrieved 21 August, 2006 from http://www.businesstoolchest.com/articles/data/20031217130508.shtml
- Weideman, M. & Kritzinger, W. (2003). Search engine information retrieval: Empirical research on the usage of metatags to enhance Website visibility and ranking of e-commerce Websites. Proceedings of The 7th World Conference on Systemics, Cybernetics and Informatics, Volume VI (pp.231 - 236). Orlando, FL: International Institute of Informatics and Systemics.
- Weideman, M. (2004a). Ethical issues on content distribution to digital consumers via paid placements as opposed to Website visibility in search engine results. In T.W Bynum, N. Pouloudi, S. Rogerson & T. Spyrou (Eds.), *Proceedings of ETHICOMP 2004*, Volume 2 (pp. 904-915). Syros, Greece: Athens University of Economics and Business.
- Weideman, M. & Strumpfer, C. (2004). The effect of search engine keyword choice and demographic features on Internet searching success. *Information Technology and Libraries* , **23**(2), 58-65.
- Weideman, M. (2004c). Empirical evaluation of one of the relationships between the user, search engines, metadata and Websites in three-letter .com Websites. *South African Journal of Information Management*. Retrieved 20 November 2004 from http://general.rau.ac.za/infosci/raujournal/default.asp?to=peer3vol6nr3.